

**Automatic Generation of Task-specific Serial Mechanisms using
Combined Structural and Dimensional Synthesis**

Generierung aufgabenspezifischer serieller Roboterkinematiken durch
kombinierte Struktur- und Maßsynthese

Von der Fakultät für Maschinenbau
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades
Doktor-Ingenieur
genehmigte Dissertation

von

Daniel Andres Ramirez Rodriguez, M. Sc.

2018

1. Referent: Prof. Dr.-Ing. Tobias Ortmaier

2. Referent: Prof. Dr.-Ing. Jörg Wallaschek

Tag der Promotion: 18. Dezember 2018

Vorwort

Die vorliegende Arbeit ist das Ergebnis von vier Jahren Forschung am Institut für Mechatronische Systeme der Gottfried Wilhelm Leibniz Universität Hannover. Das wäre nicht möglich gewesen ohne die Hilfe und die Begleitung von mehreren Personen.

Zuerst bedanke ich mich ganz herzlich bei meinem Doktorvater, Herrn Professor Tobias Ortmaier, für sein Vertrauen und seine fachliche und persönliche Unterstützung während dieser Jahre. Ebenfalls danke ich Herrn Professor Jörg Wallaschek für die Anfertigung des Zweitgutachtens und Herrn Professor Roland Scharf für die Übernahme des Vorsitzes der Prüfungskommission.

Meinen Kollegen von imes und meinen Studenten möchte ich auch großen Dank für die tolle Zusammenarbeit äußern. Ich bin vor allem meinem Bürokollege Kai Eggers dankbar für seine Hilfe und die zahlreiche Korrekturlesen von meinen Veröffentlichungen.

Herzlicher Dank gilt meinem damaligen Gruppenleiter Herrn Dr.-Ing. Jens Kotlarski. Seine ständigen und wertvollen Ratschläge und die unzähligen fachlichen Diskussionen haben zur Gestaltung dieser Arbeit beigetragen. Wichtige Teile dieser Arbeit entstanden aus seinem Wissen und seiner Vision.

I would like to thank the faculty of mechanical Engineering of Universidad Santo Tomas Bogota and also the Administrative Department of Science, Technology, and Innovation -Colciencias-Colombia (call 529 2011) for the financial support of this project.

Meiner Familie bin ich zutiefst dankbar für seine Rückhalt während meiner Promotionszeit. Die Liebe und Geduld meiner Frau Paola und meines Sohnes Tomás haben die Fertigstellung dieser Arbeit ermöglicht, ich liebe euch.

Hannover, Dezember 2018

Daniel Ramirez

Kurzfassung

Obwohl die Vielfalt der in der Industrie verwendeten Roboterkinematiken in den letzten Jahren gestiegen ist, bieten die Roboterhersteller doch nur eine begrenzte Anzahl an unterschiedlichen kinematischen Strukturen an. Aufgrund dieser geringen Anzahl an Strukturen sind viele Roboter, vor allem in Industrienwendungen, für ihre Aufgabe überdimensioniert sowie überaktuiert und daher nicht optimal.

Bei der Entwicklung neuer Roboter manipulatoren sind sowohl die Struktur- als auch die Maßsynthese zu berücksichtigen. Während die Struktursynthese die Bestimmung der Strukturen mit den gewünschten Freiheitsgraden am Endeffektor ermöglicht, werden die Robotersegmente in der Maßsynthese optimiert. Beide Verfahren werden heutzutage jedoch getrennt durchgeführt. Das führt dazu, dass bei der Maßsynthese in der Regel nur eine einzige Struktur optimiert wird, ohne die vielzähligen möglichen kinematischen Alternativen zu berücksichtigen. Aus diesem Grund ist der aus diesem Prozess ermittelte Roboter nicht notwendigerweise für seine Aufgabe optimal.

Ziel der vorliegenden Arbeit ist die Entwicklung einer neuen Methode zur Synthese aufgabenspezifischer serieller Kinematiken unter Berücksichtigung sämtlicher geeigneter Strukturen. Durch die Ermittlung des Rangs der JACOBI Matrix und der Endeffektorgeschwindigkeit jeder, im Sinne der Spezifikation, möglichen Kombination der DENAVIT HARTENBERG Parameter werden im ersten Schritt alle aufgabengeeigneten Strukturen sowie ihre Optimierungsparameter bestimmt. Der Algorithmus benutzt anschließend die Abhängigkeit zwischen den gewünschten Endeffektorfreiheitsgraden und den geometrischen Parametern der Strukturen zur Erkennung möglicher Isomorphismen. Infolgedessen wird die Anzahl an Strukturen und Optimierungsparametern drastisch verringert, was die Optimierung sämtlicher aufgabengeeigneten Mechanismen ermöglicht. Um sowohl kinematische als auch dynamische Leistungsmerkmale berechnen zu können, werden die Kinematik und Dynamik der generierten Strukturen modelliert und ausführbarer Code für jede einzelne Struktur automatisch erzeugt. Dank diesem ist der Rechenaufwand der Dynamik im Vergleich zu üblichen numerischen Verfahren geringer. Die zu der gewünschten Aufgabe passenden Leistungsmerkmale bilden anschließend die Kostenfunktion und die Randbedingungen der Maßsynthese. Der aufgabenoptimale Manipulator ergibt sich schließlich aus der geometrischen Optimierung sämtlicher geeigneter Strukturen. Mit dem dargestellten Ansatz können Manipulatoren mit bis zu 6 Freiheitsgraden optimiert werden. Da der Großteil der Methode automatisch durchgeführt wird, ermöglicht dies eine erhebliche Reduktion des Entwicklungsaufwandes zur Auslegung neuer Roboter.

Schlagwörter: serielle Kinematiken, Struktursynthese, Maßsynthese, Modellierung.

Abstract

Although the diversity of robot kinematic structures that are used in industry has increased in the recent years, robot manufacturers still offer only a limited number of architectures. As most of the robots, especially in industrial applications, are chosen from a few different kinematic structures, they are oversized as well as overactuated and thus not optimal with respect to their task.

To develop new robot manipulators, two processes have to be considered, namely the structural and the dimensional synthesis. The former allows for determining the kinematic structures that fulfill a given number of degrees of freedom at the end effector. In the second, the dimensions of the robot links are optimized with respect to a desired performance index. However, these processes are actually performed separately. In the dimensional synthesis, only one structure is usually optimized without consideration of possible further mechanisms that are able to perform the desired task. Consequently, the obtained manipulator is not necessarily optimal with respect to its application.

This thesis presents a new approach for the synthesis of task specific serial manipulators considering all suitable structures. Initially, the task suitable structures as well as their optimization parameters are automatically determined from all DENAVIT HARTENBERG parameters combinations by inspecting the rank of the JACOBIAN matrix and the end effector velocity in symbolic form. Afterwards, isomorphisms are detected through the dependency of the desired degrees of freedom with respect to the geometric parameters. As a result, the number of structures and optimization parameters is significantly reduced allowing for the optimization of all task suitable structures. In order to be able to calculate kinematic and dynamic performance indices, the generated robotic structures are modeled and executable code is automatically generated. This code is individual for each architecture and offers a reduced computational effort compared with numeric methods. The performances indices are used to constitute the objective function and the constraints of the optimization problem. The task specific manipulator results from the geometric optimization of all suitable structures. The approach can synthesize optimal manipulators up to six degrees of freedom. As most of the method is automatically performed, it is a powerful tool for reducing the effort during the conceptual design phase of a robot. Additionally, it allows for the comparison of the performance between the structures as well as for their sensitivity with respect to changes in the optimal geometric parameters.

Keywords: serial kinematics, structural synthesis, dimensional synthesis, modeling.

Contents

Vorwort	iii
Kurzfassung	iv
Abstract	v
Nomenclature	ix
1 Introduction	1
1.1 Previous work	2
1.1.1 Structural synthesis	2
1.1.2 Geometric synthesis	4
1.1.3 Robot optimization considering modification of the architecture	5
1.2 Goal and organization of this work	7
2 Modeling of serial robot manipulators	11
2.1 Coordinates transformation	11
2.2 Forward kinematics	13
2.3 DENAVIT HARTENBERG parameters	14
2.4 Differential kinematics	16
2.5 Dynamics	17
3 Structural synthesis	21
3.1 Generation of suitable architectures	21
3.2 Detection of isomorphisms	24
3.2.1 Conditions for the link offset (d)	25
3.2.2 Conditions for the link length (a)	28
3.2.3 Conditions for the link twist (α)	31
3.2.4 Conditions for the joint angle (θ)	33
3.3 Obtained structures	35
4 Proposed combined robot synthesis	41
4.1 Optimization strategy	41
4.2 Definition of the optimization problem	44
4.3 Optimization parameters	45

4.4	Optimization algorithm	47
4.5	Modeling of general serial kinematics	50
4.5.1	Inverse kinematics	51
4.5.2	Derivation of the JACOBIAN matrix for the inverse kinematics	58
4.5.3	Collision evaluation	65
4.5.4	Dynamics	66
4.6	Postprocessing	69
5	Validation examples	71
5.1	Mechanisms with reduced number of DOF	71
5.1.1	Optimization with respect to a kinematic criterion	73
5.1.2	Optimization with respect to a dynamic criterion	77
5.2	Mechanisms with 6 DOF	82
6	Summary and future work	91
6.1	Summary	91
6.2	Future work	93
A	Architectures with 6 DOF	95
A.1	2P4R Architectures	95
A.2	1P5R Architectures	97
A.3	6R Architectures	100
B	Architectures with 4DOF	105
C	Specifications of the elements used in the dynamic optimization	109
	Bibliography	111

Nomenclature

Symbols and characters that are only used in one section or in one equation are described exclusively in the text.

General conventions

Body	capital letter (bold): B
Scalar	lowercase letter (italic): <i>i</i>
Vector	lowercase letter (bold and italic): <i>b</i>
Matrix	capital letter (bold and italic): <i>R</i>

Latin characters

a_i	length of the link i
a_i^*	length of the link i different from zero
\mathbf{b}	Rodrigues vector
\mathbf{c}	Coriolis and centrifugal effects vector
c_{N_c}	constraint of the optimization
C_i	i -th link's center of mass
d_i	offset of the link i
d_i^*	offset of the link i different from zero
\mathbf{e}_x	unit vector in x -direction
$\mathbf{e}_x^{(i)}$	unit vector in x -direction of (RF) $_i$
\mathbf{e}_y	unit vector in y -direction
$\mathbf{e}_y^{(i)}$	unit vector in y -direction of (RF) $_i$
\mathbf{e}_z	unit vector in z -direction
$\mathbf{e}_z^{(i)}$	unit vector in z -direction of (RF) $_i$
\mathbf{E}	skew symmetric matrix associated with the z -axis
\mathcal{F}	contact forces vector
\mathbf{g}	gravitation effect vector
\mathbf{g}_E	gravity acceleration vector
\mathbf{h}_g	friction effects vector (related to the gearboxes output)
h	objective function
${}_{(C_i)}\mathbf{I}_i^{(C_i)}$	inertia tensor of the i -th link described in (RF) $_{C_i}$

i	robot link, particle of the particle swarm optimization
\mathbf{I}	identity matrix
j	robot link
\mathbf{J}	JACOBIAN matrix of the manipulator
\mathbf{J}_t	translational part of the JACOBIAN
\mathbf{J}_r	rotational part of the JACOBIAN
$\bar{\mathbf{J}}$	homogeneous JACOBIAN matrix
$(j)\mathbf{J}_{j+1,n}$	JACOBIAN of the last $(n - j)$ joints
$\mathbf{J}_{\text{IK},k}$	JACOBIAN matrix for the inverse kinematics
k_{PSO}	iteration of the PSO algorithm
k_{dof}	step in the structural synthesis
k	iteration of the inverse kinematics
k_{dof}	iteration of the structural synthesis algorithm
k_{PSO}	iteration of the particle swarm optimization
\mathbf{K}_{ps}	diagonal matrix to modify the step size of the inverse kinematic algorithm when the pseudoinverse of $\mathbf{J}_{\text{IK},k}$ is used
\mathbf{K}_{tr}	diagonal matrix to modify the step size of the inverse kinematic algorithm when the transpose of $\mathbf{J}_{\text{IK},k}$ is used
L_c	characteristic length
\bar{L}_{links}	average length of the robot links
\mathbf{M}	mass matrix
m_i	mass of the i -th link
n	total number of links of a serial manipulator
N_{ar}	number of architectures obtained from the structural synthesis
$N_{\text{ar}}^{(x)}$	number of architectures obtained from the structural synthesis whose first axis is parallel to the x -axis
$N_{\text{ar}}^{(y)}$	number of architectures obtained from the structural synthesis whose first axis is parallel to the y -axis
$N_{\text{ar}}^{(z)}$	number of architectures obtained from the structural synthesis whose first axis is parallel to the z -axis
N_{task}	number of poses of a required task
\mathbf{P}	prismatic joint
\mathbf{p}	optimization parameters vector
p_w	optimization parameter
$\tilde{\mathbf{p}}$	optimal geometric parameters vector
\hat{p}_{min_w}	minimum value of the parameter p_w during the sensitivity analysis
\hat{p}_{max_w}	maximum value of the parameter p_w during the sensitivity analysis
\mathbf{q}	joint coordinates vector
q_i	joint coordinate

$\dot{\mathbf{q}}$	joint velocities vector
\dot{q}_i	joint velocity
$\dot{q}_{M_i i}$	velocity of the motor i before the gear train
${}^{(i)}\mathbf{r}_P$	position vector of the point P described in $(RF)_i$
${}^{(i)}\mathbf{r}_B$	position vector of $(RF)_B$ described in $(RF)_i$
${}^{(i)}\mathbf{r}_P^*$	homogeneous position vector
$\mathbf{r}_{i-1,i}$	relative position vector between $(RF)_{i-1}$ and $(RF)_i$
\mathbf{r}_{EE}	position vector of the reference frame fixed to the end effector
${}^{(W)}\mathbf{r}_{\text{pose}_s}$	end effector position of the required task
R	revolute joint
iR_j	rotation matrix of $(RF)_j$ with respect to $(RF)_i$
$R_x(\alpha)$	rotation matrix for a rotation angle α about x -axis
s	pose of the required task
S	skew-symmetric matrix
iT_j	homogeneous transformation matrix of $(RF)_j$ with respect to $(RF)_i$
R_{EE}	rotation matrix of the end effector reference frame with respect to the robot basis
cR_d	rotation matrix of $(RF)_{EE^*}$ with respect to $(RF)_{EE,k}$
T_{EE}	homogeneous transformation matrix of the end effector reference frame with respect to the robot basis
\mathbf{u}	rotation axis associated to the rotation cR_d
\mathbf{v}_n	linear velocity of the end effector
$\mathbf{v}_{\text{req}_n}$	required translational motion vector of the end effector
\mathbf{x}_{EE}	pose of the end effector
${}^{(W)}\mathbf{x}_{\text{task}_s}$	end effector pose of required task
\mathbf{x}_{EE^*}	desired end effector pose of the inverse kinematics
$\mathbf{x}_{EE,k}$	current end effector pose during the inverse kinematics

Greek characters

α_i	twist of the link i
Δt_s	time required to move the end effector from ${}^{(W)}\mathbf{x}_{\text{task}_s}$ to ${}^{(W)}\mathbf{x}_{\text{task}_{s+1}}$
$\Delta \mathbf{x}_k$	error vector between the desired end effector pose and the current pose during the inverse kinematics
$\Delta \psi_k$	orientation error between the desired end effector pose and the current pose during the inverse kinematics
ϕ	rotation angle associated with the rotation cR_d
${}^{(W)}\Phi_{\text{task}_s}$	end effector orientation of the required task
κ	condition number of \bar{J}

κ_{IK}	condition number of \bar{J}_{IK}
κ_{IK}^*	threshold value of κ_{IK} to determine the closeness to a singularity during the inverse kinematics algorithm
θ_i	joint angle of the link i
τ	vector of generalized forces associated to the generalized joint coordinates
τ_L	vector of actuators' generalized forces
ω_n	angular velocity of the end effector
ω_{req_n}	required rotational motion vector of the end effector
ω_j	angular velocity of the j -th link
ξ	velocity of the end effector
ξ_{req}	required motion vector
ξ_{req_i}	i -th element of the required motion vector different from zero
$\xi_{req}^{(z)}$	required motion vector for manipulators whose first joint axis is parallel to z -axis
$\xi_{req}^{(y)}$	required motion vector for manipulators whose first joint axis is parallel to y -axis
$\xi_{req}^{(x)}$	required motion vector for manipulators whose first joint axis is parallel to x -axis

Coordinate systems

$(RF)_i$	reference frame fixed to body i or link i
$(RF)_j$	reference frame fixed to body j or link j
$(RF)_B$	reference frame fixed to body B
$(RF)_{EE^*}$	desired end effector reference frame in the inverse kinematics algorithm
$(RF)_{EE,k}$	current end effector reference frame in the k -th iteration of the inverse kinematics
$(RF)_W$	world reference frame

Abbreviations

EE	end effector
DOF	degrees of freedom
CRS	combined robot synthesis
DH	DENAVIT-HARTENBERG
PSO	particle swarm optimization

1 Introduction

According to the international federation of robotics (IFR), the mechanical structure of industrial robots can be classified in four groups [IFR17]: articulated robots, SCARA robots, Cartesian robots, and parallel robots. The mechanical structure (also called architecture) of a robot concerns the number, the type, and the arrangement of the joints. As an example of two different architectures, figure 1.1 presents a SCARA robot and an articulated robot KUKA KR 16 with six degrees of freedom (DOF). The SCARA robot (figure 1.1(a)) consists of three consecutive revolute joints

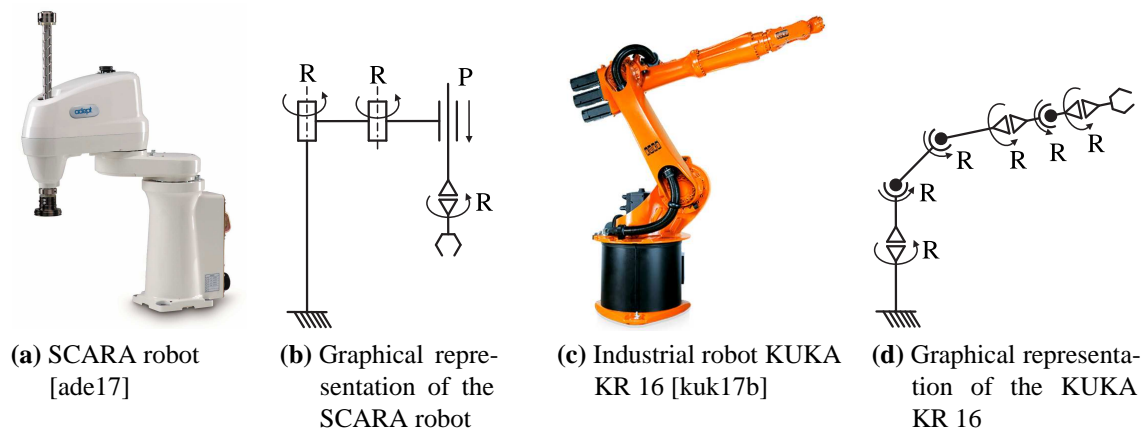


Figure 1.1: Example of two robots with different architectures. The graphical representation is according to VDI 2861 [VDI]

and one prismatic joint. All joint axes are parallel to each other as shown in figure 1.1(b). In the case of the KUKA KR 16 (figure 1.1(c)), its structure is formed by six revolute joints. The first joint axis is vertical, the second and third axes are parallel to each other but perpendicular to the first one. The last three joint axes intersect in one common point and form the wrist of the robot.

The number of architectures employed in both industrial and service robotics has increased in the last years. Among others, examples in industrial field are the LBR iiwa [Kuk17a], the UR serie from Universal Robots [Uni17], the Franka [Fra17], the Yumi [Yum17] and the Baxter [Bax17]. A similar trend can be observed in the development of robotic systems for medical applications [Hag11, BM03, RDM⁺99, Cyb17], space applications [DLR17], and personal service [Car17, Pro17, mar17, LWA17a].

The development of robots with new architectures has been partly favored by the (also increasing) offer of modular robots [Sch17c, Sch17a, LWA17b, igu17, Fes17]. In addition, open sources solutions [PLC17, ROS17] as well as improvements in rapid prototyping technologies such as 3D

printing [Fis10] allow for a rapid evaluation and construction of new kinematic concepts. However, there is still a lack of methods to determine the most appropriate architecture for a given application. Such an approach can be useful in the industrial field to choose the most appropriate architecture for a given task from a set of predefined robots. Furthermore, it can be applied in the design of user-tailored or task-customized robot solutions.

The usual conceptual robot design includes the structural and the geometric synthesis [Mer06, SK08]. The structural synthesis aims to find the architectures that fulfill a predefined number of DOF. In the latter, the dimensioning of the links (basically the links lengths) is determined in order to perform a task in the best possible way regarding given requirements. Previous works related to the structural synthesis as well as the geometrical synthesis are reviewed in section 1.1. At the end of the chapter, section 1.2 introduces the goal of the present work as well as its organization.

1.1 Previous work

The present section summarizes previous works related to the structural synthesis (section 1.1.1) followed by the approaches for the geometric synthesis (section 1.1.2) of serial manipulators. Approaches that have considered the modification of the architecture as part of the robot synthesis are introduced in section 1.1.3.

1.1.1 Structural synthesis

The goal of the structural synthesis (also called topologic or type synthesis) is to find the architectures that fulfill the necessary DOF to perform a given task. Most of the existing approaches have been developed for parallel manipulators, but they can also be applied to serial manipulators. They are based on graphs theory, the characteristic equation of position and orientation, screw theory, the Lie groups, and evolutionary morphology.

Graphs can be employed to describe mechanisms, and in the same way, they can be manipulated through the adjacency matrix to discover new mechanism topologies [Tsa99, PC08, PC05, HC15, LY17]. The approach helps to calculate the number of DOF at the end effector (EE) but not their type or direction [Mer06]. Due to the description methodology, two different mechanisms can be represented by the same graph (isomorphisms). In [KLZ99], neural networks were applied to identify these isomorphisms. An evolutionary approach was proposed in [GWXQ09] to modify the topology of serial and parallel mechanisms in order to find architectures that fulfill a set of required DOF. In this process, the movement characteristics of the generated kinematic chains were identified using graph theory. Lu [LL05] proposed the generation of embryonic graphs to generate new topologies. In addition, graphs have also been used in the synthesis of hybrid mechanisms [CBH08].

In the position and orientation characteristic (POC) equation [YLJ⁺09, YLS⁺13], the relations between the joint axes are represented by symbols indicating parallelism, perpendicularity, or other conditions. With them, a matrix representing the mechanism motion output is established. Thanks to special predefined operators between the kinematic chains, synthesis (inverse approach) and analysis (direct approach) of mechanisms can be carried out even for reconfigurable mechanisms [MLSY09].

Through the screw theory [Hun73], it is possible to describe the motion type (twist) and the restrictions (wrench) of a given mechanism [KG07a]. An advantage of this description is that the screw can show which DOF are rotational and which are translational. In addition, relations between the links of the same mechanism can be studied. During the synthesis, predefined operations between screws of kinematic chains are performed in order to generate new mechanisms. A known drawback in this case is that the calculated mechanism mobility is instantaneous and, therefore, can change inside the workspace [Mer06]. However, in [Dai12] and [PGM06], mechanisms were analyzed and synthesized using finite displacement screws. Kong and Gosselin [KG06, KG07b] proposed the use of a virtual kinematic chain to describe a desired EE movement. Then several kinematic chains were combined in a parallel architecture to generate the same movement. With this procedure, many parallel configurations up to 5 DOF have been proposed [Gla10, Kon03, KG07a, KG07b, Kon13]. Kuo [KD10], introduced several configuration rules to assemble serial kinematic chains considering six types of joints (revolute, prismatic, cylindrical, universal, spherical, flat). The rules were derived from the evaluation of the EE twist over the full motion rang of the joints.

Since the group of rigid body displacements can be treated as a Lie group, Hervé [Her99, HSJ⁺91, SH93, LMXL03] suggested the use of the Lie groups theory to carry out the type synthesis of parallel mechanisms. Therein, the movement of each kinematic chain was described by a displacement subgroup and the movement of the EE was calculated as the intersection set of these displacement subsets. For a serial chain, the displacement group of the EE results from the union of the links displacement subgroups. An important advantage of the group displacement is that the mobility of a mechanism is not instantaneous. Several serial [CKPA10, Her99, LH09] as well as parallel [LH09, LH10, LHH04, MLL07, RHNT06] structures up to 5 DOF have been proposed through the method. Recently, Caro [CKPA10] introduced several rules to generate Schönflies motion generators based on the groups of displacement. Moreover, he evaluated the generated mechanisms with regard to their complexity [KCAP07]. Nevertheless, additional comparison between the structures has not been considered.

Through the evolutionary morphology and the theory of linear transformations [Gog04, Gog05b], serial and parallel architectures can be automatically generated [Gog08, Gog09a, Gog10, Gog12, Gog14]. In [Gog02], several six DOF serial manipulators with only revolute joints were presented and classified with respect to their singularities in six different families. This methodology has been also used in the synthesis of isotropic [Gog04, Gog05a, Gog06c, Gog06b, Gog06a, Gog07] as well as decoupled parallel robots [RAF⁺07, Gog09b].

Since the approaches based on POC equation, screw theory and Lie groups are emphasized to provide architectures that fulfill a set of DOF, they do not provide information about the performance of the generated architectures with respect to other criteria. Furthermore, in order to connect the structural and the geometric synthesis, it is necessary to determine the modifiable geometric parameters of each architecture that can be used as optimization parameters. However, they can not be straightforwardly obtained with these approaches. In the usual design of robot manipulators, the geometric synthesis, is performed only for one architecture [Mer05b, MD08, SK08]. Consequently, the variable geometric parameters are arbitrarily chosen by the designer.

Evolutionary morphology, in turn, allows for extracting variable geometric parameters of the generated architectures. However, the methodology does not allow for an exact control over the direction of the joint axes because the direction of consequent joint axes is only defined as perpendicular or parallel. Additionally, it requires a high computational effort.

1.1.2 Geometric synthesis

The goal of the geometric synthesis (also known as dimensional synthesis) is the determination of the robot links dimensions in order to optimize the robot performance regarding a set of given requirements [Mer06, SK08]. As robot manipulators are mostly designed considering several criteria, multiobjective optimization is commonly used in this process. However, some examples of optimization regarding one single performance criterion can be cited.

The workspace has been considered as a design criterion in [GB02] to determine the dimensions of a planar parallel manipulator and a Stewart Gough platform in [Mer97]. An additional example is the dynamic isotropy proposed by Jiang [JHT10] to optimize the dynamic behavior of a Stewart Gough platform. Many examples of synthesis using several performance criteria can be found in the literature [COC08, RKL12, LMRD10, HM05, Mer05a, Kir00, KCZ12].

The fulfillment of the requirements is evaluated through performance indices. Some of the most commonly used indices are related to:

- robot size [PS15b, WGLZ12, HSKC12, KK93],
- robot workspace [CCGC07, VWT86, KB05],
- robot manipulability [Mer05a, PS15c, VWT86, KK93, KB05], and
- robot dynamics [PS15b, DGPC02, CCGC07, SKK02, WBH15, JSF07].

Detailed surveys about further performance indices and their classification can be found in [PS15a, SK08, WGLZ12, MSC⁺12a, COC07].

To deal with several performance indices, two techniques, Pareto fronts and weighted functions, have been employed [Wei09, GC00]. A Pareto front [Kir00, LLGCM10, RSP⁺11, LMRD10, AHSA09] is specially useful for antagonistic requirements. Nonetheless, the maximum number

of parameters that can be visualized is three. In addition, its application would be laborious for both the analysis and comparison of several architectures.

Weighted functions [ZN10, PSM⁺12, Mer05b] are composed by the weighted sum of several performance indices. They allow for the combination of several indices in one single value. However, the weight factors are arbitrarily assigned by the designer and the result of the optimization is very sensitive to their selection. A further issue is the mixture of different physical units, which can lead to expressions without physical meaning or the necessity of introducing additional scaling factors.

To solve the multiobjective optimization problem, different approaches have been applied. The minimax method was used in [COC07] to optimize one serial and one parallel manipulators using several scalar objective functions which consider workspace size, presence of singularities within the workspace, and manipulator stiffness. The approach attempts to minimize the worst-case value among the objective functions. In addition, weighting factors are used in order to scale the functions and make them dimensionless. Hao and Merlet [HM05] divided the design requirements into compulsory and relaxable. The geometries that satisfy all compulsory requirements simultaneously were found using interval analysis [HM05]. Afterward, a compromise for the relaxable requirements was determined by exploring the solution regions. Simulated annealing was also applied in [Che98, PS14] to optimize the geometry of several serial manipulators. The most widespread optimization approaches are genetic algorithms (GA) [Kir00, RSP⁺11, LMRD10, KCZ12]. As result, a Pareto front is obtained and the selection of the best solution is made by the designer. Until now, all proposed optimization algorithms have shown appropriate performance, however, there has been no comparison between them.

Even though the type of architecture has a close relation to the performance of robot manipulators [NA89, SK08, PS15b, PKPM⁺16, CBLW16], the dimensional synthesis is usually carried out for only one architecture. This is done under the assumption that a good optimization of the geometric parameters is enough to achieve adequate results [Mer05b, MD08]. Only a few publications, which will be introduced in the next section, addressed the optimization of manipulators considering several architectures.

1.1.3 Robot optimization considering modification of the architecture

Many of the studies considering changes of the robot architecture have been carried out for modular robots. Brandstötter introduced the ‘curved manipulator’ [BH15, BAH15, Bra16], which is a modular serial manipulator with six revolute joints (6R) that can be reconfigured according to a given task. This robot can be adapted to an individual tasks by assembling predefined links in different order and orientation.

Fischer [Fis10] used a set of predefined modules to build task suited robots. His goal was to provide a tool for rapid prototyping of robot systems. The modules were constructed using 3D

printing technology. Some of the modules were equipped with motors in order to provide an active prototype. The arrangement of the modules was determined by variation of the joint positions and the order of the modules. The performance of the suitable configurations was presented in a graphical interface for a ‘manual’ evaluation.

In [RSP⁺11], a GA was used to determine the geometry of a manipulator for maintenance operations in a tunnel boring machine. The manipulator was compounded of rotational joints and straight links. The orientation of the joint axes was limited to x -, y - or z -direction and the links length was limited to a set of predefined discrete values. The required task was defined with a trajectory. The tracking error, the number of DOF, the robot length, and the collision with the environment were considered as indicators for performance evaluation.

Patel [PS14, PS15b, PS15c], by his part, proposed the use of simulated annealing to optimize the geometric parameters of several six DOF manipulators for specific tasks considering joints limits, manipulability and robot size. In the studied manipulators, the last three joint axes crossed each other forming a wrist joint. Therefore, the inverse kinematic problem could be separated into a position and an orientation problem. The position problem was solved for the first three joints and the orientation problem for the last three revolute joints using the PSO optimization algorithm. As shown in [Plu16], this optimization strategy did not provide satisfactory results in manipulators with reduced mobility because the solution space is significantly constrained for particular architectures.

In [CR96, Che98] GA were used to determine the links lengths, the architecture type and the robot basis position to perform a given task while avoiding obstacles. The architecture was selected from a predefined set of robot types. An identifier was assigned to each robot type and considered as an optimization parameter. The results showed that there is a dominant mechanism to which all solutions converge after several iterations. In the published results, the robot types were not compared to each other.

Type synthesis and subsequent link optimization was introduced in [YBPG14] for linkages design. The method was applied on two spatial linkages with single DOF. Therein, finite position kinematic synthesis [PGM06, MS11, YMPG⁺12] was performed to find the joint axis. Then, in order to satisfy additional task requirements, the link lengths were modified by sliding the anchor points along the joint axes. Finite position kinematic synthesis has been also applied in the synthesis of tree topologies [SSPG14]. Graph representation of the articulated system was used to determine the topologies that can be paired with the task for the synthesis. This idea has been applied on the design of hands for manipulation tasks [TPGP16, HMKPG16]. The graph-based method was also used in [PC13] to enumerate the topologic alternatives of planar linkages (with a single DOF) for body guidance tasks.

As shown in this section, several approaches have been developed for the structural synthesis of manipulators considering the required DOF. The geometric synthesis has been addressed using different performance indices and optimization procedures. Furthermore, some authors addressed

the modification of the architecture as part of the robot synthesis. In spite of the appropriate results of these works, they only considered specific types of manipulators.

1.2 Goal and organization of this work

As introduced in the last section, there exists a variety of methods for the structural as well as the dimensional synthesis of robot manipulators. However, they are carried out independently. It means, the dimensional synthesis is usually performed with one single or even a couple of architectures which are arbitrarily chosen. Therefore, possible benefits of other suitable architectures are ignored and the optimized manipulator is not necessarily the best option for the required task. Although the problem has been addressed in the design of single DOF linkages for body guidance tasks [PC13] and in the finite position synthesis, these methods are not directly applicable in the optimization of robot manipulators. To perform a robot optimization considering all possible structures, it is necessary to define the set of structures capable of performing the task. For each of these structures, the extraction of the optimization parameters as well as the kinematic and dynamic modeling have to be automatically performed. Furthermore, the optimization algorithm have to be able to deal with any type of structure. An additional comparison between the performance of the structures has to be considered as well.

This work presents a combined robot synthesis (CRS) approach for serial manipulators up to 6 DOF (figure 1.2), which aims to obtain the best suitable manipulator for a required task. The desired degrees of freedom of the end effector (EE DOF) are established as part of the requirements. Then, in the *structural synthesis*, all architectures that exhibit the DOF are determined without isomorphisms allowing for an efficient consideration of all suitable architectures. The suitable architectures are described using the Denavit-Hartenberg parameters. This allows for the identification of the geometric parameters that can be modified without altering the EE DOF. The parameters are taken into account in the *modeling* phase which is automatically performed for each architecture. As a result, the equations for the differential kinematics and the dynamics are obtained in closed-form. Additionally, executable code (MATLAB[®] and C) is automatically generated for their calculation improving the computational efficiency. Once the kinematics and the dynamics modeling are available, they can be used to calculate the necessary performance indices according to the task requirements. Afterward, *each architecture is optimized* with respect to a main optimization criteria using the established geometric parameters as well as the position of the robot base as optimization parameters. Remaining performance requirements are considered as constraints in the optimization. In the next step, *the task optimal manipulator is selected* considering the results of the optimization. In order to evaluate the *sensitivity* of the chosen solution, the variation of the main performance index due to variations of the geometric parameters is studied for a set of architectures with the best performance in the final step.

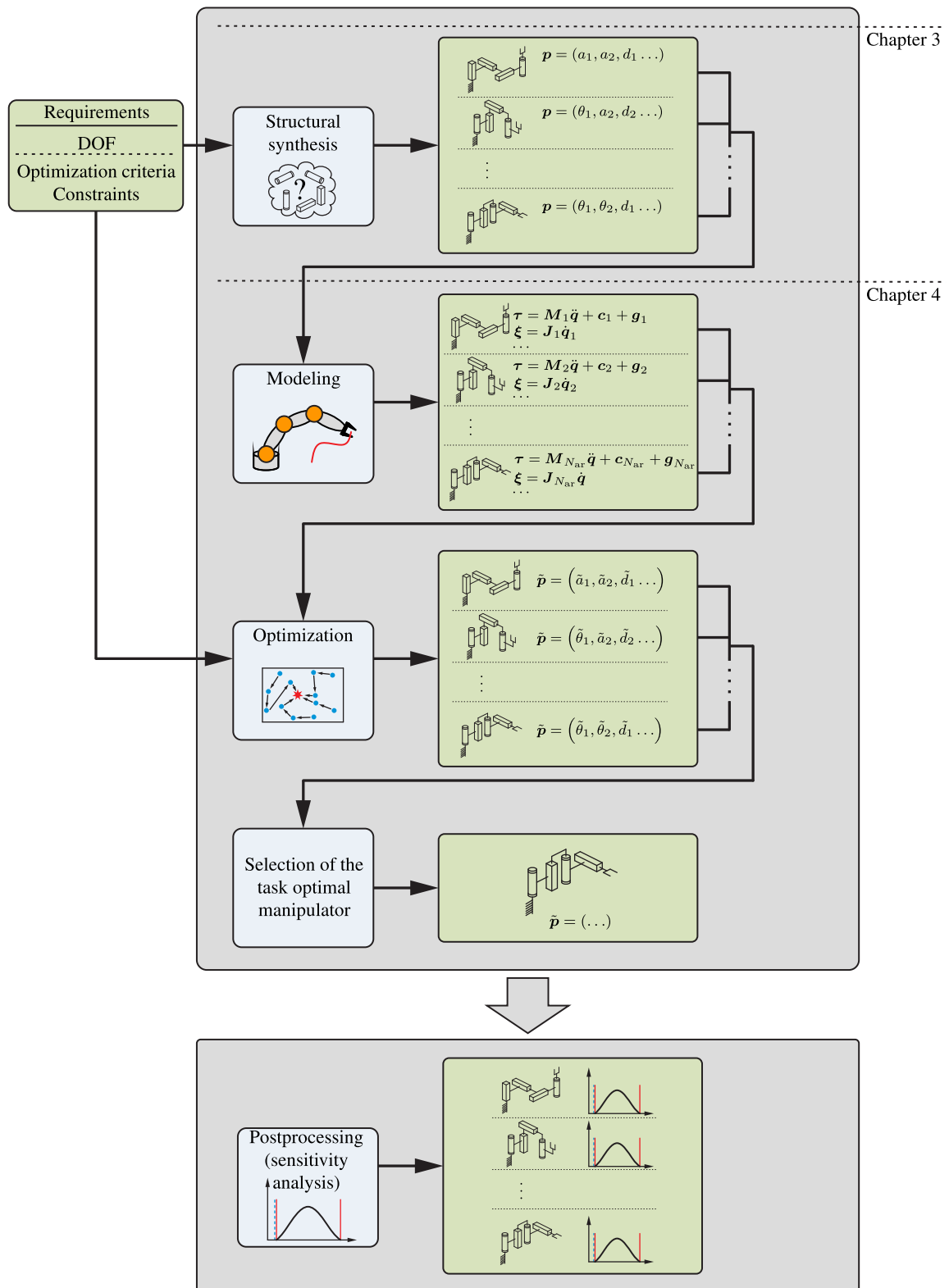


Figure 1.2: Proposed combined robot synthesis

The detailed description of each stage in the CRS is presented in this work, which is divided as follows: chapter 2 presents the fundamental concepts applied in the CRS. Based on the position and orientation description of rigid bodies (section 2.1), the forward kinematics of a serial manipulator (section 2.2) is introduced. An efficient way to systematize the robot kinematics calculation and to describe the robot links geometry is the DENAVIT HARTENBERG formalism shown in section 2.3. The modeling of the differential kinematics (section 2.4) as well as the dynamics (section 2.5) of a general serial manipulator are then explained.

A new approach for the structural synthesis of serial manipulators is introduced in chapter 3. The required DOF are used to determine the architectures that are able to perform the task as explained in section 3.1. The set of suitable architectures is reduced to a minimum set of mechanisms without isomorphisms according to the criteria in section 3.2. The procedure is illustrated step by step in section 3.3 through the synthesis of an exemplary architecture.

The next chapter (chapter 4) addresses the optimization strategy of the suitable structures (section 4.1). It explains how the required task is defined to be used in the CRS and formulates the geometric optimization problem in section 4.2. The optimization parameters include the geometric parameters of each architecture (which were obtained as part of the structural synthesis) as well as the position of the robot base as presented in section 4.3. The particle swarm algorithm, summarized in section 4.4, is used in the CRS as optimization strategy. This section includes the procedure to find the optimal solution as well as the calculation of the objective function and the handling of constraints. Since the CRS can consider kinematic and dynamic requirements, an important aspect of the CRS is the kinematic and dynamic modeling of the manipulators. The inverse kinematics and the automatic generation of the robot dynamics are introduced in section 4.5. The inverse kinematics is calculated with a proposed numeric approach which combines the pseudoinverse and the transpose of the JACOBIAN matrix. In order to reduce the computational effort, executable code is automatically generated for the calculation of the manipulator dynamics. A further characteristic of the CRS is the possibility of analyzing the sensitivity of the obtained solutions as explained in section 4.6.

The next chapter validates the capability of the CRS through three exemplary optimizations. Section 5.1 presents the first two examples using 4 DOF manipulators. In the first case, the optimization is performed with respect to a kinematic performance index. In the second case, the same task is employed to obtain the optimal manipulator with respect to a dynamic performance index. In order to show the capability of the CRS to deal with manipulators up to 6 DOF, in the third example (section 5.2), full kinematics manipulators are optimized with respect to a kinematic criterion to perform a different task.

The work finishes with the conclusions and a brief description of future research topics derived from the results.

2 Modeling of serial robot manipulators

The present chapter describes the theoretical background needed for the modeling of serial robot manipulators. These concepts will be used in the subsequent chapters to develop the CRS. Section 2.1 presents the position and orientation description of a rigid body in the space using homogeneous transformation matrices. This theory is used in section 2.2 to express the end effector pose of a general serial manipulator in terms of its joint positions known as forward kinematics. The DENAVIT HARTENBERG convention in section 2.3 is a well-known approach to describe the geometry of a robot as well as to simplify its forward kinematics calculation. Furthermore, the differential kinematics and the dynamics of serial manipulators in section 2.4 and section 2.5 respectively complement the tools necessary for the modeling of a general serial manipulator. Additional information about the modeling of serial manipulators can be found in [SK08, SSV09, DK07, Cra05].

2.1 Coordinates transformation

The spatial pose of a rigid body B is represented through the body fixed reference frame $(RF)_B$, which is attached to B as shown in figure 2.1. The position vector ${}^{(i)}r_B \in \mathbb{R}^3$ describes the relative

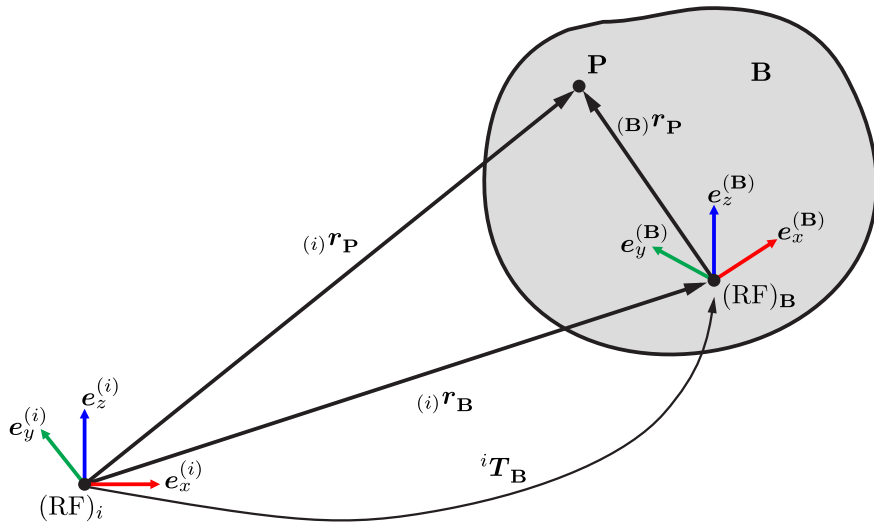


Figure 2.1: Pose description of a general rigid body B

position of $(RF)_B$ with respect to a reference frame $(RF)_i$. The orientation of $(RF)_B$ with respect to $(RF)_i$ is represented by the rotation matrix

$${}^i R_B = ({}^{(i)}e_x^{(B)}, {}^{(i)}e_y^{(B)}, {}^{(i)}e_z^{(B)}) \in SO(3), \quad (2.1)$$

where ${}^{(i)}\mathbf{e}_x^{(B)}$, ${}^{(i)}\mathbf{e}_y^{(B)}$, and ${}^{(i)}\mathbf{e}_z^{(B)}$ are orthogonal unit vectors in a righthand system and, therefore, fulfill the following properties:

$$\|\mathbf{e}_x\|_2 = \|\mathbf{e}_y\|_2 = \|\mathbf{e}_z\|_2 = 1, \quad (2.2)$$

$$\mathbf{e}_x^\top \mathbf{e}_y = \mathbf{e}_x^\top \mathbf{e}_z = \mathbf{e}_y^\top \mathbf{e}_z = 0, \quad (2.3)$$

$$\mathbf{e}_x \times \mathbf{e}_y = \mathbf{e}_z. \quad (2.4)$$

Hereafter, the unit vectors \mathbf{e}_x , \mathbf{e}_y , and \mathbf{e}_z are depicted in red, green, and blue color, respectively.

Position and orientation can be combined to form the homogeneous transformation matrix

$${}^i\mathbf{T}_B = \left(\begin{array}{ccc|c} {}^i\mathbf{R}_B & & & ({}^i)\mathbf{r}_B \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \in \text{SE}(3) \quad (2.5)$$

of $(\text{RF})_B$ with respect to $(\text{RF})_i$.

The relative position vector ${}^{(B)}\mathbf{r}_P$ describes the position of P with respect to the origin of $(\text{RF})_B$ in terms of $(\text{RF})_B$. It is used to determine the position vector ${}^{(i)}\mathbf{r}_P$ of point P , belonging to the rigid body B expressed in $(\text{RF})_i$ as follows

$${}^{(i)}\mathbf{r}_P = ({}^i)\mathbf{r}_B + {}^i\mathbf{R}_B ({}^{(B)}\mathbf{r}_P) = ({}^i)\mathbf{r}_B + ({}^i)\mathbf{r}_{B,P}. \quad (2.6)$$

Vectors ${}^{(i)}\mathbf{r}_P$ and ${}^{(B)}\mathbf{r}_P$ can be expanded with a fourth homogeneous coordinate to form the homogeneous position vectors

$${}^{(i)}\mathbf{r}_P^* = ({}^{(i)}\mathbf{r}_P^\top, 1)^\top \quad \text{and} \quad ({}^{(B)}\mathbf{r}_P^* = ({}^{(B)}\mathbf{r}_P^\top, 1)^\top. \quad (2.7)$$

Equation (2.6) can now be rewritten in terms of the homogeneous transformation matrix and the homogeneous position vectors as

$${}^{(i)}\mathbf{r}_P^* = {}^i\mathbf{T}_B ({}^{(B)}\mathbf{r}_P^*). \quad (2.8)$$

Rigid transformations can be concatenated. Therefore, in order to describe successive transformations, homogeneous transformation matrices, as well as rotation matrices, are multiplied. As shown in figure 2.2, the pose of the reference frame $(\text{RF})_n$ with respect to a fixed reference frame $(\text{RF})_0$ is described through the successive transformations from $(\text{RF})_0$ to $(\text{RF})_n$:

$${}^0\mathbf{T}_n = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \dots {}^{n-2}\mathbf{T}_{n-1} {}^{n-1}\mathbf{T}_n. \quad (2.9)$$

Similarly, the resulting rotation matrix of successive rotations is calculated as

$${}^0\mathbf{R}_n = {}^0\mathbf{R}_1 {}^1\mathbf{R}_2 \dots {}^{n-2}\mathbf{R}_{n-1} {}^{n-1}\mathbf{R}_n. \quad (2.10)$$

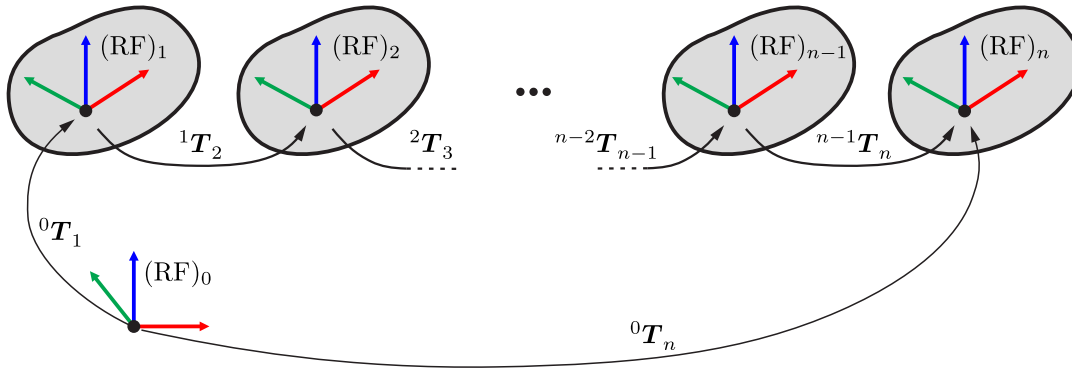


Figure 2.2: Successive rigid transformations

2.2 Forward kinematics

The model of a general serial manipulator considered in this work is composed by n joints and $n + 1$ links (each joint connecting two links) as shown in figure 2.3. Only revolute (R) and prismatic (P) joints are taken into account. The parameters describing the geometry of a link are depicted in figure 2.4 and explained in section 2.3. The joint coordinate q_i of the i -th joint represents the joint rotational angle θ_i in the case of a revolute joint, or the joint displacement d_i in the case of a prismatic joint. The base of the robot is represented by the link 0 with its reference frame $(RF)_0$. The n -th link with its reference frame $(RF)_n$ corresponds to the EE. The reference frame $(RF)_i$ is rigidly attached to the i -th link in order to describe its pose.

The relation between the robot joint coordinates and the pose of the EE is called the forward kinematics of the robot. This can be expressed as

$$\mathbf{x}_{EE} = \mathbf{f}(\mathbf{q}), \quad (2.11)$$

with

$$\mathbf{q} = (q_1, q_2, \dots, q_n)^T \quad (2.12)$$

being the joint coordinates vector and \mathbf{x}_{EE} the pose of the EE. The successive transformations shown in (2.9) can be used to find the homogeneous transformation matrix \mathbf{T}_{EE} of the EE reference frame $(RF)_n$ with respect to the robot base reference frame $(RF)_0$:

$$\mathbf{T}_{EE}(\mathbf{q}) = {}^0\mathbf{T}_n(\mathbf{q}) = {}^0\mathbf{T}_1(q_1) {}^1\mathbf{T}_2(q_2) \dots {}^{i-1}\mathbf{T}_i(q_i) \dots {}^{n-1}\mathbf{T}_n(q_n). \quad (2.13)$$

It has to be noted that the homogeneous transformation matrix ${}^{i-1}\mathbf{T}_i$, which defines the transformation between the reference frames $(RF)_{i-1}$ and $(RF)_i$, is a function of the joint coordinate q_i .

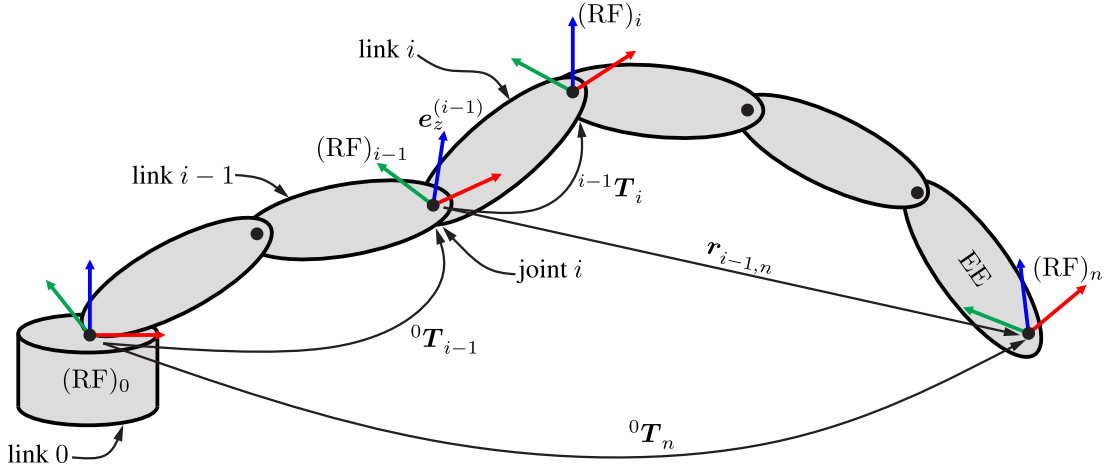


Figure 2.3: Location of the body fixed reference frames in a general serial manipulator

As $T_{EE}(\mathbf{q})$ in (2.13) presents the same structure shown in (2.5), the rotation matrix R_{EE} and the position vector \mathbf{r}_{EE} of the reference frame fixed to the EE can be extracted from $T_{EE}(\mathbf{q})$ using

$$\left(\begin{array}{c|c} R_{EE}(\mathbf{q}) & \mathbf{r}_{EE}(\mathbf{q}) \\ \hline 0 & 1 \end{array} \right) = T_{EE}(\mathbf{q}), \quad (2.14)$$

where T_{EE} , R_{EE} , and \mathbf{r}_{EE} are used instead of ${}^0T_{EE}$, ${}^0R_{EE}$, and ${}_{(0)}\mathbf{r}_{EE}$ just to simplify the notation.

2.3 DENAVIT HARTENBERG parameters

In this work, the classical DENAVIT HARTENBERG (DH) formalism [Pau81] is used to describe the geometry of a general robot link. According to this convention, the reference frame $(RF)_i$, is located as shown in figure 2.4, fulfilling the next conditions:

- the origin of the reference frame $(RF)_i$ is located in the intersection of joint axis $i + 1$ with the common normal between joint axes i and $i + 1$,
- the axis e_z of $(RF)_i$ must be pointing in the direction of the joint axis $i + 1$,
- the axis e_x of $(RF)_i$ must be pointing in the direction of the common normal between joint axes i and $i + 1$.

Following this considerations, the DENAVIT HARTENBERG parameters can be defined as:

- joint angle θ_i – the angle between $e_x^{(i-1)}$ and $e_x^{(i)}$ measured about the axis $e_z^{(i-1)}$ (i -th joint axis). If the joint i is a revolute (R) joint, θ_i corresponds to the joint coordinate q_i ,
- link offset d_i – the distance between $S_{N_{i-1}}$ (intersection of joint axis i with the common perpendicular between joint axes $i - 1$ and i) and D_{N_i} (intersection of joint axis i with the

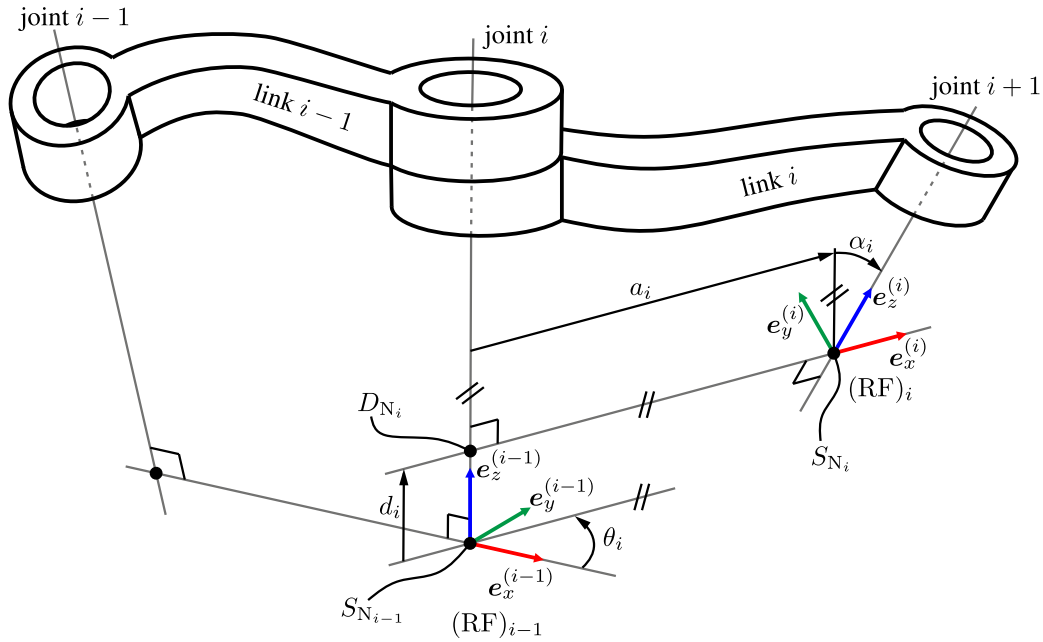


Figure 2.4: Location of the reference frame fixed to the robot link i according to DENAVIT HARTENBERG convention

common perpendicular between joint axes i and $i + 1$) measured along $e_z^{(i-1)}$. If the joint i is a prismatic (P) joint, d_i corresponds to the joint coordinate q_i ,

- link length a_i – the distance between $e_z^{(i-1)}$ and $e_z^{(i)}$ measured in the direction of $e_x^{(i)}$,
- link twist α_i – the angle between $e_z^{(i-1)}$ and $e_z^{(i)}$ measured about $e_x^{(i)}$.

The transformations due to the joint angle θ_i and the link twist α_i can be expressed as

$$\mathbf{T}_{r_z}(\theta_i) = \begin{pmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{T}_{r_x}(\alpha_i) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (2.15)$$

where $c\theta_i = \cos(\theta_i)$, $c\alpha_i = \cos(\alpha_i)$, $s\theta_i = \sin(\theta_i)$, and $s\alpha_i = \sin(\alpha_i)$ with the purpose of simplifying the notation. Further, the transformation due to the parameters a_i and d_i is represented by

$$\mathbf{T}_t(a_i, 0, d_i) = \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.16)$$

The transformation between $(\text{RF})_{i-1}$ and $(\text{RF})_i$ can be calculated using the successive transformations $\mathbf{T}_{r_z}(\theta_i)$, $\mathbf{T}_t(a_i, 0, d_i)$, and $\mathbf{T}_{r_x}(\alpha_i)$:

$${}^{i-1}\mathbf{T}_i = \mathbf{T}_{r_z}(\theta_i) \mathbf{T}_t(a_i, 0, d_i) \mathbf{T}_{r_x}(\alpha_i) = \begin{pmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (2.17)$$

which is the homogeneous transformation matrix of the i -th link.

2.4 Differential kinematics

The differential kinematics describes the relation between the joint rates (also called generalized joint velocities) $\dot{\mathbf{q}} = (\dot{q}_1 \dots \dot{q}_n)^T$ and the velocity $\boldsymbol{\xi}$ of the EE. The first three components of $\boldsymbol{\xi}$ correspond to the linear velocity \mathbf{v}_n of the origin of reference frame $(\text{RF})_n$. The last three components represent the angular velocity $\boldsymbol{\omega}_n$ of $(\text{RF})_n$. The EE velocity can be determined from the joint rates as follows

$$\boldsymbol{\xi} = \begin{pmatrix} \mathbf{v}_n \\ \boldsymbol{\omega}_n \end{pmatrix} = \mathbf{J}\dot{\mathbf{q}}. \quad (2.18)$$

Both \mathbf{v}_n and $\boldsymbol{\omega}_n$ are represented in this case with respect to $(\text{RF})_0$. The variable \mathbf{J} is the JACOBIAN matrix of the manipulator

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_t \\ \mathbf{J}_r \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{j}}_{t_1} & \dots & \dot{\mathbf{j}}_{t_i} & \dots & \dot{\mathbf{j}}_{t_n} \\ \dot{\mathbf{j}}_{r_1} & \dots & \dot{\mathbf{j}}_{r_i} & \dots & \dot{\mathbf{j}}_{r_n} \end{pmatrix}. \quad (2.19)$$

In (2.19), $\dot{\mathbf{j}}_{t_i}$ represents the effect of \dot{q}_i on the linear velocity of the EE and $\dot{\mathbf{j}}_{r_i}$ the effect on the angular velocity. The matrices $\mathbf{J}_t \in \mathbb{R}^{3 \times n}$ and $\mathbf{J}_r \in \mathbb{R}^{3 \times n}$ are called the translational and rotational part of the JACOBIAN, respectively [SSVO09, Tsa99]. Using the description shown in figure 2.3, the i -th column of the JACOBIAN can be calculated (in case of a P joint) as

$$\begin{pmatrix} \dot{\mathbf{j}}_{t_i} \\ \dot{\mathbf{j}}_{r_i} \end{pmatrix} = \begin{pmatrix} \mathbf{e}_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \quad (2.20)$$

and (in case of a R joint) as

$$\begin{pmatrix} \dot{\mathbf{j}}_{t_i} \\ \dot{\mathbf{j}}_{r_i} \end{pmatrix} = \begin{pmatrix} \mathbf{e}_z^{(i-1)} \times \mathbf{r}_{i-1,n} \\ \mathbf{e}_z^{(i-1)} \end{pmatrix}. \quad (2.21)$$

Hereafter, the simplified notation $\mathbf{r}_{i-1,n}$ will be used instead of ${}_{(0)}\mathbf{r}_{i-1,n}$ to indicate that the vector $\mathbf{r}_{i-1,n}$ is described in $(\text{RF})_0$. The same holds true for ${}_{(0)}\mathbf{e}_z^{(i-1)}$, which will be written as $\mathbf{e}_z^{(i-1)}$ only for simplicity.

The unit vector $\mathbf{e}_z^{(i-1)}$, which corresponds to the z -axis of $(\text{RF})_{i-1}$, is collinear with the axis of the i -th joint. This vector can be obtained using the homogeneous transformation matrix ${}^0\mathbf{T}_{i-1}$:

$$\begin{pmatrix} \mathbf{e}_z^{(i-1)} \\ 0 \end{pmatrix} = {}^0\mathbf{T}_{i-1} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad (2.22)$$

with

$${}^0\mathbf{T}_{i-1} = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \dots {}^{i-2}\mathbf{T}_{i-1}. \quad (2.23)$$

The relative position vector $\mathbf{r}_{i-1,n}$ from the origin of $(\text{RF})_{i-1}$ to the origin of $(\text{RF})_n$ is given by

$$\mathbf{r}_{i-1,n} = \mathbf{r}_n - \mathbf{r}_{i-1}, \quad (2.24)$$

with

$$\begin{pmatrix} \mathbf{r}_n \\ 1 \end{pmatrix} = {}^0\mathbf{T}_n \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \mathbf{r}_{i-1} \\ 1 \end{pmatrix} = {}^0\mathbf{T}_{i-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \quad (2.25)$$

As it can be seen in (2.20) and (2.21), the physical units of \mathbf{J}_t differ from the units of \mathbf{J}_r . In some cases, as will be shown in subsequent sections, a dimensionally homogeneous JACOBIAN matrix $\bar{\mathbf{J}}$ must be calculated by means of the characteristic length L_c [RAGPP95]:

$$\bar{\mathbf{J}} = \begin{pmatrix} \frac{1}{L_c} \mathbf{J}_t \\ \mathbf{J}_r \end{pmatrix}, \quad (2.26)$$

with

$$L_c = \sqrt{\frac{\sum_{i=1}^n \|\mathbf{e}_z^{(i-1)} \times \mathbf{r}_{i-1,n}\|_2^2}{n}}. \quad (2.27)$$

Further approaches to calculate the characteristic length as well as to avoid the dimensional inhomogeneities problem can be found in the literature [Ang06, ALC92, AC00, RAGPP95, Gos92, MSPC16, LJ06], however, the evaluation of these methods is not part of this work.

2.5 Dynamics

The dynamics equations of a serial manipulator can be derived using the LAGRANGE formalism [SSVO09], the NEWTON-EULER formulation [SSVO09], methods based on the virtual work [CBCP13], or the Kane's equations [KL85, LLL00].

In the LAGRANGE formalism, the Lagrangian L of a system is calculated using the difference of the kinetic energy T and the potential energy U :

$$L = T - U. \quad (2.28)$$

From (2.28), the dynamics of a general serial manipulator with n joints can be described as

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} = \boldsymbol{\tau}, \quad (2.29)$$

where $\boldsymbol{\tau} \in \mathbb{R}^{n \times 1}$ corresponds to the vector of the generalized forces associated to the generalized joint coordinates \mathbf{q} . The vector $\boldsymbol{\tau}$ includes the contribution of the actuators' generalized forces, the joint friction generalized forces, as well as the joint generalized forces induced by external loads on the EE (contact forces, payload). If the manipulator links are considered as rigid bodies and the elasticity in the actuators and gearboxes is neglected, equation (2.29) yields the equation of the inverse dynamics:

$$\boldsymbol{\tau}_L = \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) - \mathbf{J}^T(\mathbf{q}) \boldsymbol{\mathcal{F}} + \mathbf{h}_g(\mathbf{q}, \dot{\mathbf{q}}), \quad (2.30)$$

being $\boldsymbol{\tau}_L \in \mathbb{R}^{n \times 1}$ the actuators' generalized forces vector, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ the mass matrix, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times 1}$ the Coriolis and centrifugal effects vector, and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^{n \times 1}$ the gravitational effect vector. The vector $\boldsymbol{\mathcal{F}} \in \mathbb{R}^{6 \times 1}$ represents the contact forces exerted by environment on the EE. The friction effects related to the joint side (gearbox output) are gathered in $\mathbf{h}_g(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times 1}$.

To calculate (2.30), the physical properties of the link i are defined according to the variables depicted in figure 2.5. The origin of the reference frame $(\text{RF})_{C_i}$ is located on the center of mass C_i of the link i and has the same orientation as $(\text{RF})_i$, which was defined through the DH parameters in section 2.3. The position of C_i is described through the position vector \mathbf{r}_{C_i} :

$$\begin{pmatrix} {}^{(0)}\mathbf{r}_{C_i} \\ 1 \end{pmatrix} = {}^0\mathbf{T}_i \begin{pmatrix} {}^{(i)}\mathbf{r}_{C_i} \\ 1 \end{pmatrix}. \quad (2.31)$$

The matrix ${}^0\mathbf{T}_i$ can be found analog to (2.5):

$${}^0\mathbf{T}_i = \left(\begin{array}{ccc|c} {}^0\mathbf{R}_i & & & {}^{(0)}\mathbf{r}_i \\ \hline 0 & 0 & 0 & 1 \end{array} \right) = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \dots {}^{i-1}\mathbf{T}_i. \quad (2.32)$$

The mass matrix $\mathbf{M}(\mathbf{q})$ is calculated using the mass m_i of the i -th link and the inertia tensor ${}_{(C_i)}\mathbf{I}_i^{(C_i)}$ relative to its center of mass:

$$\mathbf{M}(\mathbf{q}) = \sum_{i=1}^n \left(m_i \mathbf{J}_{\mathbf{t}(C_i)}^T \mathbf{J}_{\mathbf{t}(C_i)} + \mathbf{J}_{\mathbf{r}(C_i)}^T {}^0\mathbf{R}_{i(C_i)} \mathbf{I}_i^{(C_i)} {}^0\mathbf{R}_i^T \mathbf{J}_{\mathbf{r}(C_i)} \right), \quad (2.33)$$

with

$${}_{(C_i)}\mathbf{I}_i^{(C_i)} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix}. \quad (2.34)$$

Values I_{xx} , I_{yy} , and I_{zz} denote the moment of inertia of the link i about the axes x , y , and z of $(\text{RF})_{C_i}$, respectively. Entries I_{xy} , I_{xz} , and I_{yz} are the products of inertia with respect to the same axes of $(\text{RF})_{C_i}$.

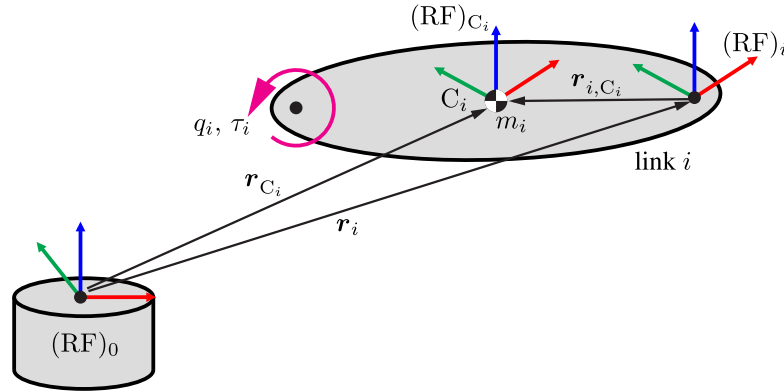


Figure 2.5: Simplified description of the link i

The matrices $\mathbf{J}_{t(C_i)} \in \mathbb{R}^{3 \times n}$ and $\mathbf{J}_{r(C_i)} \in \mathbb{R}^{3 \times n}$ in (2.33) correspond to the translational and rotational part of the JACOBIAN that maps the joint rates into the linear and angular velocity of $(\text{RF})_{C_i}$, respectively:

$$\mathbf{J}_{t(C_i)} = \begin{pmatrix} \dot{\mathbf{j}}_{t_1}^{(C_i)} & \dot{\mathbf{j}}_{t_2}^{(C_i)} & \dots & \dot{\mathbf{j}}_{t_i}^{(C_i)} & \mathbf{0} & \dots & \mathbf{0} \end{pmatrix} \quad (2.35)$$

$$\mathbf{J}_{r(C_i)} = \begin{pmatrix} \dot{\mathbf{j}}_{r_1}^{(C_i)} & \dot{\mathbf{j}}_{r_2}^{(C_i)} & \dots & \dot{\mathbf{j}}_{r_i}^{(C_i)} & \mathbf{0} & \dots & \mathbf{0} \end{pmatrix}. \quad (2.36)$$

$\mathbf{J}_{t(C_i)}$ and $\mathbf{J}_{r(C_i)}$ are calculated in a similar way as (2.20) and (2.21), but referred to $(\text{RF})_{C_i}$ instead of $(\text{RF})_n$. It means, for a P joint

$$\dot{\mathbf{j}}_{t_i}^{(C_i)} = \dot{\mathbf{j}}_{t_i} \quad \text{and} \quad \dot{\mathbf{j}}_{r_i}^{(C_i)} = \dot{\mathbf{j}}_{r_i}; \quad (2.37)$$

for a R joint

$$\dot{\mathbf{j}}_{t_i}^{(C_i)} = \mathbf{e}_z^{(i-1)} \times \mathbf{r}_{i-1, C_i} \quad \text{and} \quad \dot{\mathbf{j}}_{r_i}^{(C_i)} = \dot{\mathbf{j}}_{r_i}. \quad (2.38)$$

The i -entry of the vector corresponding to the Coriolis and centrifugal effects is calculated as

$$c_i(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{j=1}^n \sum_{l=1}^n \left(\frac{\partial M_{ij}}{\partial q_l} - \frac{1}{2} \frac{\partial M_{lj}}{\partial q_i} \right) \dot{q}_l \dot{q}_j, \quad (2.39)$$

where M_{rc} represents the element in the r -th row and c -th column of the mass matrix $\mathbf{M}(\mathbf{q})$.

Similarly, the i -th entry of the gravitational effect vector is computed by:

$$g_i(\mathbf{q}) = - \sum_{j=1}^n m_j \frac{\partial (\mathbf{g}_E^T(0) \mathbf{r}_{C_j})}{\partial q_i}, \quad (2.40)$$

where \mathbf{g}_E is the gravity acceleration vector.

In order to calculate the motor forces/torques, the friction effects related to the motor side \mathbf{h} is used instead of the friction effects related to the joint side \mathbf{h}_g . Each entry h_i of \mathbf{h} is the sum of the viscous friction torque h_{v_i} and the COULOMB friction torque h_{c_i}

$$h_i = \underbrace{r_{v_i} \dot{q}_{M_i}}_{h_{v_i}} + \underbrace{r_{c_i} \text{sign}(\dot{q}_{M_i})}_{h_{c_i}}, \quad (2.41)$$

where r_{v_i} and r_{c_i} are the viscous and COULOMB friction coefficients, respectively. Considering the gear reduction ratio ρ_i , the velocity of the motor before the gear train \dot{q}_{M_i} is given by

$$\dot{q}_{M_i} = \rho_i \dot{q}_i. \quad (2.42)$$

The gear reduction ratio of all joints can be grouped in the matrix

$$\boldsymbol{\rho}' = \text{diag} \left(\frac{1}{\rho_1}, \dots, \frac{1}{\rho_n} \right). \quad (2.43)$$

In addition, the inertias of the complete drive trains ${}_{(M_i)}I_{zz}^{(M_i)}$ are grouped in

$$\mathbf{B}_m = \text{diag} \left(\rho_1 {}_{(M_1)}I_{zz}^{(M_1)}, \dots, \rho_n {}_{(M_n)}I_{zz}^{(M_n)} \right), \quad (2.44)$$

to determine the motor forces/torques vector $\boldsymbol{\tau}_M$ through

$$\boldsymbol{\tau}_M = \boldsymbol{\rho}' \left(\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) - \mathbf{J}^T(\mathbf{q}) \mathcal{F} \right) + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}_m \ddot{\mathbf{q}}. \quad (2.45)$$

3 Structural synthesis

The first step of the CRS is obtaining a set of architectures (here also called structures) that are able to perform a required task, i. e. with the necessary EE DOF. Furthermore, the geometric parameters that have no influence on the EE DOF need to be determined for each architecture. These parameters are used as optimization parameters in the dimensional synthesis of each architecture. The determination of the set of task-suitable architectures as well as their corresponding optimization parameters is the focus of the present chapter. Section 3.1 deals with the generation of all suitable structures described using the DH convention. Similar mechanisms (called isomorphisms) have to be detected within the set of all suitable architectures (section 3.2) and are grouped in order to reduce the number of structures being considered. In the same step, additional geometric parameters used as optimization parameters are identified. An example of the synthesis as well as some results comparing the obtained structures with and without the detection of isomorphisms are shown at the end of the chapter in section 3.3. Some of the results introduced in this chapter were previously published in [RKO15b, RKO15a].

The general procedure of the structural synthesis, shown in figure 3.1, begins with generating structures having 1 DOF, which correspond to the first links of the suitable architectures. Then, the isomorphisms in this set are detected and grouped in order to find a reduced set of suitable architectures with 1 DOF. In a second stage, single links are added to the first mechanisms to generate suitable architectures with 2 DOF. The isomorphisms are detected again in this set to form a reduced set with 2 DOF. The process is repeated until the required number and type of DOF are achieved.

3.1 Generation of suitable architectures

The required DOF of a given task can be defined through the required motion vector

$$\boldsymbol{\xi}_{\text{req}} = \begin{pmatrix} \boldsymbol{v}_{\text{req}_n} \\ \boldsymbol{\omega}_{\text{req}_n} \end{pmatrix} = (\xi_{\text{req}_1}, \xi_{\text{req}_2}, \xi_{\text{req}_3}, \xi_{\text{req}_4}, \xi_{\text{req}_5}, \xi_{\text{req}_6})^T \in \mathbb{R}^6, \quad (3.1)$$

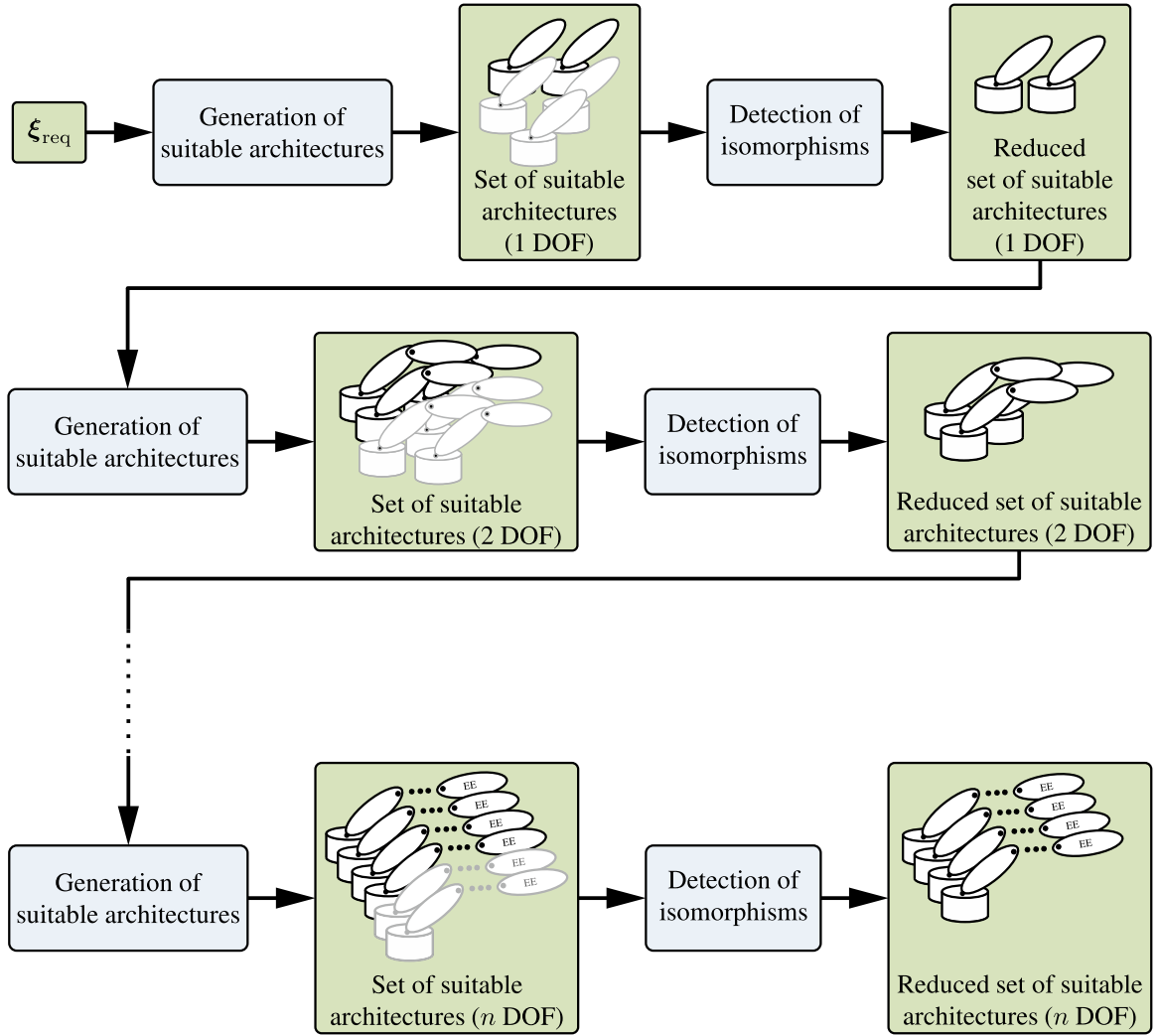


Figure 3.1: Proposed approach for the structural synthesis

where the first three entries describe translational motions of the EE along x -, y -, and z -axis, respectively. The last three entries represent rotational motions about x -, y -, and z -axis. The vector ξ_{req} corresponds to the motion vector ξ defined in (2.18):

$$\xi = \begin{pmatrix} \mathbf{v}_n \\ \boldsymbol{\omega}_n \end{pmatrix} = (\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6)^T. \quad (3.2)$$

Elements of ξ_{req} being zero ($\xi_{\text{req}_i} = 0$) indicate that there is no required motion in this direction. For instance, a manipulator with the required motion vector

$$\xi_{\text{req}} = (\xi_{\text{req}_1}, \xi_{\text{req}_2}, \xi_{\text{req}_3}, 0, 0, \xi_{\text{req}_6})^T \quad (3.3)$$

possesses at the EE three translational DOF and one rotational DOF about the z -axis. The total required DOF in total are the number of $\xi_{\text{req}_i} \neq 0$, which is four in this case.

Since the manipulators are described using the DH convention, the first joint axis is always parallel to the z -axis and, hence, all structures will have a motion along or about it. Sometimes this motion is not needed. Therefore, the required motion vector can be modified using:

$$\xi_{\text{req}}^{(z)} = \xi_{\text{req}}, \quad (3.4)$$

$$\xi_{\text{req}}^{(y)} = \begin{pmatrix} \mathbf{R}_x^T(-\pi/2) & \mathbf{0}_{[3 \times 3]} \\ \mathbf{0}_{[3 \times 3]} & \mathbf{R}_x^T(-\pi/2) \end{pmatrix} \xi_{\text{req}}, \quad \xi_{\text{req}}^{(x)} = \begin{pmatrix} \mathbf{R}_y^T(\pi/2) & \mathbf{0}_{[3 \times 3]} \\ \mathbf{0}_{[3 \times 3]} & \mathbf{R}_y^T(\pi/2) \end{pmatrix} \xi_{\text{req}}, \quad (3.5)$$

where $\xi_{\text{req}}^{(z)}$, $\xi_{\text{req}}^{(y)}$, and $\xi_{\text{req}}^{(x)}$ are the required motion vectors for manipulators whose first joint axes are parallel to z -, y -, and x -axis, respectively (see figure 3.2). The matrices $\mathbf{R}_x(-\pi/2)$ and $\mathbf{R}_y(\pi/2)$ are rotation matrices for a rotation angle of $-\pi/2$ and $\pi/2$ about x - and y -axis:

$$\mathbf{R}_x(-\pi/2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}, \quad \mathbf{R}_y(\pi/2) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}. \quad (3.6)$$

This modification (rotation) of the required motion vector allows, additionally, for considering several orientations of the robot base in the subsequent optimization (see section 4.3).

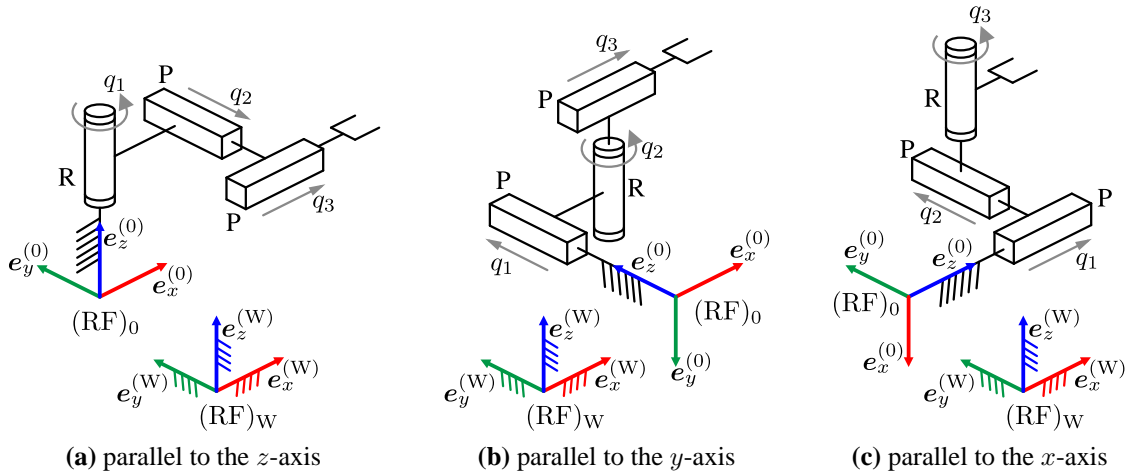


Figure 3.2: Possible orientations of the first joint axis for exemplary architectures.

The suitable architectures are generated from a discrete set of DH parameters (see section 2.3). Since only P and R joints are considered, $d_i = q_i$ if the joint i is a P joint and $\theta_i = q_i$ in the case of a R joint (see figure 2.3 and figure 2.4). Furthermore, each DH parameter is limited to two possible values:

$$\theta_i = \{0, \pi/2\}, \quad d_i = \{0, d_i^*\}, \quad a_i = \{0, a_i^*\}, \quad \alpha_i = \{0, \pi/2\}. \quad (3.7)$$

The notation $d_i = d_i^*$ as well as $a_i = a_i^*$ indicates that the parameters d_i and a_i are different from zero. Since in real robots subsequent joint axes are usually parallel or perpendicular, the values of α_i are chosen to reflect this fact. Parameter $\alpha_i = 0$ represents parallel axes meanwhile $\alpha_i = \pi/2$ represents perpendicular axes. The case $\alpha_i = -\pi/2$ is not considered because the results are the same as using $\alpha_i = \pi/2$. All 16 possible parameter combinations for the i -th link are enumerated in table 3.1, 8 combinations for a P joint and 8 combinations for a R joint.

These possible links will be used to generate the architectures. The suitability of an architecture is evaluated through the following two conditions:

1. the entries of the required motion vector ξ_{req} that are zero must also be zero in the motion vector of the evaluated architecture ξ , i. e. if $\xi_{\text{req}_l} = 0$ then $\xi_l = 0$ for the same l

$$\xi_l = 0 \quad \forall l = \{1 \dots 6 \mid \xi_{\text{req}_l} = 0\}, \quad (3.8)$$

2. the rank of the JACOBIAN \mathbf{J} for the evaluated architecture must be equal to the number of DOF in each iteration k_{dof} (see figure 3.1), i. e.

$$\text{rank}(\mathbf{J}) = k_{\text{dof}}. \quad (3.9)$$

These conditions have to be fulfilled for any \mathbf{q} and $\dot{\mathbf{q}}$. Hence, they are evaluated in symbolic form.

Table 3.1: List of possible DH parameter combinations for the i -th link

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	0	q_i	0	0	P	$\pi/2$	q_i	0	0	R	q_i	0	0	0	R	q_i	d_i^*	0	0
P	0	q_i	0	$\pi/2$	P	$\pi/2$	q_i	0	$\pi/2$	R	q_i	0	0	$\pi/2$	R	q_i	d_i^*	0	$\pi/2$
P	0	q_i	a_i^*	0	P	$\pi/2$	q_i	a_i^*	0	R	q_i	0	a_i^*	0	R	q_i	d_i^*	a_i^*	0
P	0	q_i	a_i^*	$\pi/2$	P	$\pi/2$	q_i	a_i^*	$\pi/2$	R	q_i	0	a_i^*	$\pi/2$	R	q_i	d_i^*	a_i^*	$\pi/2$

3.2 Detection of isomorphisms

The concept of isomorphisms has been treated in the structural synthesis of mechanisms by means of graph theory to identify different graphs that describe the same mechanism [Mer06, Tsa01]. In this work, different set of DH parameters describing the same architecture are called isomorphisms. An example of such isomorphisms is shown in figure 3.3. Regarding the architecture shown on the left hand of figure 3.3, the parameters of the first link are $\theta_1 = 0$ and $a_1 = a_1^*$, while in case of the architecture given on the right hand $\theta_1 = \pi/2$ and $a_1 = 0$. As it can be seen, θ_1 and a_1 can take any value without influencing the motion directions of the EE. Therefore, both

manipulators can be considered as particular cases of the same architecture, in which θ_1 and a_1 can take any value including zero. The variables fulfilling this condition will be labelled $\theta_1, \theta_2, \dots, d_1, d_2, \dots, a_1, a_2, \dots$ or $\alpha_1, \alpha_2, \dots$ in the description of the structures (e. g. appendix A and appendix B) As part of the structural synthesis, the isomorphisms are gathered in groups that can be represented by a common DH parameters matrix in order to reduce the computational effort during the optimization, and to determine the parameters that can be used as optimization parameters.

During the detection of isomorphisms, the influence of each DH parameter on the motion directions of the EE is evaluated using the conditions explained in the following sections. These are based on the linear independence between the required motion directions and the direction each DH parameter is measured along.

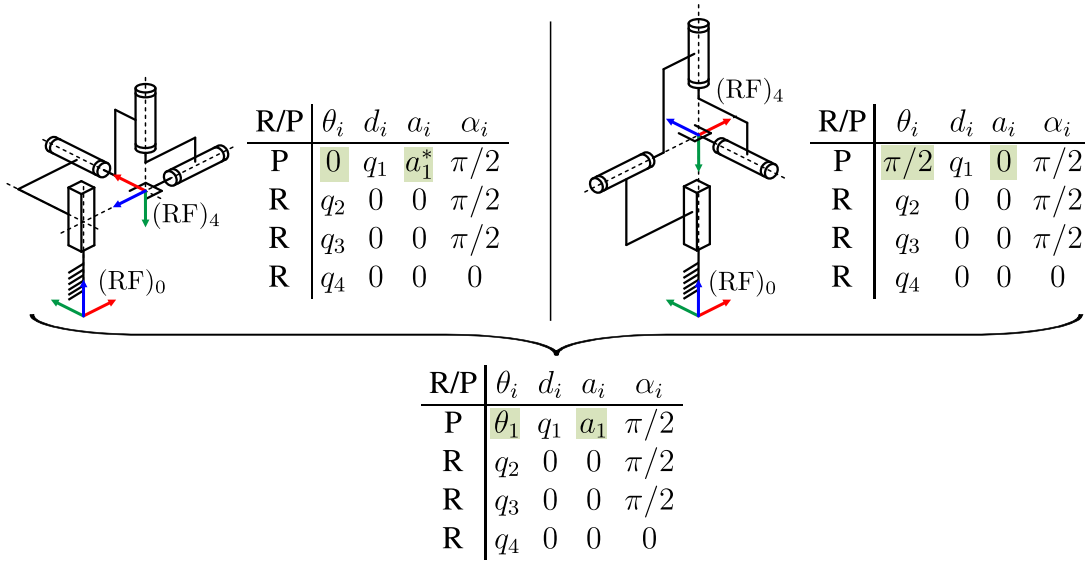


Figure 3.3: Example of two isomorphisms with $\xi_{\text{req}} = (0, 0, \xi_{\text{req}_3}, \xi_{\text{req}_4}, \xi_{\text{req}_5}, \xi_{\text{req}_6})^T$

In the next sections, several conditions to establish this linear independence are analyzed for the link j of figure 3.4. It is worth noting that subindex j represents the link under consideration, while i is used as index in the summation to indicate any other link.

3.2.1 Conditions for the link offset (d)

To analyze the influence of the parameter d_j (j -th link in figure 3.4) on the EE motion direction, equation (2.18) can be rewritten as

$$\xi = \begin{pmatrix} \mathbf{v}_n \\ \boldsymbol{\omega}_n \end{pmatrix} = \sum_{i=1}^n \left[\dot{\theta}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \times \mathbf{r}_{i-1,n} \\ \mathbf{e}_z^{(i-1)} \end{pmatrix} + \dot{d}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \right], \quad (3.10)$$

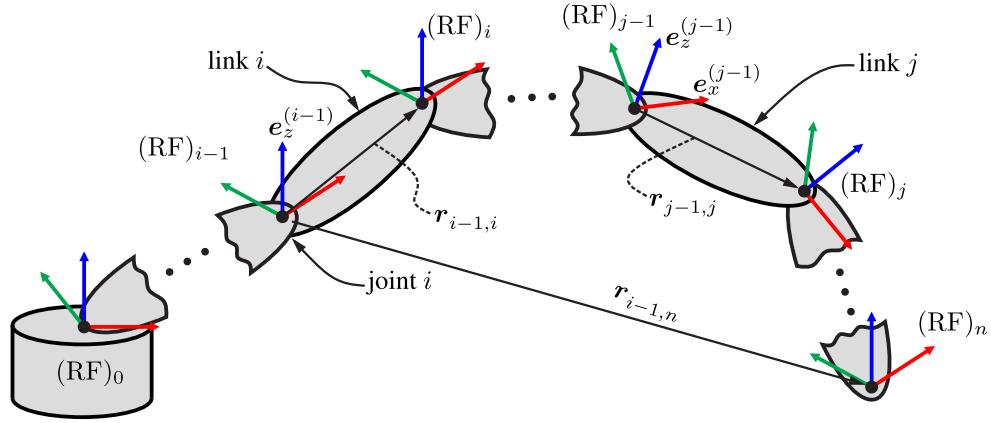


Figure 3.4: Location of the reference frames for the detection of isomorphisms

where $e_z^{(i-1)}$ and $r_{i-1,n}$ are defined in section 2.4 and depicted in figure 3.4. For a R joint, $\dot{\theta}_i = \dot{q}_i$ and $\dot{d}_i = 0$, while for a P joint $\dot{\theta}_i = 0$ and $\dot{d}_i = \dot{q}_i$. The position vector $r_{i-1,n}$ can be expressed as the sum of the relative position vector of each link

$$r_{i-1,n} = r_{i-1,i} + r_{i,i+1} + \dots + r_{n-1,n}, \quad (3.11)$$

and replaced in (3.10) obtaining

$$\xi = \sum_{i=1}^n \left[\dot{\theta}_i \begin{pmatrix} e_z^{(i-1)} \times (r_{i-1,i} + r_{i,i+1} + \dots + r_{n-1,n}) \\ e_z^{(i-1)} \end{pmatrix} + \dot{d}_i \begin{pmatrix} e_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \right]. \quad (3.12)$$

The term accompanying θ_i can be divided into two parts: $i \leq j$ and $i > j$, then

$$\begin{aligned} & \dot{\theta}_i \begin{pmatrix} e_z^{(i-1)} \times (r_{i-1,i} + \dots + r_{n-1,n}) \\ e_z^{(i-1)} \end{pmatrix} \\ &= \dot{\theta}_i \begin{pmatrix} e_z^{(i-1)} \times (r_{i-1,i} + \dots + r_{j-1,j}) \\ \mathbf{0} \end{pmatrix} + \dot{\theta}_i \begin{pmatrix} e_z^{(i-1)} \times (r_{j,j+1} + \dots + r_{n-1,n}) \\ e_z^{(i-1)} \end{pmatrix}. \end{aligned} \quad (3.13)$$

Combining (3.12) and (3.13) leads to

$$\begin{aligned} \xi &= \sum_{i=1}^j \left[\dot{\theta}_i \begin{pmatrix} e_z^{(i-1)} \times (r_{i-1,i} + \dots + r_{j-1,j}) \\ \mathbf{0} \end{pmatrix} \right. \\ &\quad \left. + \dot{\theta}_i \begin{pmatrix} e_z^{(i-1)} \times (r_{j,j+1} + \dots + r_{n-1,n}) \\ e_z^{(i-1)} \end{pmatrix} + \dot{d}_i \begin{pmatrix} e_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \right] \\ &\quad + \sum_{i=j+1}^n \left[\dot{\theta}_i \begin{pmatrix} e_z^{(i-1)} \times (r_{i-1,i} + \dots + r_{n-1,n}) \\ e_z^{(i-1)} \end{pmatrix} + \dot{d}_i \begin{pmatrix} e_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \right]. \end{aligned} \quad (3.14)$$

From this equation, $\mathbf{r}_{j-1,j}$ can be factorized:

$$\begin{aligned} \boldsymbol{\xi} = & \sum_{i=1}^{j-1} \left[\begin{pmatrix} \left(\dot{\theta}_1 \mathbf{e}_z^{(0)} + \dots + \dot{\theta}_i \mathbf{e}_z^{(i-1)} \right) \times \mathbf{r}_{i-1,i} \\ \mathbf{0} \end{pmatrix} \right] + \begin{pmatrix} \left(\dot{\theta}_1 \mathbf{e}_z^{(0)} + \dots + \dot{\theta}_j \mathbf{e}_z^{(j-1)} \right) \times \mathbf{r}_{j-1,j} \\ \mathbf{0} \end{pmatrix} \\ & + \sum_{i=1}^j \left[\dot{\theta}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \times (\mathbf{r}_{j,j+1} + \dots + \mathbf{r}_{n-1,n}) \\ \mathbf{e}_z^{(i-1)} \end{pmatrix} + d_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \right] \\ & + \sum_{i=j+1}^n \left[\dot{\theta}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \times (\mathbf{r}_{i-1,i} + \dots + \mathbf{r}_{n-1,n}) \\ \mathbf{e}_z^{(i-1)} \end{pmatrix} + d_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \right]. \end{aligned} \quad (3.15)$$

If the j -th joint is a prismatic joint, d_j corresponds to the joint coordinate, consequently, the j -th link is not evaluated with respect to d_j (see table 3.2 at the end of this section). In the case of a rotational joint, $\dot{d}_j = 0$. Moreover, in (3.15), $\mathbf{r}_{j-1,j}$ represents the position vector from the origin of $(\text{RF})_{j-1}$ to the origin of $(\text{RF})_j$, which can be calculated using the parameters a_j and d_j of the j -th link:

$$\mathbf{r}_{j-1,j} = \underbrace{{}^0\mathbf{R}_1 \dots {}^{j-2}\mathbf{R}_{j-1} {}^{j-1}\mathbf{R}_{z_{j-1}}(\theta_j)}_{a_j \mathbf{e}_x^{(j)}} \begin{pmatrix} a_j \\ 0 \\ 0 \end{pmatrix} + \underbrace{{}^0\mathbf{R}_1 \dots {}^{j-2}\mathbf{R}_{j-1} {}^{j-1}\mathbf{R}_{z_{j-1}}(\theta_j)}_{d_j \mathbf{e}_z^{(j-1)}} \begin{pmatrix} 0 \\ 0 \\ d_j \end{pmatrix}. \quad (3.16)$$

Additionally, the angular velocity $\boldsymbol{\omega}_j$ of the j -th link is

$$\boldsymbol{\omega}_j = \dot{\theta}_1 \mathbf{e}_z^{(0)} + \dots + \dot{\theta}_j \mathbf{e}_z^{(j-1)}. \quad (3.17)$$

Using (3.16) and (3.17), the term containing $\mathbf{r}_{j-1,j}$ in (3.15) can be rewritten as:

$$\left(\dot{\theta}_1 \mathbf{e}_z^{(0)} + \dots + \dot{\theta}_j \mathbf{e}_z^{(j-1)} \right) \times \mathbf{r}_{j-1,j} = \boldsymbol{\omega}_j \times a_j \mathbf{e}_x^{(j)} + \boldsymbol{\omega}_j \times d_j \mathbf{e}_z^{(j-1)}. \quad (3.18)$$

Regarding (3.18), the product

$$\boldsymbol{\vartheta}_j = \boldsymbol{\omega}_j \times d_j \mathbf{e}_z^{(j-1)} \in \mathbb{R}^3 \quad (3.19)$$

can be considered to establish two cases in which the value of d_j has no influence on the motion direction of the EE. Firstly, if $\boldsymbol{\omega}_j$ is parallel to $\mathbf{e}_z^{(j-1)}$, $\boldsymbol{\vartheta}_j = \mathbf{0}$ regardless of the value of d_j . Secondly, the components of $\boldsymbol{\vartheta}_j$ are included in $\boldsymbol{\xi}_{\text{req}}$, i. e. if $\xi_{\text{req}_l} = 0$ for any $l = 1 \dots 3$, the l -th element of $\boldsymbol{\vartheta}_j$ must also be zero. If one of these conditions is fulfilled, the parameter d_j of the j -th link can take any value without changing the motion direction of the EE (see table 3.2).

3.2.2 Conditions for the link length (a)

The influence of the link length a_j on the EE motion can be studied using the figure 3.4 again. If the j -th joint is a prismatic joint, $\dot{\theta}_j = 0$. In this case, (3.15) can be rewritten as

$$\begin{aligned} \xi = \sum_{i=1}^{j-1} & \left[\begin{pmatrix} \left(\dot{\theta}_1 \mathbf{e}_z^{(0)} + \dots + \dot{\theta}_i \mathbf{e}_z^{(i-1)} \right) \times \mathbf{r}_{i-1,i} \\ \mathbf{0} \end{pmatrix} \right] \\ & + \begin{pmatrix} \left(\dot{\theta}_1 \mathbf{e}_z^{(0)} + \dots + \dot{\theta}_{j-1} \mathbf{e}_z^{(j-2)} \right) \times \mathbf{r}_{j-1,j} \\ \mathbf{0} \end{pmatrix} + \dot{d}_j \begin{pmatrix} \mathbf{e}_z^{(j-1)} \\ \mathbf{0} \end{pmatrix} \\ & + \sum_{i=1}^{j-1} \left[\dot{\theta}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \times (\mathbf{r}_{j,j+1} + \dots + \mathbf{r}_{n-1,n}) \\ \mathbf{e}_z^{(i-1)} \end{pmatrix} + \dot{d}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \right] \\ & + \sum_{i=j+1}^n \left[\dot{\theta}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \times (\mathbf{r}_{i-1,i} + \dots + \mathbf{r}_{n-1,n}) \\ \mathbf{e}_z^{(i-1)} \end{pmatrix} + \dot{d}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \right]. \quad (3.20) \end{aligned}$$

The term containing $\mathbf{r}_{j-1,j}$ can be reformulated using a similar simplification as presented in section 3.2.1:

$$\left(\dot{\theta}_1 \mathbf{e}_z^{(0)} + \dots + \dot{\theta}_{j-1} \mathbf{e}_z^{(j-2)} \right) \times \mathbf{r}_{j-1,j} = \boldsymbol{\omega}_{j-1} \times a_j \mathbf{e}_x^{(j)} + \boldsymbol{\omega}_{j-1} \times d_j \mathbf{e}_z^{(j-1)}. \quad (3.21)$$

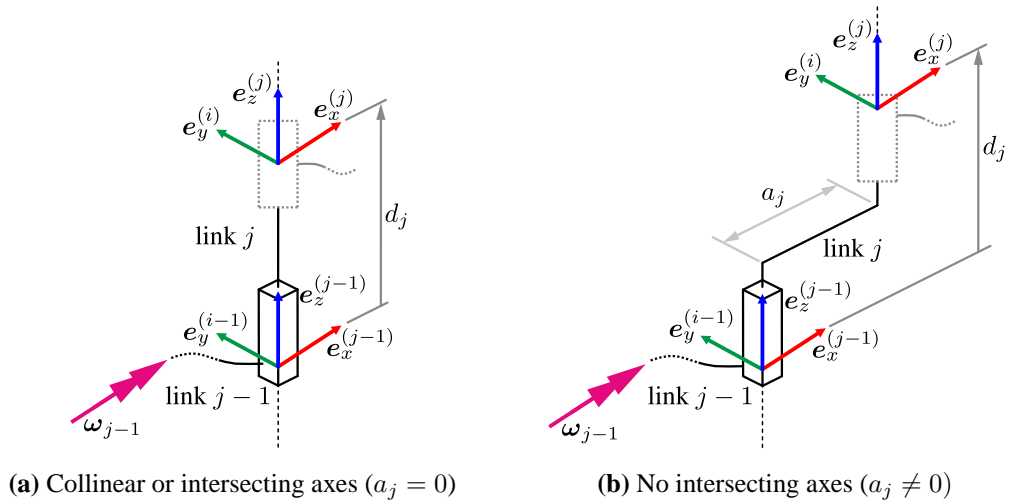


Figure 3.5: Possible link length in a link with P joint

Considering the product

$$\boldsymbol{\vartheta}_j = \boldsymbol{\omega}_{j-1} \times a_j \mathbf{e}_x^{(j)} \in \mathbb{R}^3 \quad (3.22)$$

from (3.21), the value of a_j has no influence on the motion direction of the EE in two cases. Firstly, if $\boldsymbol{\omega}_{j-1}$ is parallel to $\mathbf{e}_x^{(j)}$, $\boldsymbol{\vartheta}_j = \mathbf{0}$ regardless of the value of a_j (see figure 3.5). Secondly,

the components of ϑ_j are included in ξ_{req} , i. e. if $\xi_{\text{req}_l} = 0$ for any $l = 1 \dots 3$, the l -th entry of ϑ_j must also be zero. If one of these conditions is fulfilled, the parameter a_j of the j -th link can take any value without changing the motion direction of the EE (see table 3.2).

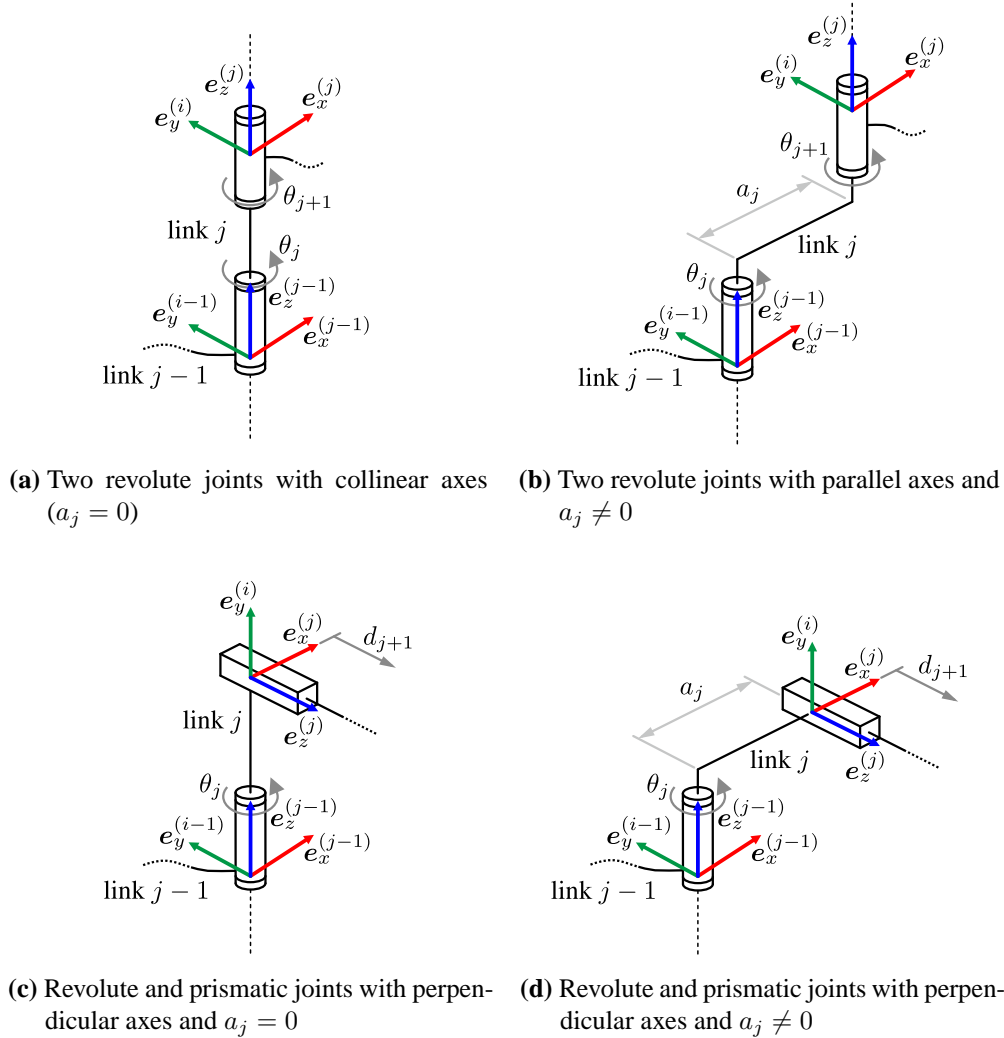


Figure 3.6: Possible configurations of two consecutive joints being the first joint a R joint

The second case to be studied is when the j -th joint is a revolute joint. In this case, $\dot{\theta}_j \neq 0$, $\dot{d}_j = 0$, and, therefore, equation (3.18) can be considered again. The product $\omega_j \times a_j e_x^{(j)}$ is different from zero because $e_z^{(j-1)}$ is perpendicular to $e_x^{(j)}$ (see figure 3.6). This relation can be observed comparing the vectors $(a_j \cos(\theta_j), a_j \sin(\theta_j), 0)^T$ and $(0, 0, 1)^T$ in (3.23) and (3.24), which are perpendicular as well:

$$a_j e_x^{(j)} = {}^0\mathbf{R}_1 \dots {}^{j-2}\mathbf{R}_{j-1} {}^{j-1}\mathbf{R}_{z_{j-1}}(\theta_j) \begin{pmatrix} a_j \\ 0 \\ 0 \end{pmatrix} = {}^0\mathbf{R}_1 \dots {}^{j-2}\mathbf{R}_{j-1} \begin{pmatrix} a_j \cos(\theta_j) \\ a_j \sin(\theta_j) \\ 0 \end{pmatrix}, \quad (3.23)$$

$$\mathbf{e}_z^{(j-1)} = {}^0\mathbf{R}_1 \dots {}^{j-2}\mathbf{R}_{j-1} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \quad (3.24)$$

It is important to underline that for $a_j = 0$ the j -th joint's motion leads to a rotational motion around the vector $\mathbf{e}_z^{(j-1)}$ only. In contrast, if $a_j \neq 0$ the joint motion results in an additional translational motion in a plane perpendicular to $\mathbf{e}_z^{(j-1)}$. The aforementioned effect of the value a_j is clarified in figure 3.6. If the axes of both consecutive parallel revolute joints are collinear ($a_j = 0$ and $\alpha_j = 0$), the kinematic chain has just one redundant DOF (figure 3.6(a)). In contrast, if $a_j \neq 0$, the mechanism has two independent DOF (figure 3.6(b)). On the other hand, if there is a perpendicular prismatic joint after the revolute joint (figure 3.6(c) and figure 3.6(d)), the amount of independent DOF is affected not only by a_j . The DOF of the EE also depend, in this case, on the parameters of the $(j + 1)$ -th link and a_j does not necessarily have to be different from zero. This condition, i. e. if the parameter a_j has (or does not have) to be different from zero, can be checked analyzing the rank of the first $j + 1$ columns of the JACOBIAN \mathbf{J} . Let

$$\mathbf{J}_{j+1}^{(a_j=0)} = \begin{pmatrix} \dot{\mathbf{j}}_{t_1}^{(a_j=0)} & \dots & \dot{\mathbf{j}}_{t_{j+1}}^{(a_j=0)} \\ \dot{\mathbf{j}}_{r_1} & \dots & \dot{\mathbf{j}}_{r_{j+1}} \end{pmatrix} \quad (3.25)$$

be the JACOBIAN matrix of the manipulator including the $(j + 1)$ -th link with $a_j = 0$ and

$$\mathbf{J}_{j+1}^{(a_j=a^*)} = \begin{pmatrix} \dot{\mathbf{j}}_{t_1}^{(a_j=a^*)} & \dots & \dot{\mathbf{j}}_{t_{j+1}}^{(a_j=a^*)} \\ \dot{\mathbf{j}}_{r_1} & \dots & \dot{\mathbf{j}}_{r_{j+1}} \end{pmatrix} \quad (3.26)$$

be the same JACOBIAN matrix with $a_j \neq 0$, the following conditions can be inferred:

- if $\text{rank} \left(\mathbf{J}_{j+1}^{(a_j=0)} \right) < \text{rank} \left(\mathbf{J}_{j+1}^{(a_j=a^*)} \right)$ then a_j must be different from zero,
- if $\text{rank} \left(\mathbf{J}_{j+1}^{(a_j=0)} \right) = \text{rank} \left(\mathbf{J}_{j+1}^{(a_j=a^*)} \right)$ then a_j can take any value including zero.

As well as in the case of prismatic joints, if the components of ϑ_j in (3.22) are included in $\boldsymbol{\xi}_{\text{req}}$, the value of a_j has no influence on the motion direction of the EE (see table 3.2).

3.2.3 Conditions for the link twist (α)

In order to analyze the link twist α_j , the calculation of the EE velocity can be separated into two parts: one considering the links 1 to j , the other including the links $j + 1$ to n (see figure 3.4). Equation (3.10) can therefore be rewritten as

$$\xi = \underbrace{\sum_{i=1}^j \left[\dot{\theta}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \times \mathbf{r}_{i-1,n} \\ \mathbf{e}_z^{(i-1)} \end{pmatrix} + \dot{d}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \right]}_{(\mathbf{v}_j^T, \boldsymbol{\omega}_j^T)^T} + \sum_{i=j+1}^n \left[\dot{\theta}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \times \mathbf{r}_{i-1,n} \\ \mathbf{e}_z^{(i-1)} \end{pmatrix} + \dot{d}_i \begin{pmatrix} \mathbf{e}_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \right]. \quad (3.27)$$

The first part includes the first j joints. Their axis orientations are not influenced by α_j (see figure 3.4). The second part consists of the last $n - j$ joints. Their axis orientations are dependent on α_j .

The vector $\mathbf{e}_z^{(i-1)}$ in (3.27) represents the orientation of the i -th joint axis. This can be expressed for $i > j$ as

$$\mathbf{e}_z^{(i-1)} = {}^0\mathbf{R}_{j-1} {}^{j-1}\mathbf{R}_j {}^j\mathbf{R}_{i-1} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = {}^0\mathbf{R}_{j-1} {}^{j-1}\mathbf{R}_j (j)\mathbf{e}_z^{(i-1)}. \quad (3.28)$$

As the orientation of the j -th reference frame with respect to the $(j - 1)$ -th reference frame results from the rotation θ_j about $\mathbf{e}_z^{(j-1)}$ followed by a rotation α_j about the new axis $\mathbf{e}_x^{(j-1)*}$ (figure 3.7), the rotation matrix

$${}^{j-1}\mathbf{R}_j = {}^{j-1}\mathbf{R}_{z_{j-1}}(\theta_j) (j-1)^*\mathbf{R}_{x_{(j-1)^*}}(\alpha_j) \quad (3.29)$$

depends on α_j .

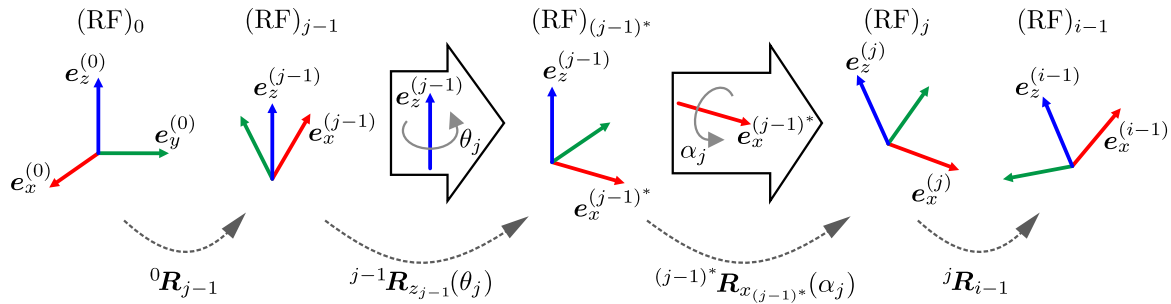


Figure 3.7: Rotation of the reference frames due to θ_j and α_j

Hence, equation (3.27) can be rewritten using (3.28) in order to factorize the product ${}^0\mathbf{R}_{j-1}{}^{j-1}\mathbf{R}_j$:

$$\begin{aligned} \boldsymbol{\xi} = & \begin{pmatrix} \mathbf{v}_j \\ \boldsymbol{\omega}_j \end{pmatrix} + \begin{pmatrix} {}^0\mathbf{R}_{j-1}{}^{j-1}\mathbf{R}_j & \mathbf{0}_{[3 \times 3]} \\ \mathbf{0}_{[3 \times 3]} & {}^0\mathbf{R}_{j-1}{}^{j-1}\mathbf{R}_j \end{pmatrix} \cdots \\ & \cdots \left(\sum_{i=j+1}^n \left[\dot{\theta}_i \begin{pmatrix} (j)\mathbf{e}_z^{(i-1)} \times \mathbf{r}_{i-1,n} \\ (j)\mathbf{e}_z^{(i-1)} \end{pmatrix} + \dot{d}_i \begin{pmatrix} (j)\mathbf{e}_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \right] \right). \end{aligned} \quad (3.30)$$

Furthermore, the JACOBIAN $(j)\mathbf{J}_{j+1,n}$ for the last $(n-j)$ joints can be defined as

$$(j)\mathbf{J}_{j+1,n} = \left((j)\mathbf{J}_{j+1}, (j)\mathbf{J}_{j+2}, \dots, (j)\mathbf{J}_n \right) \in \mathbb{R}^{6 \times (n-j)}, \quad (3.31)$$

being $(j)\mathbf{J}_i$ determined (with $i > j$) as

$$(j)\mathbf{J}_i = \begin{pmatrix} (j)\mathbf{e}_z^{(i-1)} \times \mathbf{r}_{i-1,n} \\ (j)\mathbf{e}_z^{(i-1)} \end{pmatrix} \quad (3.32)$$

for a rotational joint and

$$(j)\mathbf{J}_i = \begin{pmatrix} (j)\mathbf{e}_z^{(i-1)} \\ \mathbf{0} \end{pmatrix} \quad (3.33)$$

for a prismatic joint. The first three rows of $(j)\mathbf{J}_{j+1,n}$ represent the effect of the last $(n-j)$ joint rates on the linear velocity of the EE:

$$(j)\mathbf{J}_{j+1,n}^t = \begin{pmatrix} \mathbf{I}_{[3 \times 3]} & \mathbf{0}_{[3 \times 3]} \end{pmatrix} (j)\mathbf{J}_{j+1,n}. \quad (3.34)$$

The last three rows define the effect on the angular velocity:

$$(j)\mathbf{J}_{j+1,n}^r = \begin{pmatrix} \mathbf{0}_{[3 \times 3]} & \mathbf{I}_{[3 \times 3]} \end{pmatrix} (j)\mathbf{J}_{j+1,n}. \quad (3.35)$$

Therefore, equation (3.34) and (3.35) can be used to write

$$\begin{pmatrix} (j)\mathbf{v}_{j+1,n} \\ (j)\boldsymbol{\omega}_{j+1,n} \end{pmatrix} = \begin{pmatrix} (j)\mathbf{J}_{j+1,n}^t \\ (j)\mathbf{J}_{j+1,n}^r \end{pmatrix} \begin{pmatrix} \dot{q}_{j+1} \\ \vdots \\ \dot{q}_n \end{pmatrix}. \quad (3.36)$$

Finally, combining (3.30) and (3.36) leads to

$$\begin{aligned}\xi &= \begin{pmatrix} \mathbf{v}_j \\ \boldsymbol{\omega}_j \end{pmatrix} + \begin{pmatrix} {}^0\mathbf{R}_{j-1} {}^{j-1}\mathbf{R}_j & \mathbf{0}_{[3 \times 3]} \\ \mathbf{0}_{[3 \times 3]} & {}^0\mathbf{R}_{j-1} {}^{j-1}\mathbf{R}_j \end{pmatrix} \begin{pmatrix} (j)\mathbf{J}_{j+1,n}^t \\ (j)\mathbf{J}_{j+1,n}^r \end{pmatrix} \begin{pmatrix} \dot{q}_{j+1} \\ \vdots \\ \dot{q}_n \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{v}_j \\ \boldsymbol{\omega}_j \end{pmatrix} + \begin{pmatrix} {}^0\mathbf{R}_{j-1} {}^{j-1}\mathbf{R}_{z_{j-1}}(\theta_j) {}^{j-1*}\mathbf{R}_{x_{j-1}*}(\alpha_j) (j)\mathbf{v}_{j+1,n} \\ {}^0\mathbf{R}_{j-1} {}^{j-1}\mathbf{R}_{z_{j-1}}(\theta_j) {}^{j-1*}\mathbf{R}_{x_{j-1}*}(\alpha_j) (j)\boldsymbol{\omega}_{j+1,n} \end{pmatrix}. \quad (3.37)\end{aligned}$$

It can be inferred from (3.37) that if $\text{rank}((j)\mathbf{J}_{j+1,n}^t) = \{0, 3\}$ and $\text{rank}((j)\mathbf{J}_{j+1,n}^r) = \{0, 3\}$, the motion directions of the EE are not affected by the rotation angle α_j . When $\text{rank}((j)\mathbf{J}_{j+1,n}^t) = 3$ or $\text{rank}((j)\mathbf{J}_{j+1,n}^r) = 3$, the linear (or angular) velocity generated by the last $(n - j)$ joints leads to a motion of the EE in the x -, y -, and z -direction. This condition is not affected by a variation of α_j . Furthermore, this variation has no effect on the motion directions of the EE if $\text{rank}((j)\mathbf{J}_{j+1,n}^t) = 0$ or $\text{rank}((j)\mathbf{J}_{j+1,n}^r) = 0$ (see table 3.2).

3.2.4 Conditions for the joint angle (θ)

If the j -th joint is a revolute joint, the value of θ_j corresponds to the joint coordinate q_j . Hence, for the next analysis, only the case of a prismatic joint is considered. Similar to the case of α (see section 3.2.3), equation (3.37) shows that if $\text{rank}((j)\mathbf{J}_{j+1,n}^t) = \{0, 3\}$ and $\text{rank}((j)\mathbf{J}_{j+1,n}^r) = \{0, 3\}$, the motion directions of the EE are not affected by the rotation angle θ_j .

Additionally, if $\alpha_j = 0$, the matrix ${}^{j-1*}\mathbf{R}_{x_{j-1}*}(\alpha_j) = \mathbf{I}_{[3 \times 3]}$ in (3.29). The vectors \mathbf{e}_z^{j-1} and \mathbf{e}_z^j are parallel (see figure 3.4) and

$${}^{j-1}\mathbf{R}_{z_{j-1}}(\theta_j) (j)\mathbf{e}_z^{i-1} = (j)\mathbf{e}_z^{i-1}. \quad (3.38)$$

Hence, equation (3.28) can be rewritten as

$${}^0\mathbf{R}_{j-1} {}^{j-1}\mathbf{R}_{z_{j-1}}(\theta_j) {}^{j-1*}\mathbf{R}_{x_{j-1}*}(\alpha_j) (j)\mathbf{e}_z^{i-1} = {}^0\mathbf{R}_{j-1} (j)\mathbf{e}_z^{i-1}. \quad (3.39)$$

Inserting (3.39) in (3.37) leads to

$$\xi = \begin{pmatrix} \mathbf{v}_j \\ \boldsymbol{\omega}_j \end{pmatrix} + \begin{pmatrix} {}^0\mathbf{R}_{j-1} & \mathbf{0}_{[3 \times 3]} \\ \mathbf{0}_{[3 \times 3]} & {}^0\mathbf{R}_{j-1} \end{pmatrix} \begin{pmatrix} (j)\mathbf{J}_{j+1,n}^t \\ (j)\mathbf{J}_{j+1,n}^r \end{pmatrix} \begin{pmatrix} \dot{q}_{j+1} \\ \vdots \\ \dot{q}_n \end{pmatrix}. \quad (3.40)$$

From (3.40) can be concluded that if $\alpha_j = 0$ the matrix ${}^{j-1}\mathbf{R}_{z_{j-1}}(\theta_j)$ has no effect on the motion directions of the EE.

Further conditions can be extracted from (3.21). If $\boldsymbol{\omega}_{j-1}$ is parallel to $\mathbf{e}_z^{(j-1)}$ the product

$$\boldsymbol{\omega}_{j-1} \times d_j \mathbf{e}_z^{(j-1)} = \mathbf{0} \quad (3.41)$$

and, therefore, equation (3.21) results to

$$\left(\dot{\theta}_1 \mathbf{e}_z^{(0)} + \dots + \dot{\theta}_{j-1} \mathbf{e}_z^{(j-2)} \right) \times \mathbf{r}_{j-1,j} = \boldsymbol{\omega}_{j-1} \times a_j \mathbf{e}_x^{(j)}. \quad (3.42)$$

Moreover, $\mathbf{e}_z^{(j-1)}$ is perpendicular to $\mathbf{e}_x^{(j-1)}$ and, regarding (3.23) and (3.24), it is also perpendicular to $\mathbf{e}_x^{(j)}$. Consequently, the result

$$\boldsymbol{\vartheta}_j = \boldsymbol{\omega}_{j-1} \times {}^0\mathbf{R}_{j-1} {}^{j-1}\mathbf{R}_{z_{j-1}}(\theta_j) \begin{pmatrix} a_j \\ 0 \\ 0 \end{pmatrix} \quad (3.43)$$

lies on the same plane as the result of

$$\boldsymbol{\vartheta}_j^* = \boldsymbol{\omega}_{j-1} \times {}^0\mathbf{R}_{j-1} \begin{pmatrix} a_j \\ 0 \\ 0 \end{pmatrix}. \quad (3.44)$$

In this case the value of θ_j does not affect the motion direction of the EE.

Table 3.2 summarizes the conditions (see section 3.2.1 to section 3.2.4) to be evaluated for each DH parameter d_j , a_j , α_j , θ_j of the j -th link.

Table 3.2: Summary of the conditions for isomorphisms detection

Parameter	Joint	Condition	Possible values
d_j	R	$\boldsymbol{\omega}_j \times d_j \mathbf{e}_z^{(j-1)} = \mathbf{0} \vee \boldsymbol{\omega}_j \times d_j \mathbf{e}_z^{(j-1)} \in \boldsymbol{\xi}_{\text{req}}$	$d_j \in \mathbb{R}$
a_j	P	$\boldsymbol{\omega}_{j-1} \times a_j \mathbf{e}_x^{(j)} = \mathbf{0} \vee \boldsymbol{\omega}_{j-1} \times a_j \mathbf{e}_x^{(j)} \in \boldsymbol{\xi}_{\text{req}}$	$a_j \in \mathbb{R}$
a_j	R	$\text{rank} \left(\mathbf{J}_{j+1}^{(a_j=0)} \right) < \text{rank} \left(\mathbf{J}_{j+1}^{(a_j=a^*)} \right)$	$a_j \neq 0$
	R	$\boldsymbol{\omega}_{j-1} \times a_j \mathbf{e}_x^{(j)} \in \boldsymbol{\xi}_{\text{req}}$	$a_j \in \mathbb{R}$
α_j	R, P	$\text{rank} \left({}_{(j)}\mathbf{J}_{j+1,n}^t \right) = \{0, 3\} \wedge \text{rank} \left({}_{(j)}\mathbf{J}_{j+1,n}^r \right) = \{0, 3\}$	$\alpha_j \in \mathbb{R}$
θ_j	P	$\text{rank} \left({}_{(j)}\mathbf{J}_{j+1,n}^t \right) = \{0, 3\} \wedge \text{rank} \left({}_{(j)}\mathbf{J}_{j+1,n}^r \right) = \{0, 3\}$	$\theta_j \in \mathbb{R}$
	P	$\alpha_j = 0 \vee \boldsymbol{\omega}_j \times d_j \mathbf{e}_z^{(j-1)} = \mathbf{0}$	

3.3 Obtained structures

The complete algorithm for the structural synthesis is presented in algorithm 3.1. The input is the required motion vector ξ_{req} . Then, $\xi_{\text{req}}^{(y)}$ and $\xi_{\text{req}}^{(x)}$ are calculated. The structural synthesis is carried out for each of these three vectors. In the first step, the possible combinations of DH parameters shown in table 3.1 are used as "possible links". The JACOBIAN and the motion vector of each link are calculated in order to evaluate its suitability as explained in section 3.1. The links that fulfill the two conditions of suitability (set of suitable architectures SA) are then evaluated with regard to the presence of isomorphisms (table 3.2). The resulting set is stored in the reduced set of suitable architectures RSA . The links in the reduced set of suitable architectures are used to generate mechanisms with 2 DOF by adding one link from table 3.1. The suitability and the presence of isomorphisms are evaluated again for each of these architectures. This process is repeated until the desired DOF are achieved. The example below illustrates the generation of one architecture with

$$\xi_{\text{req}} = (0, 0, \xi_{\text{req}_3}, \xi_{\text{req}_4}, \xi_{\text{req}_5}, \xi_{\text{req}_6})^T. \quad (3.45)$$

The four different links shown in figure 3.8 represent the reduced set of suitable architectures obtained from the evaluation of the first links ($k_{\text{dof}} = 1$). Exemplary, the DH parameters of figure 3.8(a) result from the two set of parameters

$$\begin{array}{c|cccc} \text{R/P} & \theta_i & d_i & a_i & \alpha_i \\ \hline \text{P} & 0 & q_1 & a_1^* & 0 \end{array} \quad \text{and} \quad \begin{array}{c|cccc} \text{R/P} & \theta_i & d_i & a_i & \alpha_i \\ \hline \text{P} & 0 & q_1 & 0 & 0 \end{array}, \quad (3.46)$$

which are chosen as suitable architectures (with 1 DOF) from table 3.1. Indeed, the motion vector for these links is

$$\xi = J\dot{q} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \dot{q}_1 = \begin{pmatrix} 0 \\ 0 \\ \dot{q}_1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (3.47)$$

Comparing ξ from (3.47) with ξ_{req} from (3.45), it can be seen that the first two entries are zero in both cases, i. e. $\xi_1 = \xi_{\text{req}_1} = 0$ and $\xi_2 = \xi_{\text{req}_2} = 0$. Hence, both links of (3.46) are considered as suitable structures. Additionally, the $(j-1)$ -th link corresponds to the ground and $\omega_{j-1} = \mathbf{0}$. The condition $\omega_{j-1} \times a_j e_x^{(j)} = \mathbf{0}$ is fulfilled (table 3.2 for a_j). Therefore, both structures are considered as isomorphisms and can be described using the parameters given in figure 3.8(a).

For the structures shown in figures 3.8(b) to 3.8(d) the same procedure was applied.

Algorithm 3.1: Structural synthesis algorithm

```

Input:  $\xi_{\text{req}}$ 
1  $\xi_{\text{req}}^{(z)} = \xi_{\text{req}}$ 
2 calculate  $\xi_{\text{req}}^{(y)}$  and  $\xi_{\text{req}}^{(x)}$  // equation (3.5)
3 foreach  $\xi_{\text{req}}^{(\text{ax})} \in (\xi_{\text{req}}^{(z)}, \xi_{\text{req}}^{(y)}, \xi_{\text{req}}^{(x)})$  do
4   for  $k_{\text{dof}} = 1$  to  $DOF$  do
5      $Links \leftarrow$  possible_links // from table 3.1
6     if  $k_{\text{dof}} = 1$  then
7       foreach  $L \in Links$  do
8          $J \leftarrow$  Jacobian( $L$ ) // equation (2.19)
9          $\xi = J\dot{q}$  // equation (2.18)
10        if  $\xi_l = 0 \forall l = \{1 \dots 6 \mid \xi_{\text{req}_l}^{(\text{ax})} = 0\}$  then // equation (3.8)
11           $SA \leftarrow \{SA, L\}$  // set of suitable architectures
12        end
13      end
14       $RSA \leftarrow$  detect_isomorphisms( $SA$ ) // see section 3.2
15    else
16      foreach  $robot \in RSA$  do
17        foreach  $L \in Links$  do
18           $robot^* \leftarrow$  add_link_to_robot( $robot, L$ )
19           $J \leftarrow$  Jacobian( $robot^*$ ) // equation (2.19)
20           $\xi = J\dot{q}$  // equation (2.18)
21          if  $\xi_l = 0 \forall l = \{1 \dots 6 \mid \xi_{\text{req}_l} = 0\}$  then // equation (3.8)
22            if rank( $J$ ) =  $k_{\text{dof}}$  then // equation (3.9)
23               $SA^* \leftarrow \{SA^*, robot^*\}$  // architectures with  $k_{\text{dof}}$  DOF
24            end
25          end
26        end
27      end
28       $RSA \leftarrow$  detect_isomorphisms( $SA^*$ ) // see section 3.2
29    end
30  end
Output:  $RSA$  // reduced set of suitable architectures
31 end

```

The architectures found in the second stage ($k_{\text{dof}} = 2$) are shown in figure 3.9. For explanation reasons, the architecture shown in figure 3.9(e) is taken as example (see figure 3.10). The considered architecture results from the combination of two isomorphisms:

$$\begin{array}{c|cccc}
 \text{R/P} & \theta_i & d_i & a_i & \alpha_i \\
 \hline
 \text{P} & 0 & q_1 & a_1 & 0 \\
 \text{R} & q_2 & d_2^* & 0 & \pi/2
 \end{array}
 \quad \text{and} \quad
 \begin{array}{c|cccc}
 \text{R/P} & \theta_i & d_i & a_i & \alpha_i \\
 \hline
 \text{P} & 0 & q_1 & a_1 & 0 \\
 \text{R} & q_2 & 0 & 0 & \pi/2
 \end{array}
 \quad (3.48)$$

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	0	q_1	a_1	0	P	0	q_1	a_1	$\pi/2$	P	$\pi/2$	q_1	a_1	0	P	$\pi/2$	q_1	a_1	$\pi/2$
(a)					(b)					(c)					(d)				

Figure 3.8: First links obtained from the structural synthesis ($k_{\text{dof}} = 1$)

The first two entries of the motion vector are zero as in ξ_{req} (see (3.45)):

$$\xi = J\dot{q} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dot{q}_1 \\ 0 \\ 0 \\ \dot{q}_2 \end{pmatrix}. \quad (3.49)$$

For both isomorphisms, the condition $\omega_2 \times d_2 e_z^{(1)} = \mathbf{0}$ is fulfilled because ω_2 is parallel to $e_z^{(1)}$. Therefore, the parameter d_2 can take any value including zero.

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	0	q_1	a_1	0	P	0	q_1	a_1	$\pi/2$	P	$\pi/2$	q_1	a_1	0	P	$\pi/2$	q_1	a_1	$\pi/2$
R	q_2	d_2	0	0	R	q_2	d_2	0	0	R	q_2	d_2	0	0	R	q_2	d_2	0	0
(a)					(b)					(c)					(d)				
R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	0	q_1	a_1	0	P	0	q_1	a_1	$\pi/2$	P	$\pi/2$	q_1	a_1	0	P	$\pi/2$	q_1	a_1	$\pi/2$
R	q_2	d_2	0	$\pi/2$	R	q_2	d_2	0	$\pi/2$	R	q_2	d_2	0	$\pi/2$	R	q_2	d_2	0	$\pi/2$
(e)					(f)					(g)					(h)				

Figure 3.9: Architectures with 2 DOF

Carrying out the same procedure, the structures shown in figure 3.11 are obtained in the third stage ($k_{\text{dof}} = 3$). For the architecture in figure 3.11(e), $\text{rank}(\mathbf{J}) = 3$ and the first two entries of the motion vector are again zero:

$$\xi = J\dot{q} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & \sin(q_2) \\ 0 & 0 & -\cos(q_2) \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dot{q}_1 \\ \dot{q}_3 \sin(q_2) \\ -\dot{q}_3 \cos(q_2) \\ \dot{q}_2 \end{pmatrix}. \quad (3.50)$$

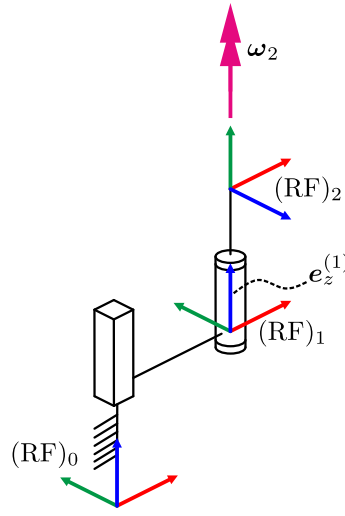


Figure 3.10: Sketch of the architecture in figure 3.9(e)

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	0	q_1	a_1	0	P	0	q_1	a_1	$\pi/2$	P	$\pi/2$	q_1	a_1	0	P	$\pi/2$	q_1	a_1	$\pi/2$
R	q_2	d_2	0	$\pi/2$	R	q_2	d_2	0	$\pi/2$	R	q_2	d_2	0	$\pi/2$	R	q_2	d_2	0	$\pi/2$
R	q_3	0	0	0	R	q_3	0	0	0	R	q_3	0	0	0	R	q_3	0	0	0
(a)					(b)					(c)					(d)				
R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	0	q_1	a_1	0	P	0	q_1	a_1	$\pi/2$	P	$\pi/2$	q_1	a_1	0	P	$\pi/2$	q_1	a_1	$\pi/2$
R	q_2	d_2	0	$\pi/2$	R	q_2	d_2	0	$\pi/2$	R	q_2	d_2	0	$\pi/2$	R	q_2	d_2	0	$\pi/2$
R	q_3	0	0	$\pi/2$	R	q_3	0	0	$\pi/2$	R	q_3	0	0	$\pi/2$	R	q_3	0	0	$\pi/2$
(e)					(f)					(g)					(h)				

Figure 3.11: Architectures with 3 DOF

It has to be noted, that the structures in figure 3.11 result from the architectures in figures 3.9(e) to 3.9(h). For the other structures (figures 3.9(a) to 3.9(d)), it was not possible to find any additional link fulfilling the conditions in (3.8) and (3.9).

Finally, for structures in figures 3.11(a) to 3.11(d), it was not possible to find additional links. However, the architecture in figure 3.12 is obtained by adding the links

$$\begin{array}{c|cccc} \text{R/P} & \theta_i & d_i & a_i & \alpha_i \\ \hline \text{R} & q_4 & 0 & 0 & \pi/2 \end{array} \quad \text{and} \quad \begin{array}{c|cccc} \text{R/P} & \theta_i & d_i & a_i & \alpha_i \\ \hline \text{R} & q_4 & 0 & 0 & 0 \end{array} \quad (3.51)$$

to the structures in figures 3.11(e) to figure 3.11(h) and then detecting the possible isomorphisms.

Figure 3.13 shows that θ_1 , a_1 , α_1 , d_2 , and α_4 have no influence on the manipulator's EE DOF.

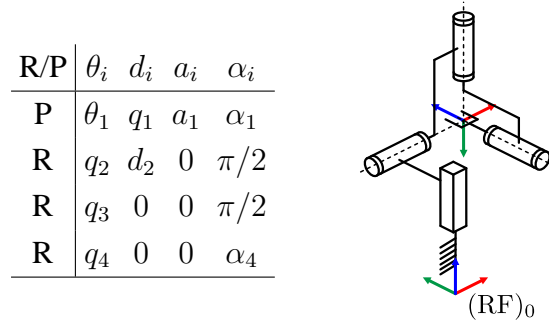


Figure 3.12: Example of an architecture with $\xi_{\text{req}} = (0, 0, \xi_{\text{req}_3}, \xi_{\text{req}_4}, \xi_{\text{req}_5}, \xi_{\text{req}_6})^T$, the optimization vector is $p_{\text{geom}} = (\theta_1, d_2, a_1, \alpha_1, \alpha_4)^T$

Therefore, these parameters can have any value and can be used as optimization parameters of the architecture. In the motion vector

$$\xi = \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & s(q_2) & -c(q_3) \\ 0 & 0 & -c(q_2) & c(q_2)s(q_3) \\ 0 & 1 & 0 & s(q_2)s(q_3) \end{pmatrix}}_{\mathbf{J}} \underbrace{\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix}}_{\dot{\mathbf{q}}} = \begin{pmatrix} 0 \\ 0 \\ \dot{q}_1 \\ \dot{q}_2 - \dot{q}_4 c(q_3) \\ \dot{q}_3 s(q_2) + \dot{q}_4 c(q_2) s(q_3) \\ \dot{q}_4 s(q_2) s(q_3) - \dot{q}_3 c(q_2) \end{pmatrix}, \quad (3.52)$$

it can be observed that $\text{rank}(\mathbf{J}) = 4$ and the first two entries of ξ are zero. This is one of the two architectures obtained for the required motion vector in (3.45) and was used as example.

In order to observe the importance of the isomorphisms detection, table 3.3 presents the number of architectures found for several motion required vectors. $N_{\text{ar}}^{(z)}$, $N_{\text{ar}}^{(y)}$, and $N_{\text{ar}}^{(x)}$ are the number of architectures whose first axis is parallel with the z , y , and x -axis, respectively. N_{ar} represents the total number of architectures. As it can be noted, the number of architectures is significantly reduced through the detection of isomorphisms. Therefore, the geometric optimization can now be performed for all suitable architectures.

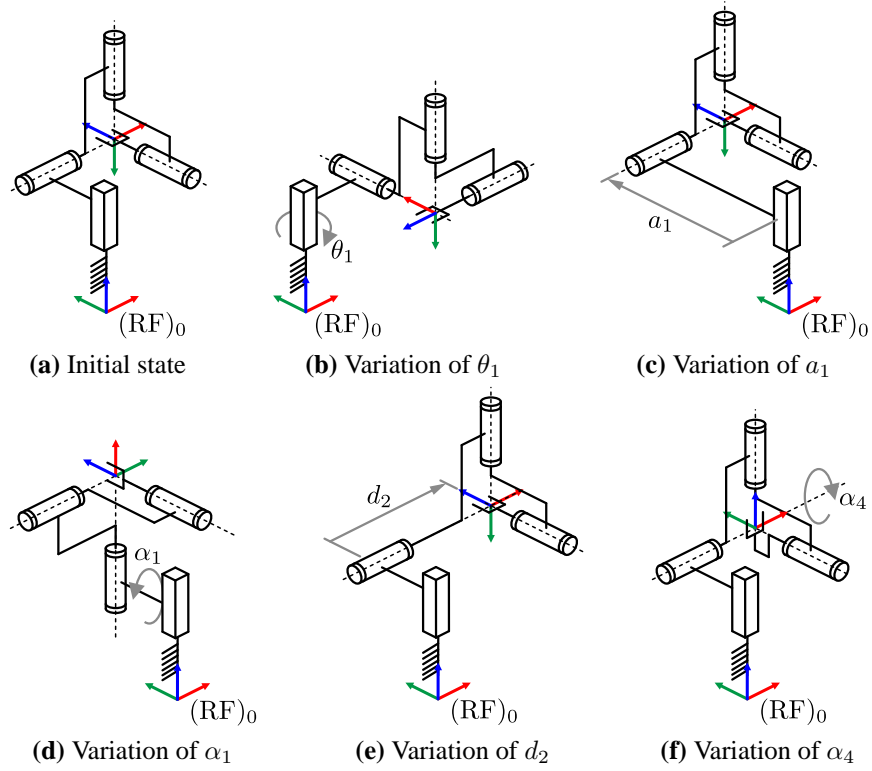


Figure 3.13: Variation of the modifiable geometric parameters for the exemplary architecture

Table 3.3: Number of architectures generated with and without isomorphisms detection

ξ_{req}	Without isomorphisms detection				With isomorphisms detection			
	$N_{\text{ar}}^{(z)}$	$N_{\text{ar}}^{(y)}$	$N_{\text{ar}}^{(x)}$	N_{ar}	$N_{\text{ar}}^{(z)}$	$N_{\text{ar}}^{(y)}$	$N_{\text{ar}}^{(x)}$	N_{ar}
$(0,0,\xi_{\text{req}_3},0,0,\xi_{\text{req}_6})^T$	12	0	0	12	2	0	0	2
$(\xi_{\text{req}_1},0,0,0,0,\xi_{\text{req}_6})^T$	0	0	4	4	0	0	1	1
$(\xi_{\text{req}_1},\xi_{\text{req}_2},0,0,0,0)^T$	0	8	8	16	0	1	1	2
$(0,\xi_{\text{req}_2},\xi_{\text{req}_3},0,0,\xi_{\text{req}_6})^T$	8	24	0	32	1	2	0	3
$(\xi_{\text{req}_1},\xi_{\text{req}_2},0,0,0,\xi_{\text{req}_6})^T$	112	64	64	240	4	3	3	10
$(0,0,0,\xi_{\text{req}_4},\xi_{\text{req}_5},\xi_{\text{req}_6})^T$	2	2	2	6	1	1	1	3
$(\xi_{\text{req}_1},\xi_{\text{req}_2},\xi_{\text{req}_3},0,0,0)^T$	32	32	32	96	1	1	1	3
$(0,0,\xi_{\text{req}_3},\xi_{\text{req}_4},\xi_{\text{req}_5},\xi_{\text{req}_6})^T$	20	0	0	20	2	0	0	2
$(\xi_{\text{req}_1},\xi_{\text{req}_2},\xi_{\text{req}_3},0,0,\xi_{\text{req}_6})^T$	1984	704	704	3392	17	9	9	35
$(0,\xi_{\text{req}_2},\xi_{\text{req}_3},\xi_{\text{req}_4},\xi_{\text{req}_5},\xi_{\text{req}_6})^T$	220	220	388	828	7	5	11	23
$(\xi_{\text{req}_1},\xi_{\text{req}_2},\xi_{\text{req}_3},\xi_{\text{req}_4},\xi_{\text{req}_5},\xi_{\text{req}_6})^T$	11,272,192		33,816,576	326	326	326		978

4 Proposed combined robot synthesis

After the generation of the task suitable structures explained in chapter 3, the next step in the CRS is their geometric optimization as introduced in section 4.1. The goal of this optimization is to determine the geometry of the optimal manipulator for a desired task, which is defined besides the formal optimization problem in section 4.2. The optimization parameters correspond to the geometric parameters of the suitable structures as well as the position of the robot base (section 4.3). A short summary of the particle swarm optimization algorithm, which is employed to solve the optimization problem, is presented in section 4.4.

In a usual dimensional synthesis, only one architecture is optimized and, therefore, the modeling process is performed for this single architecture. However, this is not the case of the CRS. Here, a general modeling is mandatory, which involves the forward, inverse, and differential kinematics as well as the robot dynamics. The forward and differential kinematics were previously addressed in chapter 2. The inverse kinematics, which represents a challenge for serial kinematic chains, is solved by means of a numeric method. This approach as well as additional elements to be considered in the general dynamic modeling are introduced in section 4.5

An important feature of the CRS is the comparison of the performance for several structures and the postprocessing of the results as shown in section 4.6, where the sensitivity of the manipulators with the best performance is analyzed.

4.1 Optimization strategy

After the structural synthesis, every obtained architecture is optimized as depicted in figure 4.1. Prior to the optimization, each architecture is modeled considering the optimization parameters \mathbf{p} . For this purpose, a MATLAB[®] class called *SerialChain* was programmed. An object of this class represents an architecture and contains the information about the type of joints, the geometry of the robot (both fixed and variable parameters) as well as the necessary functions to calculate the forward, inverse, and differential kinematics. These functions are formulated for general serial manipulators with up to 6 DOF. The forward and differential kinematics are computed as introduced in section 2.2 and section 2.4, respectively. The inverse kinematics is solved using a numeric approach which is explained in section 4.5.1. Furthermore, the *SerialChain* class includes functions to evaluate the kinematic and dynamic performance of a manipulator.

Due to the capabilities of the *SerialChain* class, several performance indices can be evaluated for any serial manipulator up to 6 DOF. They are one way to quantify kinematic and dynamic

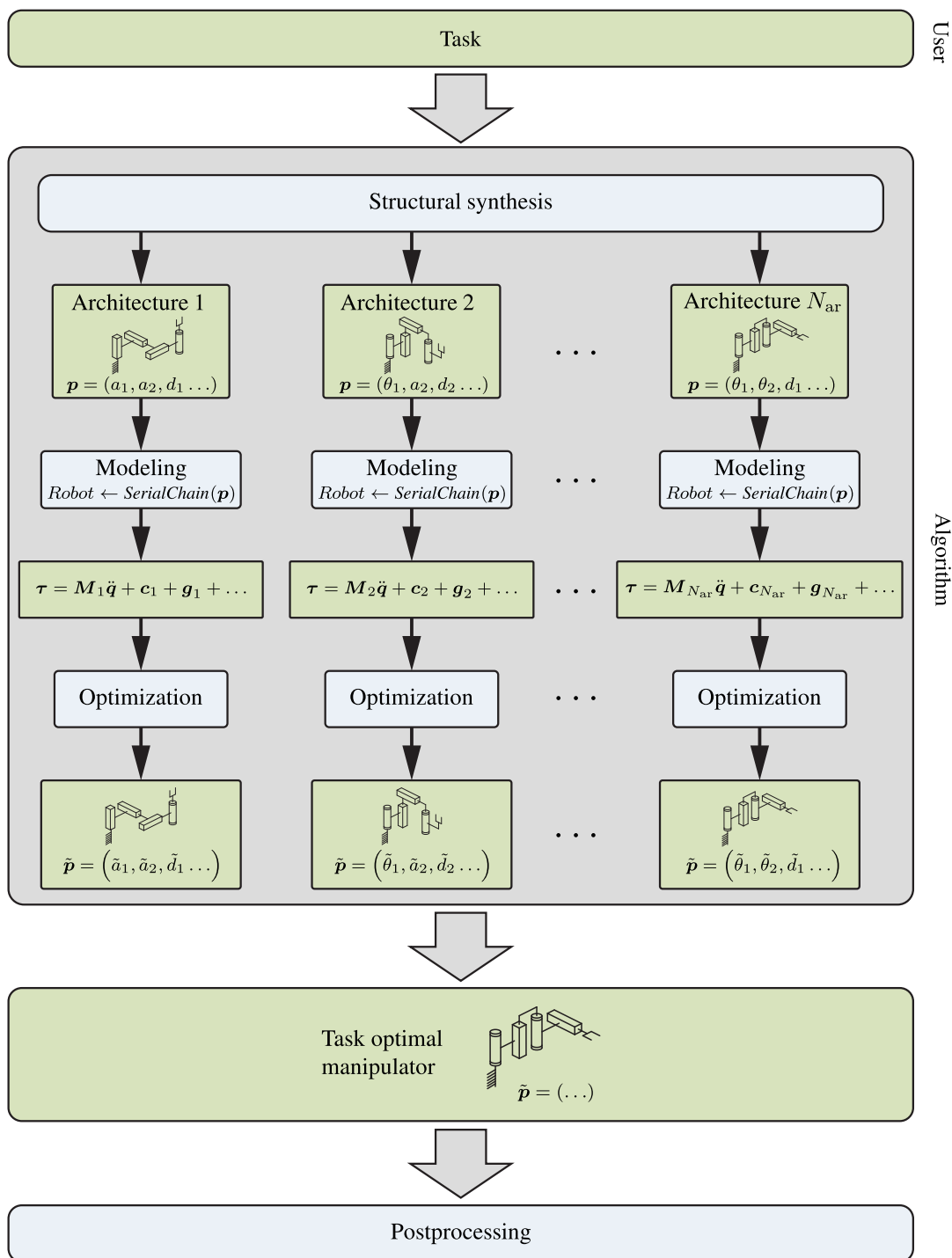


Figure 4.1: Proposed strategy for the optimization of the suitable structures

characteristics of a given robot. Robot manipulators are usually optimized regarding several performance indices [SK08, COC08]. This multiobjective optimization problem can be addressed by determining a Pareto front or by calculating the objective function as the weighted sum of the desired performance indices [Wei09]. Pareto fronts demand the analysis of each structure and are

impracticable in the combined robot synthesis (CRS) due to the high number of architectures to be considered. In the weighted sum of indices, for its part, the indices included in the sum usually have different physical units, e. g. workspace (m^3) and manipulability (no physical unit). This produces values of the objective functions that are difficult to interpret or without physical meaning. Additionally, the weight factors are often subjectively given by the designer, which leads to a high user-dependency of the results. Further information about performance indices that have been applied in synthesis of manipulators can be found in [PS15a, SK08, WGLZ12, MSC⁺12a, COC07]. In the practical design of robot manipulators, the user requirements are usually established as a principal attribute to be optimized and a set of boundary specifications. These are given as threshold values that have to be achieved by the robot and are considered as constraints. Hence, the optimization in the CRS is carried out regarding one main criterion, i. e. the objective function is calculated with one single performance index.

Due to the characteristics of the geometric optimization problem, the minimax method [COC07], and global optimization techniques as simulated annealing [PS14] and genetic algorithms (GA) [LMRD10] have been employed to solve it. When the robot architecture is considered as part of the optimization problem, the joints type selection represents a discrete problem in which it is not possible to establish a gradient between several solutions. Therefore, the characteristics of GA have been exploited in previous works [CR96, RSP⁺11]. However, since GA is a heuristic method, it does not ensure that all possible architectures are considered [Plu16]. The structural synthesis introduced in chapter 3 allows for avoid this problem and converts the whole optimization problem (including the choice of architecture) into a continuous problem. Here, the adjustable geometric parameters are the minimum parameters that ensure the achievement of the given task.

The PSO algorithm has shown good performance when the objective function presents local minimums [Kot13, EKG12]. Furthermore, the method can exploit the advantages of parallel computing. In this case, each particle can run on a different thread reducing the computational time. PSO has been previously applied in robot optimization problems, e. g. in the dimensioning of robots with variable geometry [Kot13, KAOH09] or in the path planning of industrial robots [HIMO12, HEKO14, HKO14]. This algorithm is used in this work to solve the geometric optimization of each architecture (see figure 4.1). Since the equivalence between PSO and other global optimization methods for solving problems with continuous search space has been discussed before [SL13, ES98, Plu15], a comparison between several optimization techniques is not part of this work and will not be further discussed.

During the optimization, a swarm is generated for each architecture. Each particle of the swarm represents the dimensions of the architecture. Due to the complexity of the constraints, penalty values are applied to deal with the constraints violation. In order to facilitate the analysis of the results and improve the performance of the optimization, different penalty values are assigned to each constraint.

With the purpose of avoiding stagnation in a local minimum, three criteria have been proposed in [Ber02]: maximum swarm radius, cluster analysis, and objective function slope. Even though the first two methods outperform the function slope [Ber02, Eve09, EB09], the latter is used in this work due to the inhomogeneities and diversity of the optimization parameters. The objective function slope restarts the algorithm after a predefined number of iterations without improvement. The PSO stops when the change in the objective function of the global best particle after restarting is less than a given threshold.

As result of the optimization, the vector $\tilde{\mathbf{p}}$, containing the optimal geometric parameters and the optimal position of the robot's base, is obtained for each architecture. The architecture with the best performance accompanied by its optimal geometric parameters corresponds to the task optimal manipulator. Afterwards, the sensitivity of the best architectures is analyzed (postprocessing) through a Monte Carlo simulation. The sensitivity of the architectures with the best performance is a useful tool to be taken into account in the robot design process.

4.2 Definition of the optimization problem

The required task is defined by a set of poses ${}^{(W)}\mathbf{x}_{\text{pose}_1}, {}^{(W)}\mathbf{x}_{\text{pose}_2}, \dots, {}^{(W)}\mathbf{x}_{\text{pose}_N}$ that have to be passed by the EE at the time t_s . Beginning from $\mathbf{x}_{\text{pose}_1}$, the manipulator should move to $\mathbf{x}_{\text{pose}_2}$ within the time Δt_1 , then $\mathbf{x}_{\text{pose}_3}$ in Δt_2 , etc. Each pose ${}^{(W)}\mathbf{x}_{\text{pose}_s}$ is defined with respect to the global reference frame (RF)_W (see figure 4.2). It is composed by the position vector ${}^{(W)}\mathbf{r}_{\text{pose}_s}$ and an orientation description ${}^{(W)}\Phi_{\text{pose}_s}$:

$${}^{(W)}\mathbf{x}_{\text{pose}_s} = \begin{pmatrix} {}^{(W)}\mathbf{r}_{\text{pose}_s} \\ {}^{(W)}\Phi_{\text{pose}_s} \end{pmatrix}, \quad (4.1)$$

with

$${}^{(W)}\mathbf{r}_{\text{pose}_s} = ({}^{(W)}r_x, {}^{(W)}r_y, {}^{(W)}r_z)^T, \quad (4.2)$$

$${}^{(W)}\Phi_{\text{pose}_s} = ({}^{(W)}\Phi_z, {}^{(W')}\Phi_y, {}^{(W'')}\Phi_x)^T. \quad (4.3)$$

The orientation ${}^{(W)}\Phi_{\text{pose}_s}$ of each pose is defined in this work using the $z'y'z''$ EULER angles representation [Cra05]. Other EULER angles representation as well as fixed angles (e. g. roll-pitch-yaw) can be also used. At the end of the task, the manipulator returns to the start pose $\mathbf{x}_{\text{pose}_1}$ (see figure 4.2). Between the poses, point-to-point (PTP) motions are generated using fifth order polynomials in the joint space [BM08].

The geometric optimization problem in the CRS is formulated for each architecture as

$$\tilde{\mathbf{p}} = \arg \min_{\mathbf{p}} h(\mathbf{p}), \quad (4.4)$$

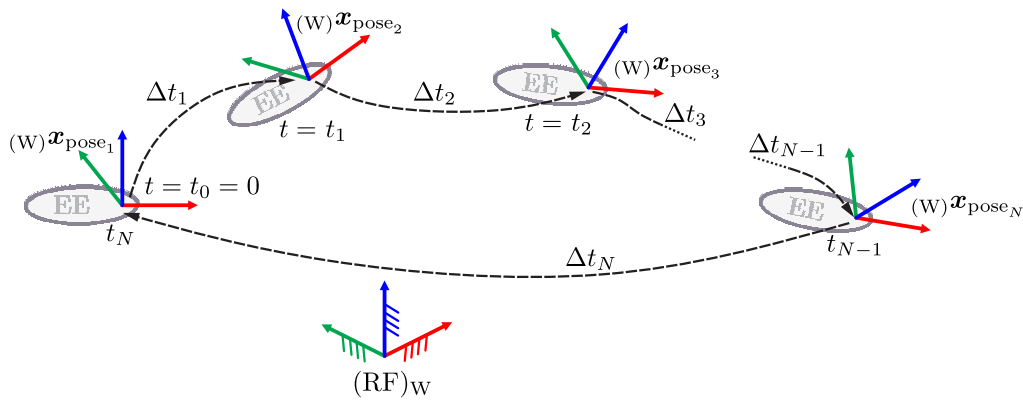


Figure 4.2: Exemplary task specification

subjected to

$$\begin{aligned}
 c_1(\mathbf{p}) &\leq 0, \\
 c_2(\mathbf{p}) &\leq 0, \\
 &\vdots \\
 c_{N_c}(\mathbf{p}) &\leq 0, \\
 \mathbf{p}_{\min} &\leq \mathbf{p} \leq \mathbf{p}_{\max},
 \end{aligned} \tag{4.5}$$

with \mathbf{p}_{\min} and \mathbf{p}_{\max} being the lower and upper bounds of \mathbf{p} respectively. The objective function $h(\mathbf{p})$ and the constraints $c_1(\mathbf{p}) \dots c_{N_c}(\mathbf{p})$ correspond to performance indices that quantify the performance of a robot with respect to the task requirements. Usual criteria to evaluate this performance are, for instance, workspace size, robot size, dexterity, manipulability, obstacles avoidance, required torque at the actuators, or energy consumption.

4.3 Optimization parameters

In the usual optimization of manipulators, only the geometry of a given robot is optimized. In the CRS in contrast, the position of the robot basis is also considered allowing to determine its optimal location in a cell or with respect to the desired task. Thanks this feature, the CRS can be also used to position and compare a set of predefined manipulators. The reference frame of a robot $(RF)_0$ is, therefore, used to describe the position and orientation of the robot base with respect to the global reference frame $(RF)_W$. The transformation ${}^W T_0$ between both frames is given by (see figure 4.3)

$${}^W T_0 = \left(\begin{array}{ccc|c}
 {}^W R_0 & & & ({}^W r_0) \\
 0 & 0 & 0 & 1
 \end{array} \right). \tag{4.6}$$

The rotation matrix ${}^W\mathbf{R}_0$ represents the orientation of $(\text{RF})_0$ with respect to $(\text{RF})_W$. The vector ${}^{(W)}\mathbf{r}_0 \in \mathbb{R}^3$ is the position of the robot base.

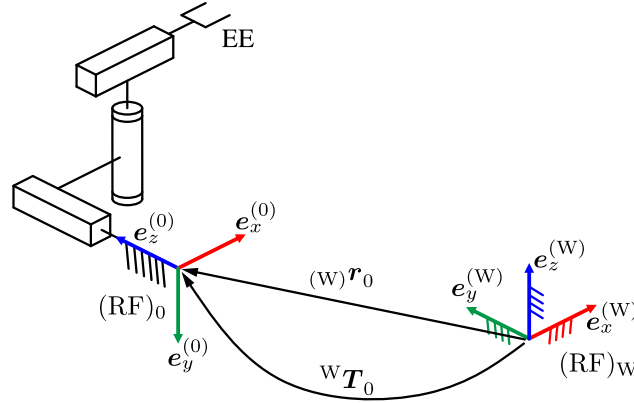


Figure 4.3: Location of the robot base with respect to the global reference frame $(\text{RF})_W$

Since the position of the EE has to be described with respect to $(\text{RF})_W$, equation (2.13) for the forward kinematics is modified as

$${}^W\mathbf{T}_{\text{EE}} = {}^W\mathbf{T}_0 {}^0\mathbf{T}_{\text{EE}}(\mathbf{q}) = {}^W\mathbf{T}_0 {}^0\mathbf{T}_1(q_1) {}^1\mathbf{T}_2(q_2) \dots {}^{n-1}\mathbf{T}_n(q_n). \quad (4.7)$$

The structural synthesis generates architectures with different orientations (see section 3.1). Hence, the orientation of the robot base is evaluated through the consideration of these architectures. The rotation matrix ${}^W\mathbf{R}_0$ in (4.6) corresponds to the identity matrix in architectures whose first joint axis is parallel to the z -axis of $(\text{RF})_W$, hereafter referred to as z -oriented architectures. In the case of architectures whose first joint axis is parallel to the y -axis of $(\text{RF})_W$ (y -oriented architectures), ${}^W\mathbf{R}_0$ results from rotating $-\pi/2$ around the x -axis. Similarly, ${}^W\mathbf{R}_0$ is calculated rotating $\pi/2$ around the y -axis if the first joint axis is parallel to the x -axis of $(\text{RF})_W$ (x -oriented architectures), i. e.:

$${}^W\mathbf{R}_0 = \mathbf{I}_{[3 \times 3]} \quad \text{if } \mathbf{e}_z^{(0)} \text{ is parallel to } \mathbf{e}_z^{(W)}, \quad (4.8)$$

$${}^W\mathbf{R}_0 = \mathbf{R}_x(-\pi/2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \quad \text{if } \mathbf{e}_z^{(0)} \text{ is parallel to } \mathbf{e}_y^{(W)}, \quad (4.9)$$

$${}^W\mathbf{R}_0 = \mathbf{R}_y(\pi/2) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \quad \text{if } \mathbf{e}_z^{(0)} \text{ is parallel to } \mathbf{e}_x^{(W)}. \quad (4.10)$$

In the optimization, the position coordinates of the robot base are considered as additional optimization parameters. Therefore, they are included in the optimization vector \mathbf{p} . For instance, the optimization vector for the 4 DOF architecture presented in figure 3.12 is given by

$$\mathbf{p} = \left(\underbrace{\theta_1, d_2, a_1, \alpha_1, \alpha_4}_{\mathbf{p}_{\text{geom}}}, {}^{(W)}\mathbf{r}_0^T \right)^T. \quad (4.11)$$

4.4 Optimization algorithm

From manipulators geometry to path planning, the optimization of robotic systems has been addressed in several works [Mer05b, COC07, PS15a, Kir00]. In the synthesis of robot manipulators, a manipulator is usually evaluated regarding one or more performance indices. These performance indices are employed to form an objective function (also called cost function or fitness function), which is minimized by means of an optimization method. The objective function is often too complex to be expressed in an algebraic equation. Additionally, it can exhibit several local minima and the search space is not easily defined due to the constraints imposed by the requirements. Since global optimization methods do not depend on the initial conditions and have shown adequate results solving this kind of problems, they are preferred for robot optimization problems. Some examples of the application of these algorithms are the genetic algorithms [LMRD10, Kir00] and the particle swarm optimization (PSO) algorithm [Kot13, KAOH09].

The PSO algorithm is based on the behavior of animals living in a group (swarm). In this way, they can enhance the ability of seeking food or defending against predators. In a swarm, the intelligence is spread over the whole group instead of being concentrated on each member. In PSO, the swarm is represented by N_{PSO} particles, each one having a position $\mathbf{p}_{k_{\text{PSO}}}^{(i)}$ and a velocity $\mathbf{v}_{k_{\text{PSO}}}^{(i)}$. The position of a particle corresponds to the optimization vector \mathbf{p} containing the optimization parameters. Each particle represents thus a possible solution of the optimization. Given an objective function $h(\mathbf{p})$, the functions $c(\mathbf{p})$, $c_{\text{eq}}(\mathbf{p})$, the matrices \mathbf{A} , \mathbf{A}_{eq} , and the vectors \mathbf{b} and \mathbf{b}_{eq} , the optimization problem [EKG12] is defined as:

$$\min_{\mathbf{p}} h(\mathbf{p}), \quad (4.12)$$

subject to:

$$\begin{aligned} \mathbf{A}\mathbf{p} &\leq \mathbf{b} \\ \mathbf{A}_{\text{eq}}\mathbf{p} &= \mathbf{b}_{\text{eq}} \\ c(\mathbf{p}) &\leq 0 \\ c_{\text{eq}}(\mathbf{p}) &= 0 \\ \mathbf{p}_{\min} &\leq \mathbf{p} \leq \mathbf{p}_{\max}. \end{aligned} \quad (4.13)$$

The lower and upper boundaries of the optimization parameters are contained in the vectors \mathbf{p}_{\min} and \mathbf{p}_{\max} . Constraints $c(\mathbf{p})$ and $c_{\text{eq}}(\mathbf{p})$ are nonlinear functions describing implicit equality and

inequality constraints, respectively [CTS04]. They are usually distinguished from the linear equality and inequality constraints \mathbf{A} and \mathbf{A}_{eq} . In the case of the CRS, several inequality constraints have to be considered as introduced in (4.5).

The objective function is calculated for the position of each particle. The goal of the algorithm is to find the position that results in the minimum value of $h(\mathbf{p})$. In order to seek for lower values of the objective function, the position of a particle i in the iteration step k_{PSO} is updated in the step $k_{\text{PSO}} + 1$ using its velocity $\mathbf{v}_{k_{\text{PSO}}+1}^{(i)}$ (see figure 4.4):

$$\mathbf{p}_{k_{\text{PSO}}+1}^{(i)} = \mathbf{p}_{k_{\text{PSO}}}^{(i)} + \mathbf{v}_{k_{\text{PSO}}+1}^{(i)}, \quad (4.14)$$

where the new particle velocity $\mathbf{v}_{k_{\text{PSO}}+1}^{(i)}$ is calculated using:

$$\begin{aligned} \mathbf{v}_{k_{\text{PSO}}+1}^{(i)} = & w_{k_{\text{PSO}}} \mathbf{v}_{k_{\text{PSO}}}^{(i)} + c_{1\text{PSO}} \text{diag} \left(\boldsymbol{\gamma}_1^{(i)} \right) \left(\mathbf{p}_b^{(i)} - \mathbf{p}_{k_{\text{PSO}}}^{(i)} \right) \\ & + c_{2\text{PSO}} \text{diag} \left(\boldsymbol{\gamma}_2^{(i)} \right) \left(\mathbf{p}_g - \mathbf{p}_{k_{\text{PSO}}}^{(i)} \right). \end{aligned} \quad (4.15)$$

Variables $\boldsymbol{\gamma}_1^{(i)}$ and $\boldsymbol{\gamma}_2^{(i)}$ are vectors of random values uniformly distributed between 0 and 1 [EB09]. Vector $\mathbf{p}_b^{(i)}$ is the personal best position of the i -th particle and \mathbf{p}_g is the best position of its neighborhood up to the current step. The particle's neighborhood is a subset \mathcal{S} of the swarm whose size is controlled by the neighborhood size $0 < \sigma \leq 1$. When $\sigma = 1$, the neighborhood corresponds to the whole swarm and \mathbf{p}_g is the global best position. Depending on the neighborhood topology [Sim13, JJ13], there are several ways to define which particles are included.

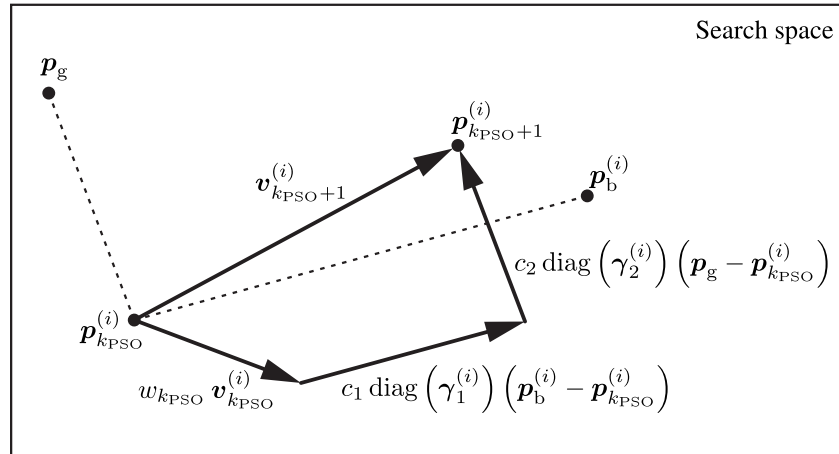


Figure 4.4: Representation of a particle's position change in the PSO

Furthermore, the parameters $w_{k_{\text{PSO}}}$, $c_{1\text{PSO}}$, and $c_{2\text{PSO}}$ are called the inertia weight [JJ13], the cognitive learning rate and the social learning rate, respectively [Rao09, Sim13]. The inertia weight helps to dampen the velocity preventing the algorithm from exploding. High values of $w_{k_{\text{PSO}}}$ improve the exploration of the algorithm but reduce its convergence rate. On the other side, the

algorithm can stagnate in a local minimum if the value of $w_{k_{\text{PSO}}}$ is too low. A common practice is to decrease the value of $w_{k_{\text{PSO}}}$ over the algorithm iterations [Sim13, ZZLC09]. The exploration of the algorithm is also enhanced when $c_{1_{\text{PSO}}} > c_{2_{\text{PSO}}}$ due to higher influence of the personal best position. Otherwise, when $c_{1_{\text{PSO}}} < c_{2_{\text{PSO}}}$, there is more influence of the neighborhood best position producing faster convergence and lower exploration. In order to ensure stability of the swarm, two conditions are suggested in [Sim13, PB07]:

$$0 < c_{1_{\text{PSO}}} + c_{2_{\text{PSO}}} < 4 \quad (4.16)$$

and

$$\frac{c_{1_{\text{PSO}}} + c_{2_{\text{PSO}}}}{2} - 1 < w_{k_{\text{PSO}}} < 1. \quad (4.17)$$

Some methods for dealing with the constraints (4.13) in PSO are penalty functions, absorption, and nearest position [Rao09, EKG12]. Details about these approaches as well as variations of the PSO can be found in [EKG12, EB09, Eve09, KES01, ES98, MHHBJ13]. In this work, constraint violations are managed by assigning penalty values to the objective function h . Different penalty values are used for each constraint $c_1(\mathbf{p}) \dots c_{N_c}(\mathbf{p})$. This facilitates the analysis of the optimization results. Thanks to the different values, it is possible to recognize which constraint is more difficult to overcome for each architecture. Additionally, different values for constraints violation are useful in the postprocessing to identify the constraints that are more influenced by the changes in the geometric parameters. Furthermore, several experiments showed that the use of different penalty values provides a faster convergence of the PSO. Assigning the highest value to the penalty value h_{p0} in figure 4.5 helps the particles to restrict quickly the exploration to the valid search space (solutions that are able to reach all task's poses).

Figure 4.5 shows the procedure to calculate the objective function h . For each architecture, a set of vectors \mathbf{p} is generated in the PSO. Each vector defines the geometry and the base position of a manipulator under evaluation. Firstly, the inverse kinematics is calculated for the poses defined in the task. If any pose can not be reached by the manipulator, the penalty value h_{p0} is assigned to h and no further evaluations are performed. Secondly, the path is planned in the joint space for the given poses. In this work PTP motions are generated using a 5-th order polynomial interpolation but other type of interpolations can be used as well. In order to improve the computational efficiency and obtain results with adequate accuracy, the interpolation step width is varied during the optimization. In the beginning, a large step width is used, allowing for finding geometries that fulfill the optimization constraints in short time. In the last optimization iterations, the step width is reduced to promote a more accurate comparison between the manipulators. After path planning, the first constraint is evaluated. If it is not fulfilled, the penalty value h_{p1} is given to h . If $c_1(\mathbf{p})$ is satisfied, the next constraint is evaluated. If all constraints are fulfilled, h is calculated to evaluate the robot with respect to the main optimization criteria. The methodology can be applied to synthesize manipulators regarding any performance index e.g. manipulability, kinematic accuracy, etc.

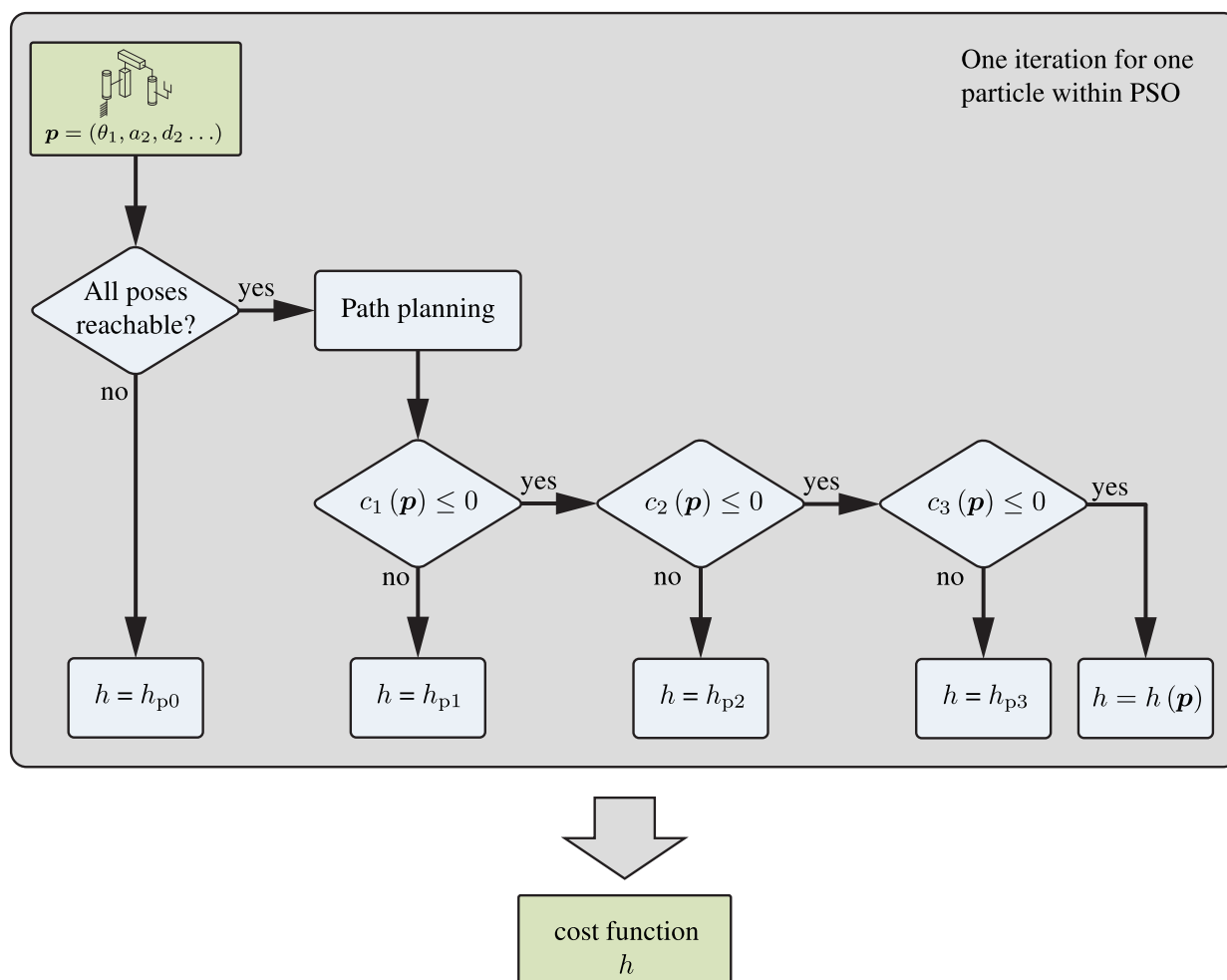


Figure 4.5: Evaluation of the objective function within the optimization

4.5 Modeling of general serial kinematics

As several architectures are considered in the optimization, each one with different optimization parameters, it is necessary to employ a general method for the kinematic and dynamic modeling of the manipulators. Kinematics modeling includes the forward kinematics, the inverse kinematics, and the differential kinematics. The first was explained in sections 2.2 and 4.3. The inverse kinematics, addressed in section 4.5.1, is solved through a new numeric approach. As the method uses the Rodrigues parameters to represent the orientation error of the EE, this requires the JACOBIAN matrix derived in section 4.5.2. The differential kinematics is calculated as introduced in section 2.4. An important aspect to be considered in the robot synthesis is the absence of collisions during the task execution. This is evaluated through the approach introduced in section 4.5.3. Furthermore, the calculation of dynamic performance indices demands the automatic dynamic modeling of the architectures as shown in section 4.5.4.

4.5.1 Inverse kinematics

The inverse kinematics calculates the joint coordinates \mathbf{q} from the EE pose \mathbf{x}_{EE} . Considering (2.11), it can be written as

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x}_{EE}). \quad (4.18)$$

Closed-form solutions of (4.18) have been proposed for some specific manipulators like the SCARA robot [Cra05], the Puma robot [WL89], or robots with specific structural characteristics [AK06, Pie68]. Algebraic manipulation of the forward kinematics equation in terms of the homogeneous transformation matrices (2.13) has been used in order to reduce the order of the resulting nonlinear equations [Cra05, RR93]. In [Gro15], the nonlinear equations of the inverse kinematics are represented as multivariate polynomials and solved using the equivalent eigenvalue problem [RR93, KO93]. Furthermore, the application of the Study parameters [Gfr00] has allowed the development of the algebraic calculation of the inverse kinematics of 6R manipulators without structural simplifications [HPS07, BAH15, APBH13]. However, as the geometric parameters of the manipulators are not constant during the CRS (link lengths and joint orientations), they should be considered as variables and the algebraic manipulation of the kinematic equations would be unfeasible. Hence, numerical methods are preferred in this work for the solution of the inverse kinematics.

Some numeric approaches are based on learning algorithms [HHIAA06], artificial neural networks [KA00], as well as on usual optimization methods [WC91, Bor16, PK93]. However, the most extended method is the Newton-Raphson method [GBF85, Ang85, SS87, BK05, Cor11, SSVO09], in which the solution of (4.18) is approximated stepwise as

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta\mathbf{q}_k \quad (4.19)$$

until the error vector $\Delta\mathbf{x}_k$ between the desired EE pose \mathbf{x}_{EE^*} and the current EE pose

$$\mathbf{x}_{EE,k} = \mathbf{f}(\mathbf{q}_k) \quad (4.20)$$

in the iteration step k is smaller than a threshold ϵ .

The error vector $\Delta\mathbf{x}_k$ is composed by two parts, one corresponds to the position error vector ${}^{(W)}\Delta\mathbf{r}_k$, the other to the orientation error vector ${}^{(W)}\Delta\boldsymbol{\psi}_k$, i. e.:

$$\Delta\mathbf{x}_k = \begin{pmatrix} {}^{(W)}\Delta\mathbf{r}_k \\ {}^{(W)}\Delta\boldsymbol{\psi}_k \end{pmatrix}, \quad (4.21)$$

with the position error

$${}^{(W)}\Delta\mathbf{r}_k = {}^{(W)}\mathbf{r}_{EE^*} - {}^{(W)}\mathbf{r}_{EE,k}(\mathbf{q}_k). \quad (4.22)$$

Vector ${}^{(W)}\mathbf{r}_{EE^*}$ represents the desired EE position and ${}^{(W)}\mathbf{r}_{EE,k}(\mathbf{q}_k)$ the current EE position vector calculated using (4.7).

To express the orientation error of the EE, let $(RF)_{EE^*}$ be the desired reference frame representing the desired EE pose (see figure 4.6) and $(RF)_{EE,k}$ the EE reference frame in the step k (current pose). The rotation matrix

$${}^c\mathbf{R}_d = {}^{EE,k}\mathbf{R}_{EE^*} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = {}^W\mathbf{R}_{EE,k}^T {}^W\mathbf{R}_{EE^*} \quad (4.23)$$

represents the orientation of $(RF)_{EE^*}$ with respect to $(RF)_{EE,k}$. The notation ${}^c\mathbf{R}_d$ will be used below only to improve the readability.

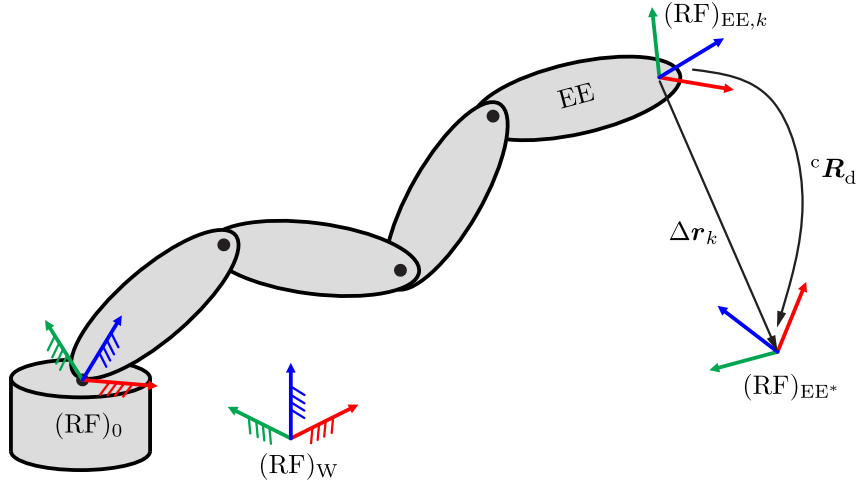


Figure 4.6: Representation of the desired and the current EE pose for the inverse kinematics

Several possibilities to express the orientation error of the EE have been proposed, e. g. through the EULER angles [GBF85] or through invariants in the rotational part of the closure equations [Ang85]. Using the latter approach, equation (4.18) produces an overdetermined nonlinear algebraic system with seven equations. In order to avoid the overdetermination and employ a more meaningful expression of the orientation error, $\Delta\psi_k$ is defined in this work by means of the Rodrigues vector \mathbf{b} [Dai15, Bon15]:

$${}^{(EE,k)}\Delta\psi_k = \mathbf{b} = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \tan\left(\frac{\phi}{2}\right) \mathbf{u}. \quad (4.24)$$

The quantities b_x , b_y , and b_z correspond to the Rodrigues parameters associated with ${}^c\mathbf{R}_d$. The vector \mathbf{u} is the unit vector parallel to the rotation axis of ${}^c\mathbf{R}_d$, the angle ϕ is the associated rotation angle (axis-angle representation).

The invariant vector of ${}^c\mathbf{R}_d$ [Ang85, Ang04], defined as

$$\text{vect}({}^c\mathbf{R}_d) = \frac{1}{2} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix} \quad (4.25)$$

is related to \mathbf{u} using

$$\text{vect}({}^c\mathbf{R}_d) = \mathbf{u} \sin(\phi) . \quad (4.26)$$

Additionally, the trace of ${}^c\mathbf{R}_d$ exhibits the property

$$\text{tr}({}^c\mathbf{R}_d) = 1 + 2 \cos(\phi) . \quad (4.27)$$

Using the half angle identity [Ste12]

$$\tan\left(\frac{\phi}{2}\right) = \frac{\sin(\phi)}{1 + \cos(\phi)} , \quad (4.28)$$

equation (4.26) and (4.27) can be combined with (4.24) to express ${}_{(EE,k)}\Delta\psi_k$ as

$$\begin{aligned} {}_{(EE,k)}\Delta\psi_k &= \tan\left(\frac{\phi}{2}\right) \frac{1}{\sin(\phi)} \text{vect}({}^c\mathbf{R}_d) \\ &= \frac{2}{1 + \text{tr}({}^c\mathbf{R}_d)} \text{vect}({}^c\mathbf{R}_d) , \end{aligned} \quad (4.29)$$

which is the orientation error vector expressed in $(\text{RF})_{EE,k}$. To write this vector with respect to $(\text{RF})_W$, the following equation applies:

$${}_{(W)}\Delta\psi_k = \frac{2}{1 + \text{tr}({}^c\mathbf{R}_d)} {}^W\mathbf{R}_{EE,k} \text{vect}({}^c\mathbf{R}_d) . \quad (4.30)$$

In order to solve (4.18) by means of (4.19), it is necessary to define the JACOBIAN $\mathbf{J}_{IK,k}$ deriving (4.21)

$$\mathbf{J}_{IK,k} = \frac{\partial \Delta \mathbf{x}_k}{\partial \mathbf{q}} = \begin{pmatrix} \frac{\partial {}_{(W)}\Delta \mathbf{r}_k}{\partial \mathbf{q}} \\ \frac{\partial {}_{(W)}\Delta \psi_k}{\partial \mathbf{q}} \end{pmatrix} . \quad (4.31)$$

This JACOBIAN $\mathbf{J}_{IK,k}$ is related to the JACOBIAN of the manipulator as defined in (2.19) by means of

$$\mathbf{J}_{IK,k} = \begin{pmatrix} \mathbf{J}_t \\ \mathbf{QJ}_r \end{pmatrix} , \quad (4.32)$$

where $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$ is determined using

$$\begin{aligned} \mathbf{Q} = & \frac{4}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} {}^W\mathbf{R}_{EE,k} \left(\text{vect}({}^c\mathbf{R}_d) \text{vect}({}^W\mathbf{R}_{EE^*} {}^c\mathbf{R}_d^T {}^W\mathbf{R}_{EE^*}^T)^T \right) \\ & + \frac{-1}{1 + \text{tr}({}^c\mathbf{R}_d)} \left({}^W\mathbf{R}_{EE^*} {}^c\mathbf{R}_d {}^W\mathbf{R}_{EE^*}^T - {}^W\mathbf{R}_{EE^*} {}^c\mathbf{R}_d^T {}^W\mathbf{R}_{EE^*}^T \right) \\ & + \frac{-1}{1 + \text{tr}({}^c\mathbf{R}_d)} {}^W\mathbf{R}_{EE,k} \left(\mathbf{I} \text{tr}({}^c\mathbf{R}_d^T) - {}^c\mathbf{R}_d^T \right) {}^W\mathbf{R}_{EE^*}^T. \end{aligned} \quad (4.33)$$

The detailed deduction of (4.33) is presented in section 4.5.2.

The difference $\Delta \mathbf{q}_k$ in (4.19) is calculated using

$$\Delta \mathbf{q}_k = \mathbf{K}_{\text{ps}} \mathbf{J}_{\text{IK},k}^\dagger(\mathbf{q}_k) \Delta \mathbf{x}_k, \quad (4.34)$$

being $\mathbf{J}_{\text{IK},k}^\dagger$ the pseudoinverse of $\mathbf{J}_{\text{IK},k}$ [SK08]

$$\mathbf{J}_{\text{IK},k}^\dagger = \mathbf{J}_{\text{IK},k}^T \left(\mathbf{J}_{\text{IK},k} \mathbf{J}_{\text{IK},k}^T \right)^{-1}, \quad (4.35)$$

and \mathbf{K}_{ps} a diagonal matrix that can be used to modify the step size during the algorithm or to adjust the influence of the translational and the rotational part of $\Delta \mathbf{x}_k$.

The pseudoinverse allows for handling manipulators with up to 5 DOF. In the case of manipulators with 6 DOF, $\mathbf{J}_{\text{IK},k}^\dagger$ becomes the inverse $\mathbf{J}_{\text{IK},k}^{-1}$. Although the approach usually shows a fast convergence rate, it does not work when $\mathbf{J}_{\text{IK},k}$ turns singular. A possibility to overcome this problem is the use of the transpose $\mathbf{J}_{\text{IK},k}^T$ in (4.34):

$$\Delta \mathbf{q}_k = \mathbf{K}_{\text{tr}} \mathbf{J}_{\text{IK},k}^T(\mathbf{q}_k) \Delta \mathbf{x}_k. \quad (4.36)$$

In this case, the inversion of the JACOBIAN is not necessary, but the transpose leads to lower convergence rate of the algorithm [BK05]. The utilization of the transpose can be justified using virtual forces [WE84, Bus09, SSVO09, Cor11]. The diagonal matrix \mathbf{K}_{tr} allows for adjusting the step size [GBF85, BK05] and the weight of the translational and the rotational part of $\Delta \mathbf{x}_k$ [SSVO09], similarly as the matrix \mathbf{K}_{ps} in (4.34).

A way to exploit the advantages of these two approaches is to switch between (4.34) and (4.36) depending on the distance to a singularity of $\mathbf{J}_{\text{IK},k}$ and on the algorithm's convergence. This proposed strategy begins using (4.34). When $\mathbf{J}_{\text{IK},k}$ is "near" to be singular or the algorithm diverges, the approach switches to (4.36) in order to overcome the singularity or the divergence phase. Once $\mathbf{J}_{\text{IK},k}$ is not "close" to a singularity and the algorithm converges again, equation (4.36) is employed again to provide higher convergence rate.

The condition number κ of the manipulator JACOBIAN matrix \mathbf{J} is one of the most common methods used in robotics as a metric of the proximity to a singularity [LA16]. This concept can be extended to singularities detection for $\mathbf{J}_{\text{IK},k}$. The condition number κ_{IK} is defined in this case as the product of the norm of the homogeneous matrices $\bar{\mathbf{J}}_{\text{IK},k}$ and $\bar{\mathbf{J}}_{\text{IK},k}^{-1}$ [LA16, SK08, SC82]. If the 2-norm (spectral norm) [Mey00] is used, the condition number equals the ratio between the largest and the smallest singular values (σ_{max} and σ_{min}) of $\bar{\mathbf{J}}_{\text{IK},k}$ [ALC92]:

$$\kappa_{\text{IK}}(\bar{\mathbf{J}}_{\text{IK},k}) = \left\| \bar{\mathbf{J}}_{\text{IK},k} \right\|_2 \left\| \bar{\mathbf{J}}_{\text{IK},k}^{-1} \right\|_2 = \frac{\sigma_{\text{max}}(\bar{\mathbf{J}}_{\text{IK},k})}{\sigma_{\text{min}}(\bar{\mathbf{J}}_{\text{IK},k})}, \quad (4.37)$$

where $1 \leq \kappa_{\text{IK}} \leq \infty$. $\kappa_{\text{IK}} = \infty$ corresponds to a singularity of $\bar{\mathbf{J}}_{\text{IK},k}$. Due to the dimensional inhomogeneities of $\mathbf{J}_{\text{IK},k}$, the use of the dimensionally homogeneous matrix $\bar{\mathbf{J}}_{\text{IK},k}$ is mandatory to calculate κ_{IK} in (4.37). This can be achieved through the characteristic length L_c , as shown in (2.26) and (2.27). Experiments with several manipulators of different DOF have shown that appropriate threshold values for switching are $200 \leq \kappa_{\text{IK}} \leq 600$.

Other norms can be applied to calculate the condition number in (4.37), e. g. the Frobenious norm, the 1-norm, or the ∞ -norm [ALC92, KHO12, Mer06]. However, taking into account the equivalence of norms in finite dimensional vector spaces [Wer11], it is expected that they produce similar results. For this reason, the use of other norms is not studied in this work.

The proposed approach to solve the inverse kinematics problem is introduced in algorithm 4.1. A 6 DOF manipulator and a randomly generated desired EE pose are used as example to show the capability of the proposed approach. The DH parameters of the exemplary manipulator are listed in figure 4.7. The homogeneous transformation matrix

$${}^0\mathbf{T}_{\text{EE}^*} = \begin{pmatrix} 0.5754 & 0.8033 & 0.1537 & 0.0208 \\ 0.4873 & -0.1858 & -0.8532 & -0.0028 \\ -0.6569 & 0.5658 & -0.4984 & 1.3158 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.38)$$

represents the desired EE pose, which is within the robot's workspace.

For this example, the threshold value κ_{IK}^* is set to $\kappa_{\text{IK}}^* = 500$. The matrices \mathbf{K}_{ps} and \mathbf{K}_{tr} are established as

$$\mathbf{K}_{\text{ps}} = \mathbf{K}_{\text{tr}} = \text{diag}(\lambda), \quad (4.39)$$

where the initial value of $\lambda = 1$ is modified during the algorithm in the same way as in Corke's robotics toolbox [Cor11]:

$$\lambda_i = \begin{cases} 0.5 \lambda_{i-1} & \text{when } \|\Delta \mathbf{x}_k\| > \|\Delta \mathbf{x}_{k-1}\|, \\ 1.09 \lambda_{i-1} & \text{when } \|\Delta \mathbf{x}_k\| \leq \|\Delta \mathbf{x}_{k-1}\|. \end{cases} \quad (4.40)$$

Algorithm 4.1: Solution of the inverse kinematics for a general serial manipulator

Input: $\mathbf{q}_0, \mathbf{x}_{EE}^*, \epsilon, \kappa_{IK}^*$

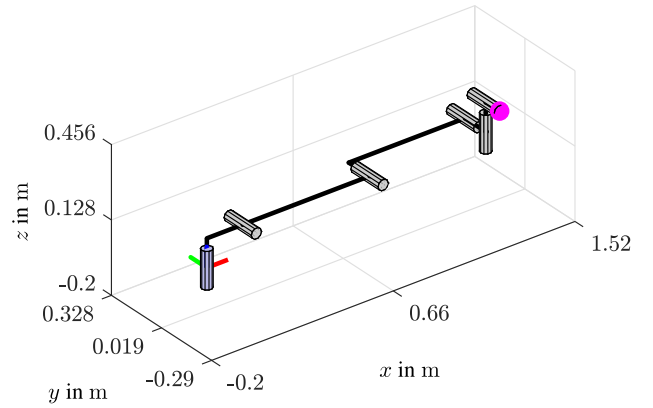
```

1  $k = 0$ 
2  $\mathbf{x}_{EE,0} = \mathbf{f}(\mathbf{q}_0)$ 
3  $\Delta \mathbf{x}_0 = (\Delta \mathbf{r}_0^T, \Delta \boldsymbol{\psi}_0^T)^T$ 
4 while  $\Delta \mathbf{x}_k > \epsilon$  do
5   if  $\kappa(\bar{\mathbf{J}}_{IK,k}) < \kappa_{IK}^*$  then // see (4.37)
6     if  $\|\Delta \mathbf{x}_k\| < \|\Delta \mathbf{x}_{k-1}\|$  then
7        $\Delta \mathbf{q}_k = \mathbf{K}_{ps} \mathbf{J}_{IK,k}^\dagger(\mathbf{q}_k) \Delta \mathbf{x}_k$  // see (4.34)
8     else
9        $\Delta \mathbf{q}_k = \mathbf{K}_{tr} \mathbf{J}_{IK,k}^T(\mathbf{q}_k) \Delta \mathbf{x}_k$  // see (4.36)
10    end
11  else
12     $\Delta \mathbf{q}_k = \mathbf{K}_{tr} \mathbf{J}_{IK,k}^T(\mathbf{q}_k) \Delta \mathbf{x}_k$  // see (4.36)
13  end
14   $\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta \mathbf{q}_k$  // see (4.19)
15   $\mathbf{x}_{EE,k+1} = \mathbf{f}(\mathbf{q}_{k+1})$  // see (4.20)
16   $\Delta \mathbf{x}_{k+1} = (\Delta \mathbf{r}_{k+1}^T, \Delta \boldsymbol{\psi}_{k+1}^T)^T$  // see (4.21), (4.22), and (4.29)
17   $k = k + 1$ 
18 end
19  $\mathbf{q}_{calc} = \mathbf{q}_{k+1}$ 

```

Output: \mathbf{q}_{calc}

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
R	q_1	0.128	0.170	$\pi/2$
R	q_2	0.000	0.600	0
R	q_3	-0.128	0.550	0
R	q_4	0.128	0.000	$-\pi/2$
R	q_5	0.128	0.000	$\pi/2$
R	q_6	0.090	0.000	0

**Figure 4.7:** Manipulator used as example for the inverse kinematics

The initial values for the joint coordinates is exemplarily set to 40° (0.6981 rad) for all joints:

$$\mathbf{q}_0 = (0.6981, 0.6981, 0.6981, 0.6981, 0.6981, 0.6981)^T \text{ in rad.} \quad (4.41)$$

The joint coordinates obtained with the proposed approach are

$$\mathbf{q}_{calc} = (1.4774, 1.8058, -0.4353, -0.8333, -1.3362, -0.9979)^T \text{ in rad.} \quad (4.42)$$

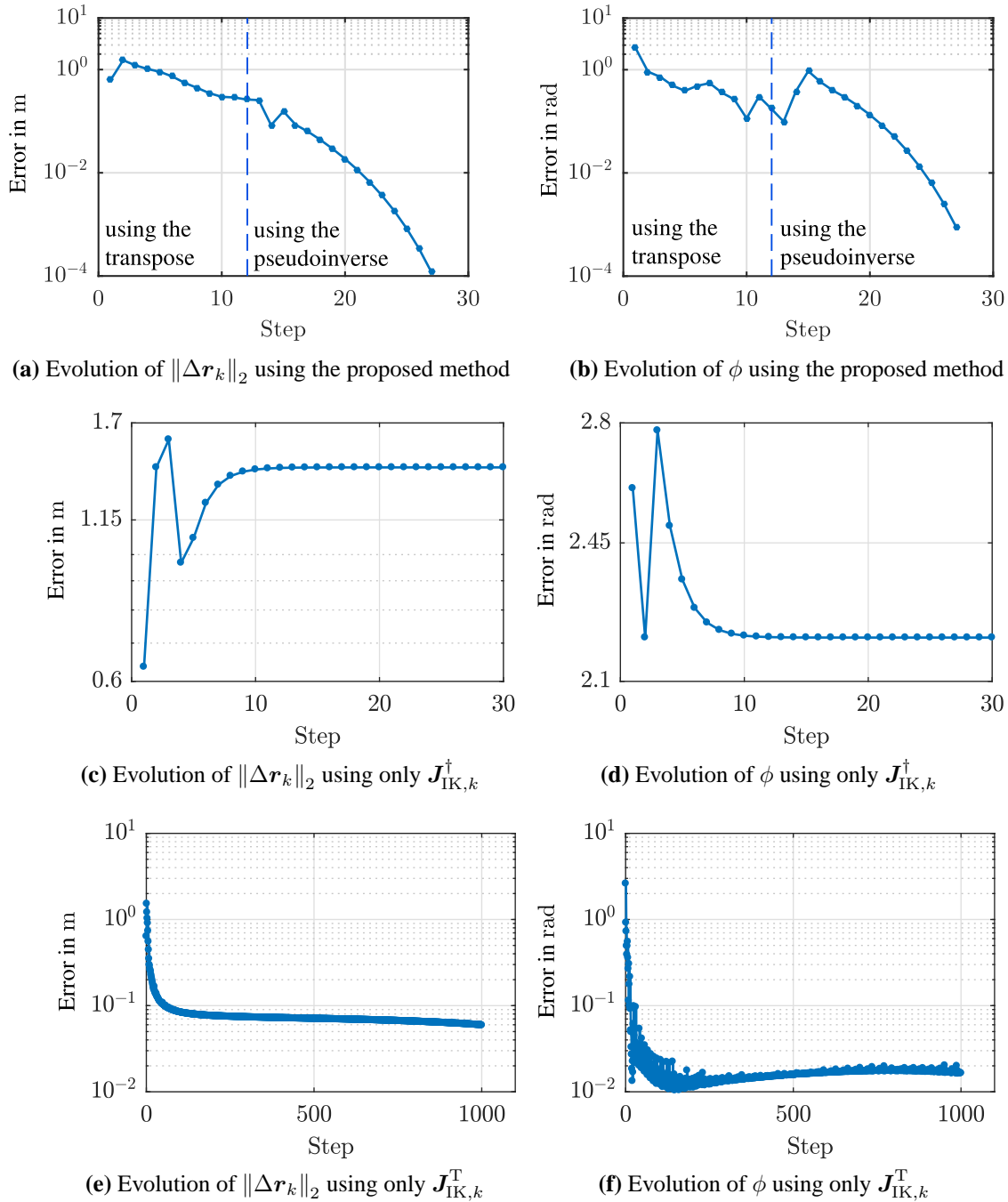


Figure 4.8: Evolution of the position error $\|\Delta \mathbf{r}_k\|_2$ and orientation error ϕ throughout the solution of the inverse kinematics problem

The performance of the approach is illustrated in figure 4.8. In figure 4.8(a) and figure 4.8(b) the convergence rate increases after the 12-th iteration. The position error in figure 4.8(a) and figure 4.8(c) is quantified through the 2-norm (euclidean norm) of the position error vector $\|\Delta \mathbf{r}_k\|_2$ (see (4.22)). The orientation error is quantified in figure 4.8(b) and figure 4.8(d) using the angle ϕ from the axis-angle representation (see (4.24)).

Figure 4.8(c) and figure 4.8(d) demonstrate how the algorithm stagnates when only $\mathbf{J}_{\text{IK},k}^\dagger$ is used to solve the inverse kinematics problem. This occurred despite the reduction of the step size within the algorithm modifying λ (see figure 4.9(b) as well as equation (4.34), (4.39), and (4.40)). Using $\mathbf{J}_{\text{IK},k}^\text{T}$ the algorithm can overcome the singularity as shown in figure 4.8(e) and figure 4.8(f), however the convergence rate is too slow in comparison with the pseudoinverse.

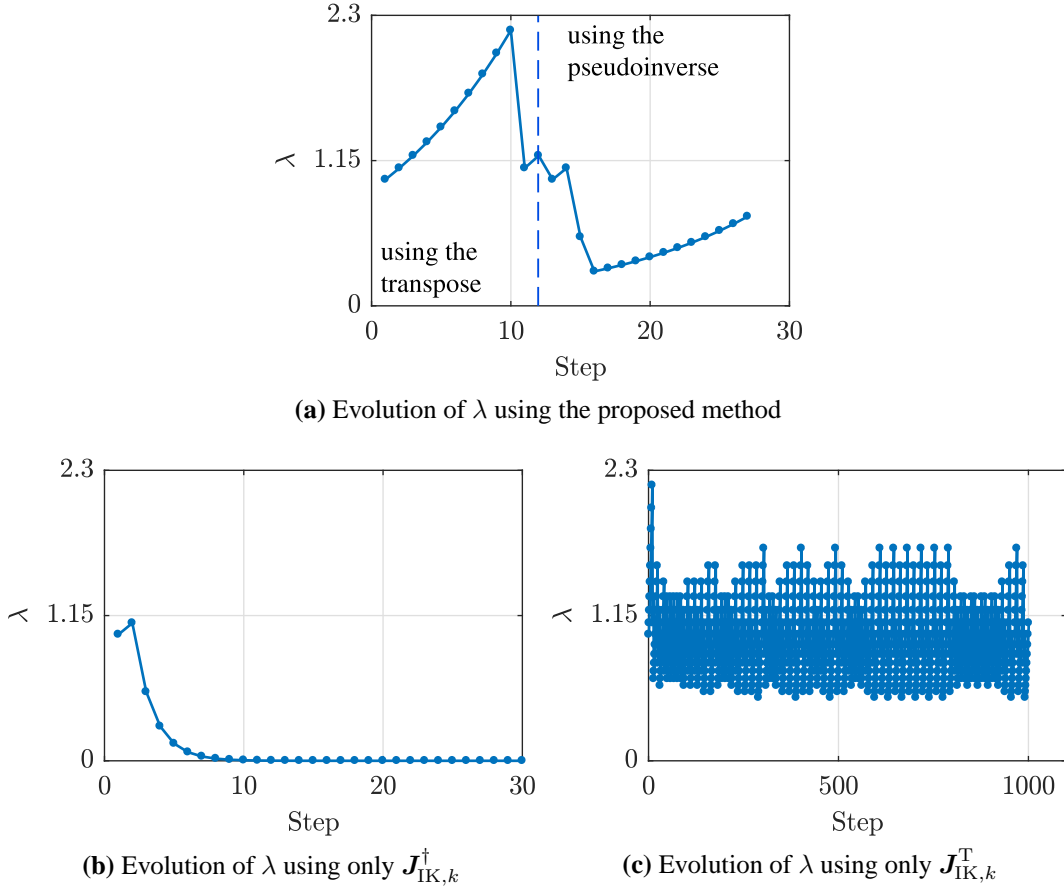


Figure 4.9: Evolution of λ throughout the solution of the inverse kinematics problem

Figure 4.10 presents the evolution of the condition number of $\bar{\mathbf{J}}_{\text{IK},k}$ when the proposed approach is used. Although the method begins using (4.34), it switches to (4.36) in the first iteration because $\kappa_{\text{IK}}(\bar{\mathbf{J}}_{\text{IK},k}) > \kappa_{\text{IK}}^*$. In the 12-th iteration, after the algorithm converges and $\kappa_{\text{IK}}(\bar{\mathbf{J}}_{\text{IK},k}) < \kappa_{\text{IK}}^*$, it switches to use (4.34) improving the convergence rate (see figures 4.8(a) and 4.8(b)).

4.5.2 Derivation of the JACOBIAN matrix for the inverse kinematics

In order to obtain the JACOBIAN for the inverse kinematics $\mathbf{J}_{\text{IK},k}$ shown in (4.31), equation (4.22) and (4.30) have to be derived with respect to the joint coordinates \mathbf{q} . The derivation of (4.22) corresponds to the EE linear velocity, however, deriving (4.30) does not results in the EE angular velocity. Hence, this section addresses the derivation of the Rodrigues vector ${}_{(W)}\Delta\psi_k$. First of

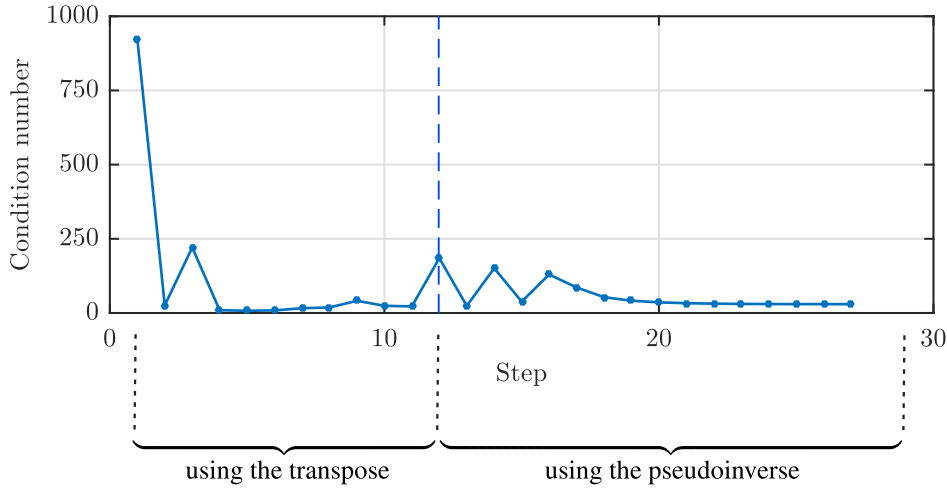


Figure 4.10: Evolution of $\kappa(\bar{\mathbf{J}}_{\text{IK},k})$ throughout the solution of the inverse kinematics problem using the proposed approach

all, the following matrices properties are useful for the derivation of $\mathbf{J}_{\text{IK},k}$. They are proofed in [Ang85].

For a vector $\boldsymbol{\delta} = (\delta_x, \delta_y, \delta_z)^\text{T}$, its associated skew-symmetric matrix is

$$\mathbf{S}(\boldsymbol{\delta}) = \begin{pmatrix} 0 & -\delta_z & \delta_y \\ \delta_z & 0 & -\delta_x \\ -\delta_y & \delta_x & 0 \end{pmatrix}. \quad (4.43)$$

For the rotation matrix ${}^{i-1}\mathbf{R}_i$ describing the orientation of $(\text{RF})_i$ with respect to $(\text{RF})_{i-1}$ in link i :

$$\frac{\partial {}^{i-1}\mathbf{R}_i}{\partial q_i} = \mathbf{E} {}^{i-1}\mathbf{R}_i, \quad (4.44)$$

where \mathbf{E} is the skew-symmetric matrix associated to $\boldsymbol{\delta} = (0, 0, 1)^\text{T}$,

$$\mathbf{E} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (4.45)$$

because the joint axis i is collinear with the z -axis of $(\text{RF})_{i-1}$.

Additionally, the following matrices properties are valid for a rotation matrix \mathbf{R} , a skew-symmetric matrix \mathbf{W} , any matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{3 \times 3}$, any vectors $\mathbf{a} \in \mathbb{R}^{n \times 1}$, $\mathbf{b} \in \mathbb{R}^{n \times 1}$, $\mathbf{c} \in \mathbb{R}^{n \times 1}$, the identity matrix $\mathbf{I} \in \mathbb{R}^{3 \times 3}$:

$$\mathbf{S}(\text{vect}(\mathbf{R})) = \frac{1}{2}(\mathbf{R} - \mathbf{R}^\text{T}). \quad (4.46)$$

$$\mathbf{W} \text{vect}(\mathbf{R}) = -\mathbf{S}(\text{vect}(\mathbf{R})) \text{vect}(\mathbf{W}). \quad (4.47)$$

$$\text{tr}(\mathbf{B}) = \text{tr}(\mathbf{B}^T) \quad \text{and} \quad \text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA}) . \quad (4.48)$$

$$\text{vect}(\mathbf{A}) = -\text{vect}(\mathbf{A}^T) . \quad (4.49)$$

$$(\mathbf{a}^T \mathbf{b}) \mathbf{c} = (\mathbf{c} \mathbf{a}^T) \mathbf{b} . \quad (4.50)$$

$$\text{tr}(\mathbf{SB}) = 2(\text{vect}(\mathbf{S}))^T \text{vect}(\mathbf{B}^T) = -2(\text{vect}(\mathbf{S}))^T \text{vect}(\mathbf{B}) . \quad (4.51)$$

$$\text{vect}(\mathbf{RBR}^T) = \mathbf{R} \text{vect}(\mathbf{B}) . \quad (4.52)$$

$$\text{vect}(\mathbf{SB}) = \frac{1}{2}(\mathbf{I} \text{tr}(\mathbf{B}) - \mathbf{B}) \text{vect}(\mathbf{S}) , \quad (4.53)$$

For reasons of clarity, equation (4.30) is repeated:

$${}^{(W)}\Delta\psi_k = \frac{2}{1 + \text{tr}({}^c\mathbf{R}_d)} {}^W\mathbf{R}_{EE,k} \text{vect}({}^c\mathbf{R}_d) . \quad (4.54)$$

Deriving the last expression with respect to \mathbf{q}

$$\frac{\partial {}^{(W)}\Delta\psi_k}{\partial \mathbf{q}} = \frac{\partial}{\partial \mathbf{q}} \left(\underbrace{\frac{2}{1 + \text{tr}({}^c\mathbf{R}_d)}}_b \underbrace{{}^W\mathbf{R}_{EE,k}}_A \underbrace{\text{vect}({}^c\mathbf{R}_d)}_c \right) . \quad (4.55)$$

The terms b , \mathbf{A} , and \mathbf{c} are used in this section only to enhance the readability.

For each element q_i of \mathbf{q} , the last expression is written as

$$\frac{\partial {}^{(W)}\Delta\psi_k}{\partial q_i} = \frac{\partial b}{\partial q_i} \mathbf{A} \mathbf{c} + b \frac{\partial \mathbf{A}}{\partial q_i} \mathbf{c} + b \mathbf{A} \frac{\partial \mathbf{c}}{\partial q_i} . \quad (4.56)$$

Each term in (4.56) is individually calculated below.

To calculate $\partial b/\partial q_i$:

$$\begin{aligned} \frac{\partial b}{\partial q_i} &= \frac{\partial}{\partial q_i} \left(\frac{2}{1 + \text{tr}({}^c\mathbf{R}_d)} \right) \\ &= \frac{-2}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \frac{\partial (\text{tr}({}^c\mathbf{R}_d))}{\partial q_i} \\ &= \frac{-2}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \text{tr} \left(\frac{\partial {}^c\mathbf{R}_d}{\partial q_i} \right) \end{aligned}$$

$$\begin{aligned}
\frac{\partial b}{\partial q_i} &= \frac{-2}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \text{tr} \left(\frac{\partial}{\partial q_i} ({}^W\mathbf{R}_{EE,k}^T {}^W\mathbf{R}_{EE*}) \right) \\
&= \frac{-2}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \text{tr} \left(\frac{\partial}{\partial q_i} ({}^{n-1}\mathbf{R}_n^T \dots {}^W\mathbf{R}_0^T {}^W\mathbf{R}_{EE*}) \right) \\
&= \frac{-2}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \text{tr} \left({}^{n-1}\mathbf{R}_n^T \dots {}^i\mathbf{R}_{i+1}^T \frac{\partial {}^{i-1}\mathbf{R}_i^T}{\partial q_i} {}^{i-2}\mathbf{R}_{i-1}^T \dots {}^W\mathbf{R}_0^T {}^W\mathbf{R}_{EE*} \right)
\end{aligned} \tag{4.57}$$

Using (4.44):

$$\begin{aligned}
\frac{\partial b}{\partial q_i} &= \frac{-2}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \text{tr} \left({}^{n-1}\mathbf{R}_n^T \dots {}^i\mathbf{R}_{i+1}^T (E^{i-1}\mathbf{R}_i)^T {}^{i-2}\mathbf{R}_{i-1}^T \dots {}^W\mathbf{R}_0^T {}^W\mathbf{R}_{EE*} \right) \\
&= \frac{-2}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \text{tr} \left(({}^W\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1} E^{i-1}\mathbf{R}_i {}^i\mathbf{R}_{i+1} \dots {}^{n-1}\mathbf{R}_n)^T {}^W\mathbf{R}_{EE*} \right) \\
\frac{\partial b}{\partial q_i} &= \frac{-2}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \text{tr} \left(\underbrace{({}^W\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1} E ({}^W\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1}))^T}_W \dots {}^{n-1}\mathbf{R}_n \right)^T {}^W\mathbf{R}_{EE*} \\
&= \frac{-2}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \text{tr} \left((W {}^W\mathbf{R}_{EE,k})^T {}^W\mathbf{R}_{EE*} \right)
\end{aligned} \tag{4.58}$$

Equation (4.48) can be employed to rewrite the last equation as

$$\frac{\partial b}{\partial q_i} = \frac{-2}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \text{tr} \left(W {}^W\mathbf{R}_{EE,k} {}^W\mathbf{R}_{EE*}^T \right)$$

Since W is a skew-symmetric matrix, (4.51) can be used in order to obtain:

$$\frac{\partial b}{\partial q_i} = \frac{-2}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \left(-2(\text{vect}(W))^T \text{vect}({}^W\mathbf{R}_{EE,k} {}^W\mathbf{R}_{EE*}^T) \right) \tag{4.59}$$

The vector invariant of the matrix \mathbf{W} can be calculated using (4.52):

$$\begin{aligned} \text{vect}(\mathbf{W}) &= \text{vect}\left({}^{\mathbf{W}}\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1} \mathbf{E} \left({}^{\mathbf{W}}\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1}\right)^{\text{T}}\right) \\ &= {}^{\mathbf{W}}\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1} \text{vect}(\mathbf{E}) \\ &= {}^{\mathbf{W}}\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1} ({}^{i-1})\mathbf{e}_z^{(i)} \end{aligned} \quad (4.60)$$

This is replaced in (4.59) to obtain

$$\begin{aligned} \frac{\partial b}{\partial q_i} &= \frac{4}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \left(\left({}^{\mathbf{W}}\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1} ({}^{i-1})\mathbf{e}_z^{(i)}\right)^{\text{T}} \text{vect}\left({}^{\mathbf{W}}\mathbf{R}_{\text{EE},k} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*}^{\text{T}}\right) \right) \\ \frac{\partial b}{\partial q_i} &= \frac{4}{(1 + \text{tr}({}^c\mathbf{R}_d))^2} \left(\text{vect}\left({}^{\mathbf{W}}\mathbf{R}_{\text{EE},k} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*}^{\text{T}}\right)^{\text{T}} \left({}^{\mathbf{W}}\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1} ({}^{i-1})\mathbf{e}_z^{(i)}\right) \right) \end{aligned} \quad (4.61)$$

To calculate $\partial \mathbf{c} / \partial q_i$:

$$\begin{aligned} \frac{\partial \mathbf{c}}{\partial q_i} &= \frac{\partial}{\partial q_i} \left(\text{vect}({}^c\mathbf{R}_d) \right) = \frac{\partial}{\partial q_i} \left(\text{vect}\left({}^{\mathbf{W}}\mathbf{R}_{\text{EE},k}^{\text{T}} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*}\right) \right) \\ &= \frac{\partial}{\partial q_i} \left(\text{vect}\left({}^{n-1}\mathbf{R}_n^{\text{T}} \dots {}^{\mathbf{W}}\mathbf{R}_0^{\text{T}} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*}\right) \right) = \text{vect} \left(\frac{\partial}{\partial q_i} \left({}^{n-1}\mathbf{R}_n^{\text{T}} \dots {}^{\mathbf{W}}\mathbf{R}_0^{\text{T}} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*} \right) \right) \end{aligned}$$

This can be solved in the same way as in (4.57) and (4.58) to obtain

$$\begin{aligned} \frac{\partial \mathbf{c}}{\partial q_i} &= \text{vect} \left(\left(\mathbf{W} {}^{\mathbf{W}}\mathbf{R}_{\text{EE},k} \right)^{\text{T}} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*} \right) = \text{vect} \left(\left(\mathbf{W} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*} {}^c\mathbf{R}_d^{\text{T}} \right)^{\text{T}} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*} \right) \\ &= \text{vect} \left({}^c\mathbf{R}_d {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*}^{\text{T}} \mathbf{W}^{\text{T}} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*} \right) \end{aligned}$$

Using (4.49), the last equation is modified as

$$\frac{\partial \mathbf{c}}{\partial q_i} = -\text{vect} \left({}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*}^{\text{T}} \mathbf{W} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*} {}^c\mathbf{R}_d^{\text{T}} \right)$$

Since ${}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*}^{\text{T}} \mathbf{W} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*}$ is skew-symmetric, equation (4.53) can be used to obtain

$$\frac{\partial \mathbf{c}}{\partial q_i} = -\frac{1}{2} \left(\mathbf{I} \text{tr}({}^c\mathbf{R}_d^{\text{T}}) - {}^c\mathbf{R}_d^{\text{T}} \right) \text{vect} \left({}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*}^{\text{T}} \mathbf{W} {}^{\mathbf{W}}\mathbf{R}_{\text{EE}^*} \right).$$

Applying (4.52) and (4.60), it results

$$\begin{aligned} \frac{\partial \mathbf{c}}{\partial q_i} &= -\frac{1}{2} \left(\mathbf{I} \operatorname{tr}({}^c\mathbf{R}_d^T) - {}^c\mathbf{R}_d^T \right) {}^W\mathbf{R}_{EE^*}^T \operatorname{vect}(\mathbf{W}) \\ &= -\frac{1}{2} \left(\mathbf{I} \operatorname{tr}({}^c\mathbf{R}_d^T) - {}^c\mathbf{R}_d^T \right) {}^W\mathbf{R}_{EE^*}^T \left({}^W\mathbf{R}_1 \dots {}^{i-2}\mathbf{R}_{i-1} \right) {}_{(i-1)}\mathbf{e}_z^{(i)}. \end{aligned} \quad (4.62)$$

To calculate $\partial A / \partial q_i$:

$$\begin{aligned} \frac{\partial A}{\partial q_i} &= \frac{\partial}{\partial q_i} ({}^W\mathbf{R}_{EE,k}) \\ &= \frac{\partial}{\partial q_i} ({}^W\mathbf{R}_0^T \dots {}^{n-1}\mathbf{R}_n^T) \end{aligned}$$

Using the same procedure as in (4.57) and (4.58), the last equation can be rewritten as

$$\frac{\partial A}{\partial q_i} = \mathbf{W} {}^W\mathbf{R}_{EE,k} \quad (4.63)$$

Now, the first term of the sum in (4.56), can be calculated:

$$\begin{aligned} \frac{\partial b}{\partial q_i} \mathbf{A} \mathbf{c} &= \frac{4}{(1 + \operatorname{tr}({}^c\mathbf{R}_d))^2} \left(\operatorname{vect}({}^W\mathbf{R}_{EE,k} {}^W\mathbf{R}_{EE^*}^T) \right)^T \left({}^W\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1} {}_{(i-1)}\mathbf{e}_z^{(i)} \right) \\ &\quad {}^W\mathbf{R}_{EE,k} \operatorname{vect}({}^c\mathbf{R}_d) \end{aligned}$$

Using the property presented in (4.50):

$$\begin{aligned} \frac{\partial b}{\partial q_i} \mathbf{A} \mathbf{c} &= \frac{4}{(1 + \operatorname{tr}({}^c\mathbf{R}_d))^2} {}^W\mathbf{R}_{EE,k} \left(\operatorname{vect}({}^c\mathbf{R}_d) \operatorname{vect}({}^W\mathbf{R}_{EE,k} {}^W\mathbf{R}_{EE^*}^T) \right)^T \\ &\quad \left({}^W\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1} \right) {}_{(i-1)}\mathbf{e}_z^{(i)} \\ \frac{\partial b}{\partial q_i} \mathbf{A} \mathbf{c} &= \frac{4}{(1 + \operatorname{tr}({}^c\mathbf{R}_d))^2} {}^W\mathbf{R}_{EE,k} \left(\operatorname{vect}({}^c\mathbf{R}_d) \operatorname{vect}({}^W\mathbf{R}_{EE^*} {}^c\mathbf{R}_d^T {}^W\mathbf{R}_{EE^*}^T) \right)^T \\ &\quad \left({}^W\mathbf{R}_0 \dots {}^{i-2}\mathbf{R}_{i-1} \right) {}_{(i-1)}\mathbf{e}_z^{(i)} \end{aligned} \quad (4.64)$$

For the second term of the sum in (4.50):

$$\begin{aligned} b \frac{\partial \mathbf{A}}{\partial q_i} \mathbf{c} &= \frac{2}{1 + \text{tr}({}^c \mathbf{R}_d)} \mathbf{W} {}^W \mathbf{R}_{EE,k} \text{vect}({}^c \mathbf{R}_d) \\ &= \frac{2}{1 + \text{tr}({}^c \mathbf{R}_d)} \mathbf{W} {}^W \mathbf{R}_{EE,k} \text{vect}({}^W \mathbf{R}_{EE,k}^T {}^W \mathbf{R}_{EE*}) \end{aligned}$$

Applying (4.52) results

$$\begin{aligned} b \frac{\partial \mathbf{A}}{\partial q_i} \mathbf{c} &= \frac{2}{1 + \text{tr}({}^c \mathbf{R}_d)} \mathbf{W} \text{vect}({}^W \mathbf{R}_{EE,k} {}^W \mathbf{R}_{EE,k}^T {}^W \mathbf{R}_{EE*} {}^W \mathbf{R}_{EE,k}^T) \\ &= \frac{2}{1 + \text{tr}({}^c \mathbf{R}_d)} \mathbf{W} \text{vect}({}^W \mathbf{R}_{EE*} {}^W \mathbf{R}_{EE,k}^T) \end{aligned}$$

Using (4.46) and (4.47)

$$\begin{aligned} b \frac{\partial \mathbf{A}}{\partial q_i} \mathbf{c} &= \frac{2}{1 + \text{tr}({}^c \mathbf{R}_d)} \left(-\mathbf{S}(\text{vect}({}^W \mathbf{R}_{EE*} {}^W \mathbf{R}_{EE,k}^T)) \right) \text{vect}(\mathbf{W}) \\ &= \frac{2}{1 + \text{tr}({}^c \mathbf{R}_d)} \left(-\frac{1}{2} \right) \left({}^W \mathbf{R}_{EE*} {}^W \mathbf{R}_{EE,k}^T - {}^W \mathbf{R}_{EE,k} {}^W \mathbf{R}_{EE*}^T \right) \text{vect}(\mathbf{W}) \quad (4.65) \end{aligned}$$

Applying (4.60), it can be rewritten as

$$\begin{aligned} b \frac{\partial \mathbf{A}}{\partial q_i} \mathbf{c} &= \frac{-1}{1 + \text{tr}({}^c \mathbf{R}_d)} \left({}^W \mathbf{R}_{EE*} {}^W \mathbf{R}_{EE,k}^T - {}^W \mathbf{R}_{EE,k} {}^W \mathbf{R}_{EE*}^T \right) ({}^W \mathbf{R}_0 \dots {}^{i-2} \mathbf{R}_{i-1})_{(i-1)} \mathbf{e}_z^{(i)} \\ &= \frac{-1}{1 + \text{tr}({}^c \mathbf{R}_d)} \left({}^W \mathbf{R}_{EE*} {}^c \mathbf{R}_d {}^W \mathbf{R}_{EE*}^T - {}^W \mathbf{R}_{EE*} {}^c \mathbf{R}_d^T {}^W \mathbf{R}_{EE*}^T \right) ({}^W \mathbf{R}_0 \dots {}^{i-2} \mathbf{R}_{i-1})_{(i-1)} \mathbf{e}_z^{(i)} \quad (4.66) \end{aligned}$$

For the third term of the sum in (4.50):

$$\begin{aligned} b \mathbf{A} \frac{\partial \mathbf{c}}{\partial q_i} &= \frac{2}{1 + \text{tr}({}^c \mathbf{R}_d)} {}^W \mathbf{R}_{EE,k} \left(-\frac{1}{2} \right) \left(\mathbf{I} \text{tr}({}^c \mathbf{R}_d^T) - {}^c \mathbf{R}_d^T \right) {}^W \mathbf{R}_{EE*}^T \\ &\quad \left({}^W \mathbf{R}_0 \dots {}^{i-2} \mathbf{R}_{i-1} \right)_{(i-1)} \mathbf{e}_z^{(i)} \\ &= \frac{-1}{1 + \text{tr}({}^c \mathbf{R}_d)} {}^W \mathbf{R}_{EE,k} \left(\mathbf{I} \text{tr}({}^c \mathbf{R}_d^T) - {}^c \mathbf{R}_d^T \right) {}^W \mathbf{R}_{EE*}^T \left({}^W \mathbf{R}_0 \dots {}^{i-2} \mathbf{R}_{i-1} \right)_{(i-1)} \mathbf{e}_z^{(i)} \quad (4.67) \end{aligned}$$

Replacing (4.64), (4.66), and (4.67) in (4.56) results

$$\begin{aligned} \frac{\partial_{(W)} \Delta \psi_k}{\partial q_i} = & \left[\frac{4}{(1 + \text{tr}({}^c \mathbf{R}_d))^2} {}^W \mathbf{R}_{EE,k} \left(\text{vect}({}^c \mathbf{R}_d) \text{vect}({}^W \mathbf{R}_{EE*} {}^c \mathbf{R}_d^T {}^W \mathbf{R}_{EE*}^T)^T \right) \right. \\ & + \frac{-1}{1 + \text{tr}({}^c \mathbf{R}_d)} \left({}^W \mathbf{R}_{EE*} {}^c \mathbf{R}_d {}^W \mathbf{R}_{EE*}^T - {}^W \mathbf{R}_{EE*} {}^c \mathbf{R}_d^T {}^W \mathbf{R}_{EE*}^T \right) \\ & \left. + \frac{-1}{1 + \text{tr}({}^c \mathbf{R}_d)} {}^W \mathbf{R}_{EE,k} \left(\mathbf{I} \text{tr}({}^c \mathbf{R}_d^T) - {}^c \mathbf{R}_d^T \right) {}^W \mathbf{R}_{EE*}^T \right] \\ & \left({}^W \mathbf{R}_0 \dots {}^{i-2} \mathbf{R}_{i-1} \right)_{(i-1)} \mathbf{e}_z^{(i)}, \end{aligned}$$

which corresponds to the expression in (4.33).

4.5.3 Collision evaluation

Besides reachability, a basic requirement in robot synthesis is the absence of collisions between the links during the execution of the path. Since the robot synthesis is carried out in an early design stage, the geometry of the links is not completely defined, hampering the evaluation of possible collisions. Nonetheless, numerous experiments have shown that a simplified collision evaluation helps to produce more feasible results. For this purpose, the geometry is approximated using cylinders as illustrated in figure 4.11 and the method introduced in [Mey01] is applied to determine collisions between links. Although the approach was developed for parallel manipulators [Kot13, Kob15], it can also be applied to serial mechanisms. Moreover, this provides a computational-efficient method that can be used during the optimization.

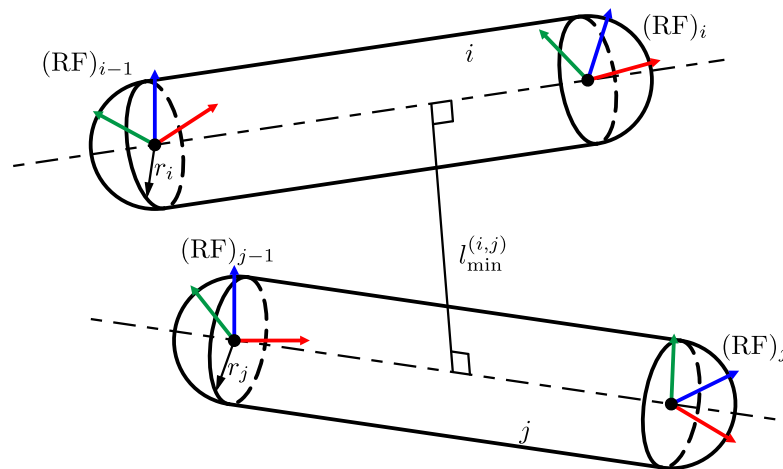


Figure 4.11: Simplification of the robot links geometry for collision evaluation

Given two cylinders, i and j , with radius r_i and r_j respectively, the problem is reduced to determine the minimum distance between the cylinders axes $l_{\min}^{(i,j)}$. The collision occurs when

$$l_{\min}^{(i,j)} \leq r_i + r_j. \quad (4.68)$$

The detailed procedure to calculate $l_{\min}^{(i,j)}$ is explained in [Mey01, Kot13, Kob15]. Further collision detection methods can be found in [MD06, Mer06] and will not be discussed here because their comparison is not part of the present work.

Possible collisions are evaluated during the task execution, i. e. not only in the desired poses, but also in the path between them. Given a robot configuration, condition (4.68) is evaluated between all non-sequential links, i. e. for

$$i \neq j, \quad i \neq j + 1, \quad \text{and} \quad i \neq j - 1. \quad (4.69)$$

In order to include the collision evaluation in the optimization, the function c_{coll} is used as constraint in the optimization. This is defined as

$$c_{\text{coll}}(\mathbf{p}) = \begin{cases} -1 & \text{if no collision is found,} \\ 1 & \text{if any collision is found.} \end{cases} \quad (4.70)$$

The introduced approach for the detection of collisions between links can be extended to handle external obstacles or place restrictions, e. g. walls or machines in the neighborhood.

4.5.4 Dynamics

Several toolboxes to calculate the dynamics of serial manipulators have already been developed [DLNK12]. Nevertheless, they can not be used to model the generated architectures because they are based on numeric computation or are intended for manipulators with constant geometric parameters. Another alternative is the calculation of the dynamics equations in closed-form. This allows, compared with numeric computations [Cor17], for a significant reduction of the calculation time [TK11, KB06, TKHO09]. However, existing symbolic computation tools for robot modeling [DLNK12, Wal07, CB17] are able to generate dynamic equations only for a single architecture.

In the modeling phase (see figure 4.1), the closed-form equations of the forward kinematics (2.13), the JACOBIAN \mathbf{J} (2.19), and the inverse dynamics (2.30) (2.45) are determined for each architecture using symbolic calculations. The resulting symbolic expressions are stored as properties of the *SerialChain* class. Then, the MATLAB[®] code for these equations is automatically generated. The generated inverse dynamics function is used in the optimization to reduce the computational effort when the dynamic performance has to be evaluated. For this function, the optimization parameters of each architecture are included as arguments besides the information related to the

desired trajectory $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$, the friction coefficients, the gear ratios, and the physical properties of the robot links (mass, inertia, and center of mass).

Current commercial robots with variable geometry have shown the feasibility of modular manipulators construction. Figure 4.12 shows an example of the robolink W from the company igus [igu17] and the Dextrous Lightweight Arm LWA 4D [LWA17a] from the company Schunk. Since different structures are considered during the the CRS, their physical properties have to be

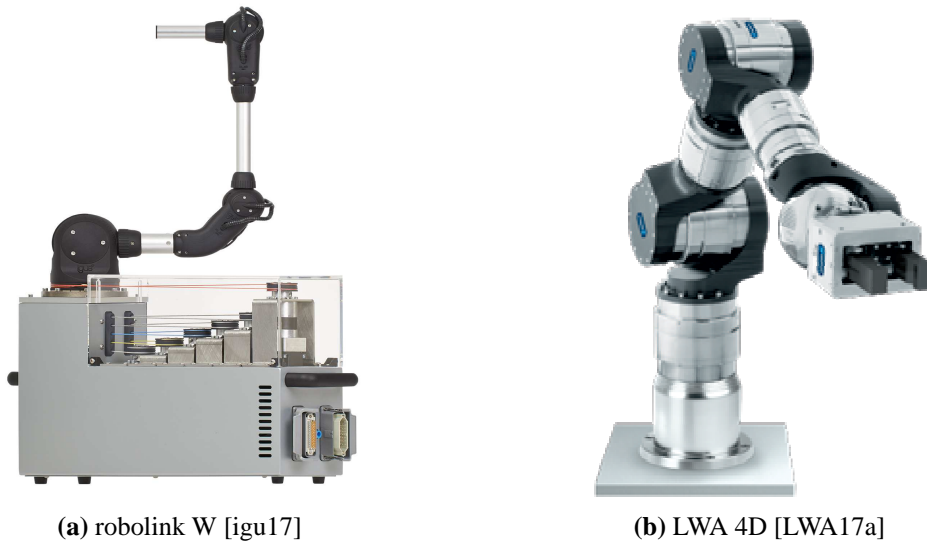


Figure 4.12: Exemplary commercial modular robots

parametrized. Hence, the construction of the robot links is approximated to standard aluminum profiles and the actuators to commercial linear and rotational modules as shown in figure 4.13. Each link is assumed to be formed by two segments, **A** and **D**, and a motor-gearbox assembly **M**. In the case of prismatic joints, **M** is the motor-gearbox assembly and **D** the rail of the linear actuator. Due to its mass, the motor-gearbox assembly has a great influence on the physical properties of the link [ZBH12, ZBH11] and, therefore, is considered in the modeling of the link. Even though the position of the motors essentially depends on the final construction, in the present work, the actuator of the joint $i + 1$ is assumed to be attached to the link i (see figure 4.13). The reference frames $(RF)_i$ and $(RF)_{i-1}$ are placed in the i -link as shown in figure 4.13 and defined in section 2.3. The origin of the reference frame $(RF)_{C_i}$ coincides with the link's center of mass C_i and is parallel to $(RF)_i$. The inertia tensor ${}_{(C_i)}\mathbf{I}_i^{(C_i)}$ is described in $(RF)_{C_i}$.

To determine the suitability of a motor-gearbox combination for the i -th joint, its characteristics are compared to the required actuator torque/force $\tau_{L_i}(t)$ and the required motor torque/force $\tau_{M_i}(t)$ using the following conditions [RJW06, Pet08, PO09]:

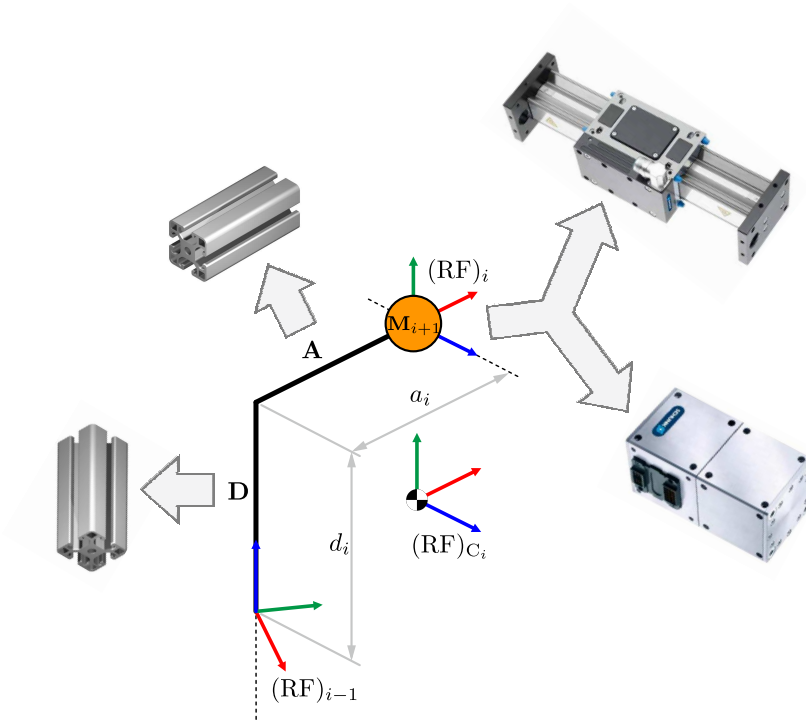


Figure 4.13: Parametrization of a robot link [Sch17b, Sch17d]

1. the allowed peak torque/force of the gearbox $\tau_{L,\text{peak}}^*$ and motor $\tau_{M,\text{peak}}^*$ must be greater than the maximum required (actuator and motor) torques/forces,

$$\tau_{L,\text{peak}}^* \geq \max \left(|\tau_{L_i}(t)| \right); \quad (4.71)$$

$$\tau_{M,\text{peak}}^* \geq \max \left(|\tau_{M_i}(t)| \right); \quad (4.72)$$

2. the nominal torque/force of the gearbox $\tau_{L,\text{nom}}^*$ and motor $\tau_{M,\text{nom}}^*$ must be greater than the rms value of $\tau_{L_i}(t)$ and $\tau_{M_i}(t)$,

$$\tau_{L,\text{nom}}^* \geq \sqrt{\frac{1}{t_N} \int_{t_0}^{t_N-t_0} \tau_{L_i}^2(t) dt}; \quad (4.73)$$

$$\tau_{M,\text{nom}}^* \geq \sqrt{\frac{1}{t_N} \int_{t_0}^{t_{N,\text{task}}} \tau_{M_i}^2(t) dt}; \quad (4.74)$$

3. the maximum allowed speed of the gearbox $\omega_{G,\text{peak}}^*$ and the motor $\omega_{M,\text{peak}}^*$ must be greater than the maximum required joint rate $\dot{q}_i(t)$ and motor rate $\dot{q}_{M_i}(t)$,

$$\omega_{G,\text{peak}}^* \geq \max \left(|\dot{q}_i(t)| \right); \quad (4.75)$$

$$\omega_{M,\text{peak}}^* \geq \max \left(|\dot{q}_{M_i}(t)| \right). \quad (4.76)$$

In the optimization process, the actuators are chosen from a group of predefined rotational and linear actuators. From this group, the components that fulfill (4.71) to (4.76) are selected and the gearbox-motor combination (hereinafter called actuator) for each joint must be selected considering the optimization criteria.

Since the goal is to show the influence of the architecture on the dynamic performance, additional analysis about the motor-gearbox efficiency is not carried out. Friction effects are neglected because they strongly depend on the final construction.

4.6 Postprocessing

As a consequence of construction limitations, it is not possible to realize a manipulator with exactly the same parameters as resulted from the optimization. Therefore, the deviation of the objective function due to variations of the optimization parameters is studied applying Monte Carlo simulation.

For this analysis of the obtained results, each optimization parameter p_w is randomly varied with uniform distribution $\mathcal{U}(\hat{p}_{\min_w}, \hat{p}_{\max_w})$ within the minimum and maximum values \hat{p}_{\min_w} and \hat{p}_{\max_w} . For length parameters, they are defined as

$$\begin{aligned}\hat{p}_{\min_w} &= \tilde{p}_w - 0.05 \bar{L}_{\text{links}}, \\ \hat{p}_{\max_w} &= \tilde{p}_w + 0.05 \bar{L}_{\text{links}},\end{aligned}\tag{4.77}$$

where \bar{L}_{links} is the average length of the links given by

$$\bar{L}_{\text{links}} = \frac{\sum_{i=1}^n |a_i| + |d_i|}{n}\tag{4.78}$$

The boundaries are set relative to the size of the robot in order to employ them in any manipulator. The factor 0.05, corresponding to a variation of 5% of \bar{L}_{links} , can be chosen depending on the application. However, this value is suggested and was used in the examples shown in this work.

Optimization parameters representing angles are not related to the size of the manipulator and, therefore, the values for $\hat{p}_{\min_w}, \hat{p}_{\max_w}$ are established as

$$\begin{aligned}\hat{p}_{\min_w} &= \tilde{p}_w - 0.0278\pi, \\ \hat{p}_{\max_w} &= \tilde{p}_w + 0.0278\pi,\end{aligned}\tag{4.79}$$

where $0.0278\pi \approx 5^\circ$. In the case of the parameters representing the position of the robot's base, their distribution boundaries are

$$\begin{aligned}\hat{p}_{\min_w} &= \tilde{p}_w - 0.05 \text{ m}, \\ \hat{p}_{\max_w} &= \tilde{p}_w + 0.05 \text{ m}.\end{aligned}\tag{4.80}$$

In this point is worth mentioning that varying the robot's base position is equivalent to shifting the complete task. Therefore, the value 0.05 m is use in the examples of this work but can be modified according to the application.

The objective function is evaluated as explained in section 4.4. Since different penalty values are used to deal with the constraints, the sensitivity analysis provides two types of information. The first is the performance variation of the manipulators fulfilling the constraints. The second is the quantity of manipulators that do not fulfill the constraints and which constraint is the most problematic due to geometric variations.

Although the sensitivity analysis is not part of the optimization process, its results are useful to evaluate the robustness of the optimized manipulators providing more information in the selection of the most appropriate architecture for the desired task.

5 Validation examples

In order to demonstrate the capability of the proposed CRS, the present chapter shows three examples of its application. The first two examples in section 5.1 deal with the same task, however, in the first one, only kinematic performance is considered. In the second optimization, minimal kinematic performance is considered as constraint, while the manipulator's cost function is with respect to dynamic performance. In both cases the task is executed by a 4 DOF manipulator. As final example, 6 DOF manipulators are optimized in section 5.2 with respect to a kinematic performance index.

5.1 Mechanisms with reduced number of DOF

The task shown in table 5.1 is used in this section as an example of the CRS for applications requiring less than 6 DOF. For this task, two manipulators will be synthesized: the first considering kinematic performance, the second considering dynamic performance.

Table 5.1: Coordinates of the desired task (4 DOF)

Pose	${}^{(W)}r_x$ in m	${}^{(W)}r_y$ in m	${}^{(W)}r_z$ in m	${}^{(W)}\Phi_z$ in $^\circ$	${}^{(W')}\Phi_y$ in $^\circ$	${}^{(W'')}\Phi_z$ in $^\circ$	Δt_s in s
${}^{(W)}\mathbf{x}_{\text{pose}_1}$	-0.5	0.5	0	0	0	-90	2
${}^{(W)}\mathbf{x}_{\text{pose}_2}$	0.5	0.8	0.7	0	0	0	1
${}^{(W)}\mathbf{x}_{\text{pose}_3}$	0.5	0.3	0.7	0	0	0	1
${}^{(W)}\mathbf{x}_{\text{pose}_4}$	0.5	0.2	0.1	0	0	-90	2

Due to the distribution of the task poses, three translational DOF are required. Additionally, the EE should rotate about an axis parallel to the z -axis, i. e. only one rotational DOF is needed. Thus, the required motion vector for the task is

$$\boldsymbol{\xi}_{\text{req}} = (\xi_{\text{req}_1}, \xi_{\text{req}_2}, \xi_{\text{req}_3}, 0, 0, \xi_{\text{req}_6})^T. \quad (5.1)$$

As result of the structural synthesis, 35 suitable architectures are obtained for this required motion vector. Their DH parameters are listed in appendix B. In order to simplify their designation, the architectures are labeled using numbers. Architectures 1 to 17 are z -oriented, architectures 18 to 26 are y -oriented and architectures 27 to 35 are x -oriented. They are illustrated in figures 5.1 to 5.3.

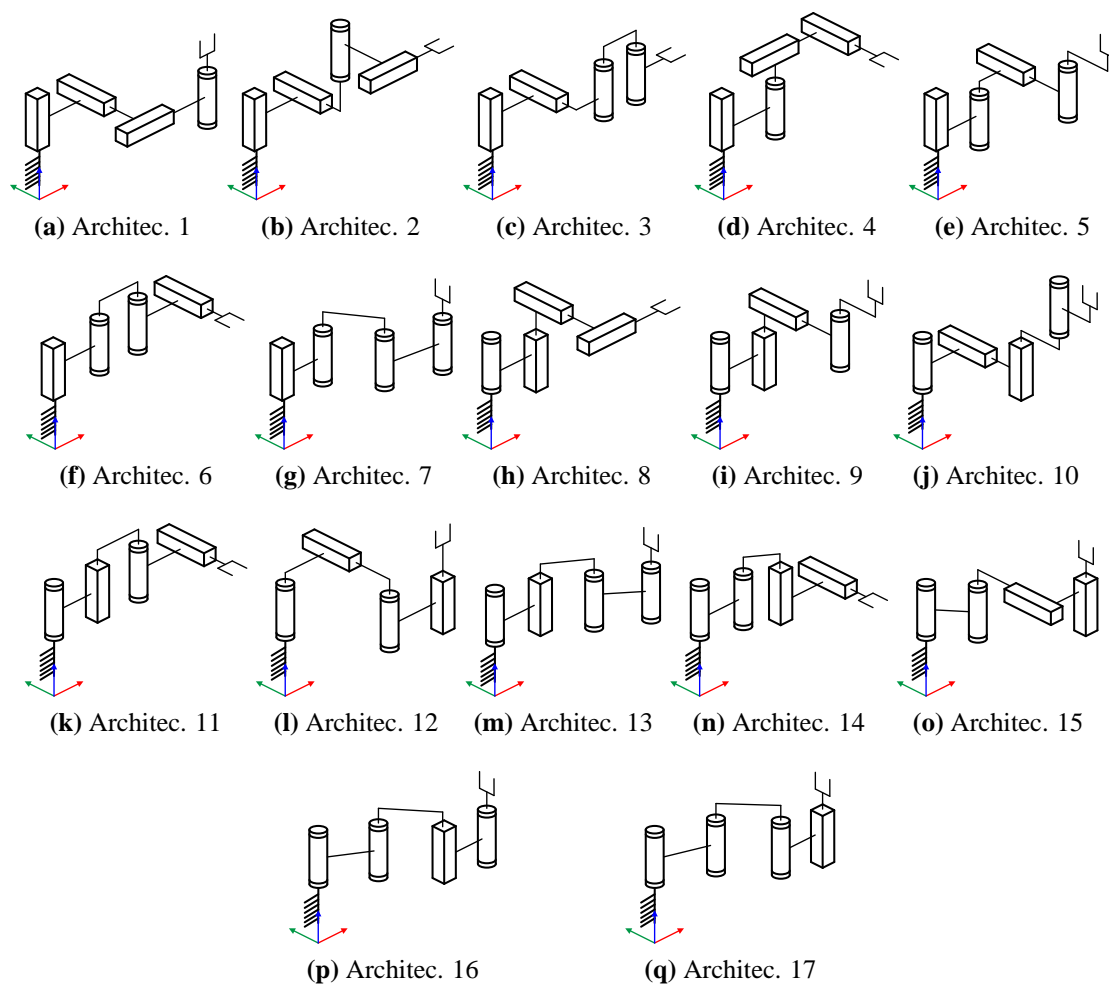


Figure 5.1: Sketches of the architectures 1 to 17 (z -oriented)

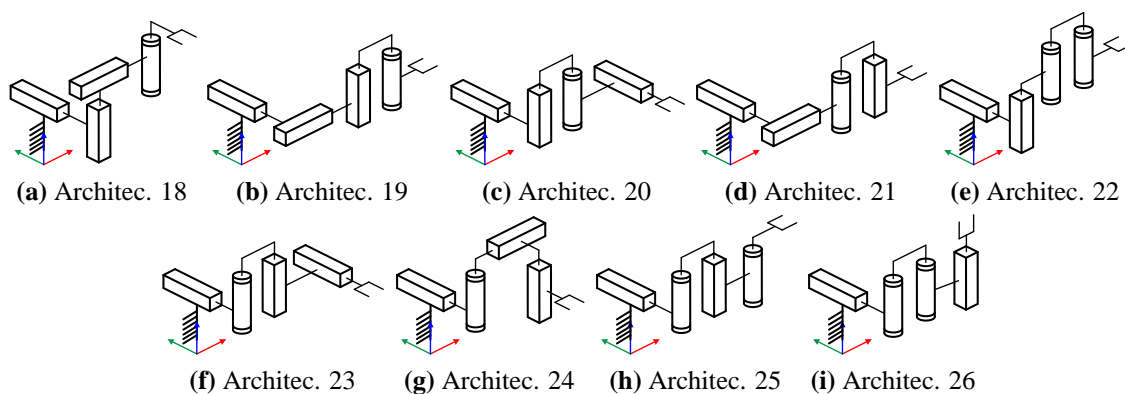


Figure 5.2: Sketches of the architectures 18 to 26 (y -oriented)

All suitable architectures can be associated with the structures reported in [CKPA10], where the structures are described through the arrangement of the joint axes. As it can be seen from appendix B, the modifiable geometric parameters are additionally used to define the architectures. This is a significant feature of the CRS.

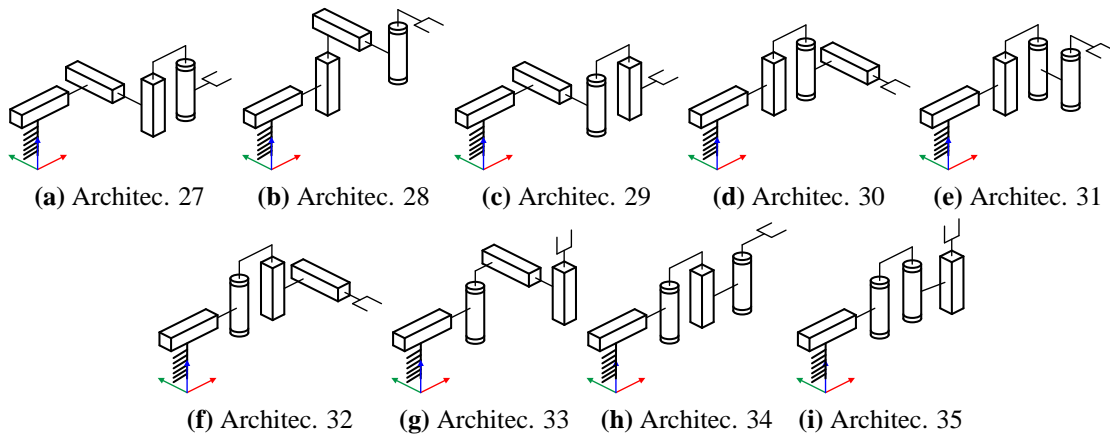


Figure 5.3: Sketches of the architectures 27 to 35 (x -oriented)

The suitable architectures are optimized in section 5.1.1 using a kinematic criterion and in section 5.1.2 with respect to dynamic performance. In both sections, the performance indices used as objective function as well as constraints are explained in detail.

5.1.1 Optimization with respect to a kinematic criterion

Kinematic performance indices evaluate the motion transmission between the joint and the EE, i. e. the ability of a manipulator to transform joint velocities into velocity at the EE [MSC⁺12a]. A common issue considered in the kinematic performance evaluation is the presence of singularities. The most frequently used indices to detect singular configurations are minimal singular value of the JACOBIAN, manipulability, and condition number of the JACOBIAN [Yos85, ALC92]. Nevertheless, the latter can not only detect if the manipulator is in a singular configuration, but also works as a metric of the proximity to the singularity [LA16].

The condition number κ of the homogeneous JACOBIAN $\bar{\mathbf{J}}$, defined in (2.26), is calculated as

$$\kappa(\bar{\mathbf{J}}) = \left\| \bar{\mathbf{J}} \right\|_2 \left\| \bar{\mathbf{J}}^{-1} \right\|_2 = \frac{\sigma_{\max}(\bar{\mathbf{J}})}{\sigma_{\min}(\bar{\mathbf{J}})}, \quad (5.2)$$

where σ_{\max} is the largest and σ_{\min} the smallest singular value of $\bar{\mathbf{J}}$. As $\bar{\mathbf{J}}$ depends on the joint coordinates \mathbf{q} , κ depends also on \mathbf{q} and, therefore, is a pose dependent index.

Instead of the condition number, the local conditioning index

$$\eta(\mathbf{q}) = \frac{1}{\kappa(\bar{\mathbf{J}}(\mathbf{q}))} \quad (5.3)$$

is more commonly used as performance index in robot synthesis [ALC92, MSC⁺12b, LA16]. To verify that the robot does not reach any task pose in a singular configuration, the local conditioning

index is calculated for the joints positions $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{N_{\text{task}}}$ corresponding to the poses of the task. The minimum of these values is then used to find the minimum conditioning index during the task.

$$\eta_{\min} = \min \left(\eta(\mathbf{q}_1), \eta(\mathbf{q}_2), \dots, \eta(\mathbf{q}_{N_{\text{task}}}) \right). \quad (5.4)$$

In the optimization process, the value of η_{\min} has to be maximized. However, the PSO algorithm attempts to minimize the objective function. Therefore, the objective function for this optimization is establish as

$$h(\mathbf{p}) = 1 - \eta_{\min}, \quad (5.5)$$

which can take values between 0 and 1. If $h(\mathbf{p}) = 1$, at least one pose is reached in a singular configuration. If $h(\mathbf{p}) = 0$ all poses correspond to isotropic configurations [RAGPP95]. The upper bound of the function allows the use of penalty values to manage the constraints violations as explain in section 4.2.

As basic requirement of the optimization, all task poses have to be reached. For a given pose, if no solution can be found for the inverse kinematics, it is considered that the pose is not reachable. Furthermore, collisions between the robot links are not allowed during the task execution. The collision evaluation is performed as explained in section 4.5.3. The penalty values assigned to the objective function in case of constraints violation (see section 4.4) are h_{p0} if any required point can not be reached and h_{p1} in case of collision.

The minimum values of the objective function h obtained after the optimization of each architecture are visualized in figure 5.4. The four best results are highlighted and their corresponding value h is presented. The architectures are labeled according to figures 5.1 to 5.3. Although all architectures are able to perform the task, for architecture 33 the algorithm could not find any combination of geometric parameters fulfilling the given constraints (no bar in figure 5.4). As it can be seen, the lowest objective function is achieved by the architecture 18. The DH parameters as well as an illustrative sketch of the manipulator with the best performance are presented in figure 5.5. The length parameters are reported in meters, the angle parameters are reported in radians.

As explained in section 4.6, an additional feature of the CRS is the possibility to study the variation of the objective function due to changes in the optimization parameters. For this example, the postprocessing was performed for the four architectures with the best performance namely 18, 1, 28, and 19. The geometric parameters of the best manipulators with these architectures are shown in figures 5.5 to 5.8.

The Monte Carlo simulation is performed for each architecture with 1,000 samples. The obtained frequency distribution of the objective function h is depicted in figures 5.9 to 5.12. The plots on the left side show the frequency distribution of the samples that fulfill the constraints. The red lines identify the lowest and highest values of h obtained from the Monte Carlo Simulation. The dotted blue line represents the minimum value of h obtained in the optimization. As it can be seen, the four architectures exhibit similar deviation.

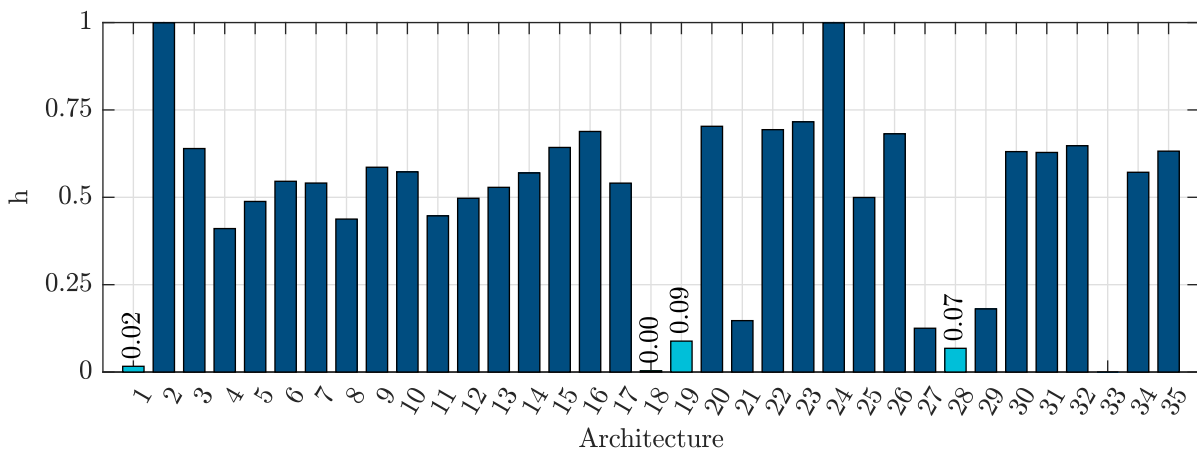


Figure 5.4: Optimization results for the suitable architectures. Architectures 1 to 17 are z -oriented, architectures 18 to 26 are y -oriented, architectures 27 to 35 are x -oriented

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
P	0.000	q_1	0.576	1.570
P	1.570	q_2	0.693	1.570
P	0.000	q_3	0.756	1.570
R	q_4	0.745	-0.001	0.663

$${}^{(w)}\mathbf{r}_0 = (-0.296, -0.310, 0.350)^T \text{ in m}$$

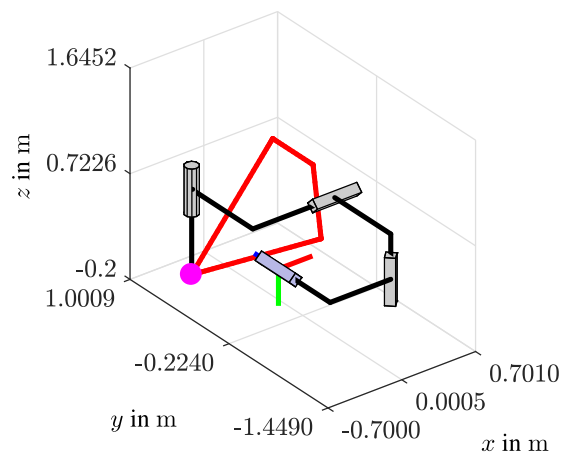


Figure 5.5: Manipulator with the minimum objective function of architecture 18 (PPPR)

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
P	1.570	q_1	0.770	1.570
P	1.570	q_2	0.631	1.570
P	0.000	q_3	0.770	1.570
R	q_4	0.743	-0.021	-3.120

$${}^{(w)}\mathbf{r}_0 = (0.036, -0.008, -0.250)^T \text{ in m}$$

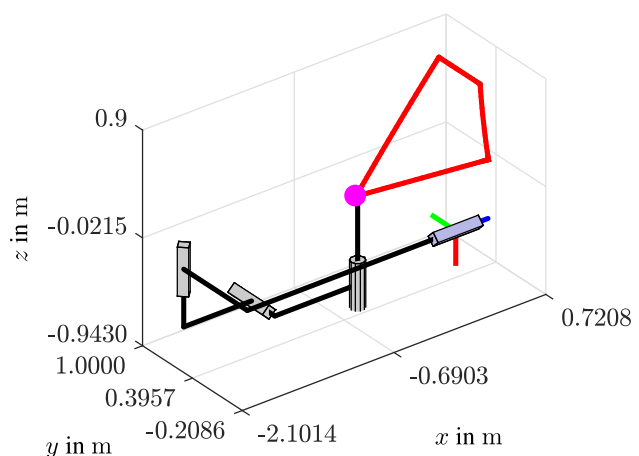


Figure 5.6: Manipulator with the minimum objective function of architecture 28 (PPPR)

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
P	2.499	q_1	0.765	1.570
P	1.570	q_2	0.765	1.570
P	1.570	q_3	0.759	1.570
R	q_4	0.655	0.000	-3.0944

$${}^{(w)}\mathbf{r}_0 = (0.193, -0.090, -0.312)^T \text{ in m}$$

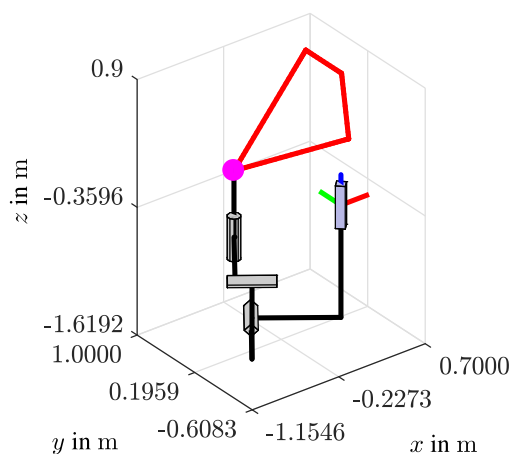


Figure 5.7: Manipulator with the minimum objective function of architecture 1 (PPPR)

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
P	1.570	q_1	0.619	1.570
P	1.570	q_2	0.699	1.570
P	0.149	q_3	0.774	0.000
R	q_4	0.482	-0.027	1.420

$${}^{(w)}\mathbf{r}_0 = (0.102, -0.272, 0.059)^T \text{ in m}$$

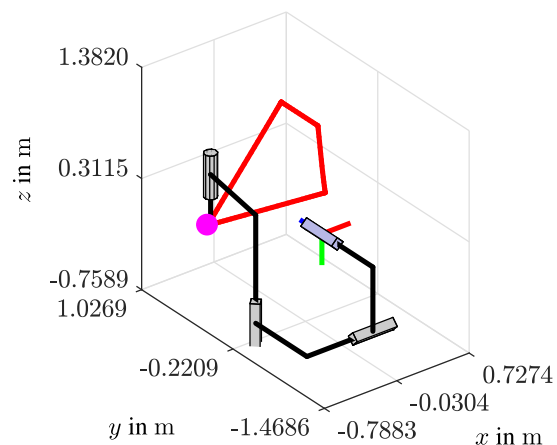
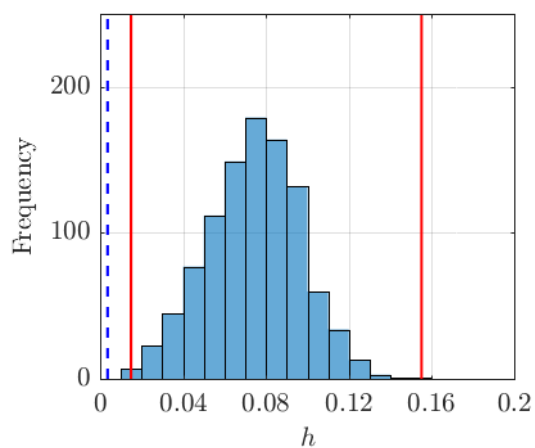
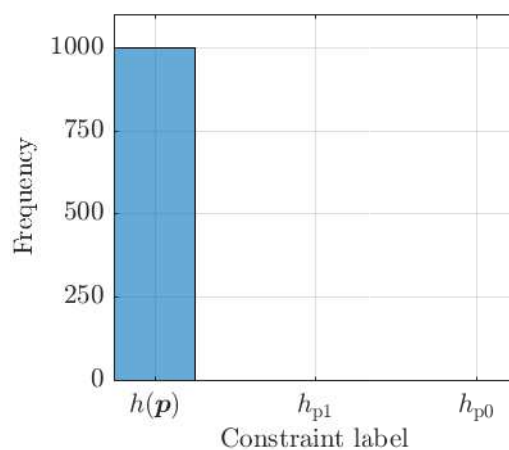


Figure 5.8: Manipulator with the minimum objective function of architecture 19 (PPPR)



(a) 1000 manipulators fulfilling the constraints



(b) All samples

Figure 5.9: Frequency distribution of h after the Monte Carlo simulation. Architecture 18 (PPPR)

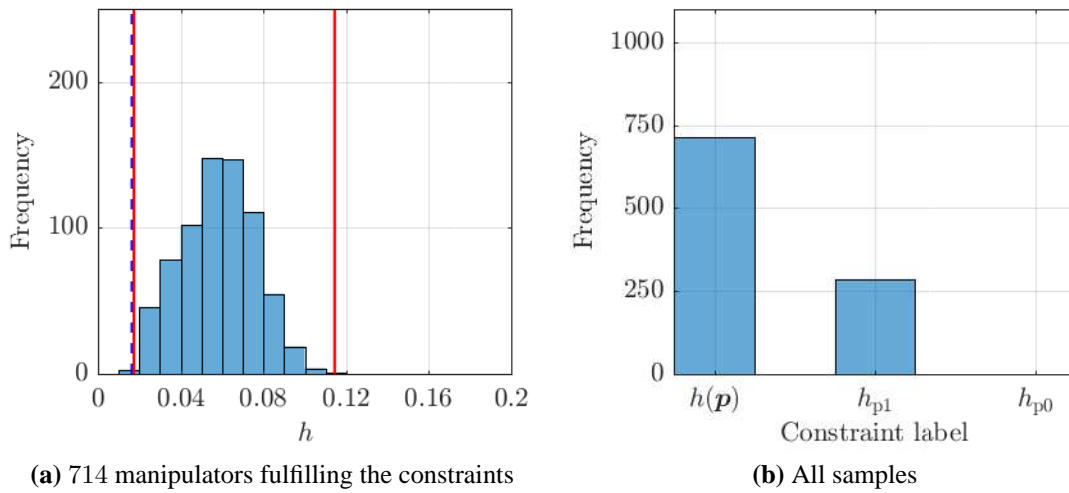


Figure 5.10: Frequency distribution of h after the Monte Carlo simulation. Architecture 1 (PPPR)

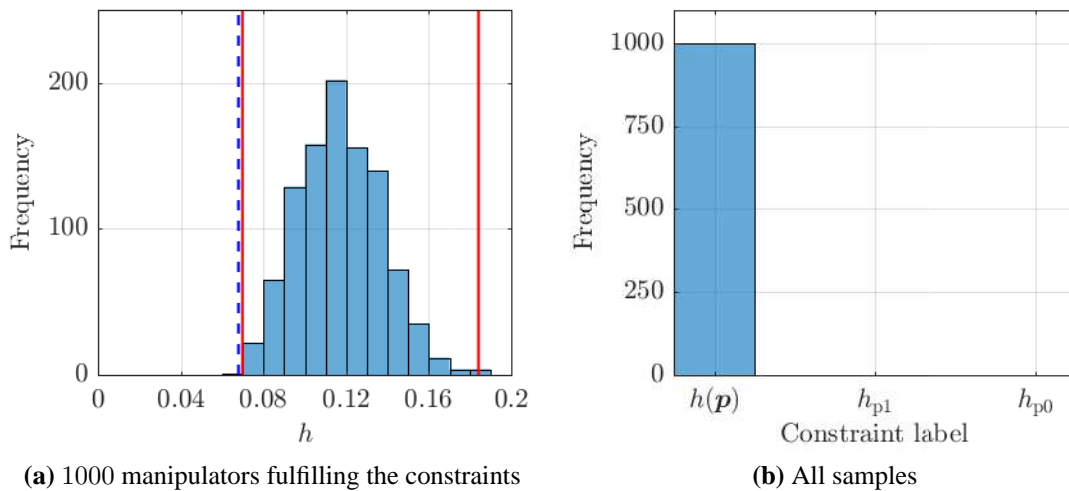


Figure 5.11: Frequency distribution of h after the Monte Carlo simulation. Architecture 28 (PPPR)

The plots on the right side show the distribution of all 1,000 samples. The label $h(\mathbf{p})$ represents samples fulfilling the constraints. The label h_{p1} corresponds to manipulators for which collision during the task execution was detected. Label h_{p0} identifies manipulators which can not reach all task poses. From figures 5.9(b) to 5.12(b) it can be inferred that architecture 1 present collision issues when the geometric parameters are modified from the optimal configuration. For the other three architectures, the reachability and the absence of collision are practically not affected by variations in the optimal manipulator.

5.1.2 Optimization with respect to a dynamic criterion

As an example of dynamic performance, the present section shows the application of the CRS in the minimization of the energy consumption for the same task listed in table 5.1. The standstill

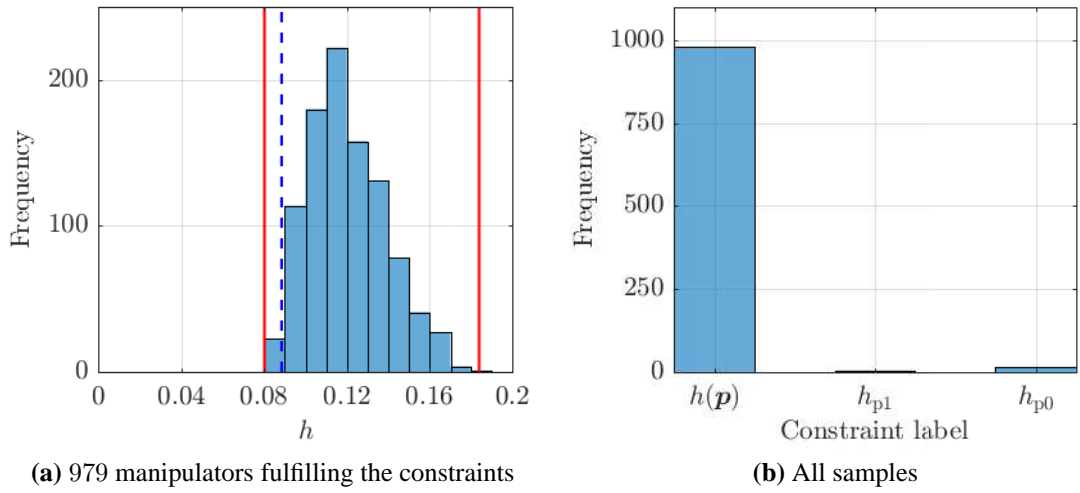


Figure 5.12: Frequency distribution of h after the Monte Carlo simulation. Architecture 19 (PPPR)

time in each pose as well as the energy consumption of the breaks and the electric supply are not considered. The specifications of the profile used for the parametrization of the links, as well as the characteristics of the actuators are listed in appendix C.

In order to calculate the objective function, the required torque/force of the motor i calculated in (2.45) can be rewritten as

$$\tau_{M_i}(t) = \frac{\tau_{L_i}(t)}{\rho_i} + {}_{(M_i)}I_{zz}^{(M_i)} \ddot{q}_i(t) \rho_i. \quad (5.6)$$

This expression is used to determine the mechanical power consumption $P_i(t)$ of the motor as

$$P_i(t) = \tau_{M_i}(t) \rho_i \dot{q}_i(t) \quad (5.7)$$

$$= \left(\tau_{L_i}(t) + {}_{(M_i)}I_{zz}^{(M_i)} \ddot{q}_i(t) \rho_i^2 \right) \dot{q}_i(t). \quad (5.8)$$

Besides mechanical power, electrical losses in the drive trains (motors, inverters, power supply, etc.) represent a significant power consumption [HIMO12]. Nevertheless, the topic is out of the goal of this work and the manipulators will be optimized regarding the mechanical power consumption. Since (specially in industrial robots) the drivers are usually connected over a common DC-bus, electrical power exchange can be considered [HIMO12]. The resulting total power $P_{\text{sum}}(t)$ is determined as

$$P_{\text{sum}}(t) = \sum_{i=1}^n P_i(t). \quad (5.9)$$

To calculate the energy consumption, only positive values of $P_{\text{sum}}(t)$ have to be considered [HKO13]. For this purpose, the Heaviside function Θ is used to set the negative values of $P_{\text{sum}}(t)$ to zero:

$$\Theta = \begin{cases} 0 & \text{if } P_{\text{sum}} \leq 0, \\ 1 & \text{if } P_{\text{sum}} > 0. \end{cases} \quad (5.10)$$

The energy E_{task_s} consumed during the motion from $\mathbf{x}_{\text{task}_s}$ to $\mathbf{x}_{\text{task}_{s+1}}$ is obtained using the integral over the time $t_{s+1} - t_s$

$$E_{\text{task}_s} = \int_{t_s}^{t_{s+1}} \Theta P_{\text{sum}}(t) dt. \quad (5.11)$$

The total energy for the task is calculated as

$$E_{\text{total}} = \sum_{s=1}^N E_{\text{task}_s}, \quad (5.12)$$

which is used as objective function in the optimization of the suitable architectures,

$$h(\mathbf{p}) = E_{\text{total}}. \quad (5.13)$$

As further requirements, the following aspects are considered:

- all poses ${}_{(W)}\mathbf{x}_{\text{pose}_s}$ have to be reached,
- the task poses ${}_{(W)}\mathbf{x}_{\text{pose}_s}$ are reached in a non-singular configuration.
- there are no collisions during the execution of the task (including the path between ${}_{(W)}\mathbf{x}_{\text{pose}_s}$),

The first requirement is fulfilled if the inverse kinematic solution exists for all task poses. When this condition is not fulfilled, the penalty value h_{p0} is assigned to the objective function. The second requisite is evaluated through a threshold value for (5.4). From the last example, $\eta_{\min} = 0.25$ (see (5.5) and figure 5.4) can be achieved for most of the suitable architectures. Therefore, for this example, the local conditioning index in each pose is conditioned to be larger than 0.25. If this condition is not fulfilled, h_{p1} is assigned to the objective function. The third constraint is evaluated using (4.68) and using the penalty value h_{p2} . Additionally, in order to consider the actuators capability as a constraint of the optimization, the function c_{act} is defined as

$$c_{\text{act}}(\mathbf{p}) = \begin{cases} -1 & \text{if actuators for all joints could be found from the predefined set,} \\ 1 & \text{if one or more actuators could not be found from the predefined set.} \end{cases} \quad (5.14)$$

If $c_{\text{act}}(\mathbf{p}) = 1$, the the penalty value h_{p3} is given to the objective function.

Figure 5.13 summarizes the results of the optimization for all suitable architectures. Although all architectures are able to perform the task, for some architectures (without bar), the optimization could not find geometric parameters that fulfill all constraints. In the plot, it is worth underlining the difference between the performance of the architectures, which confirms the significance of the kinematic structure in the manipulators design. The manipulator with the minimum energy consumption corresponds to the architecture 12 (see figure 5.1). Its geometric parameters are listed in figure 5.14.

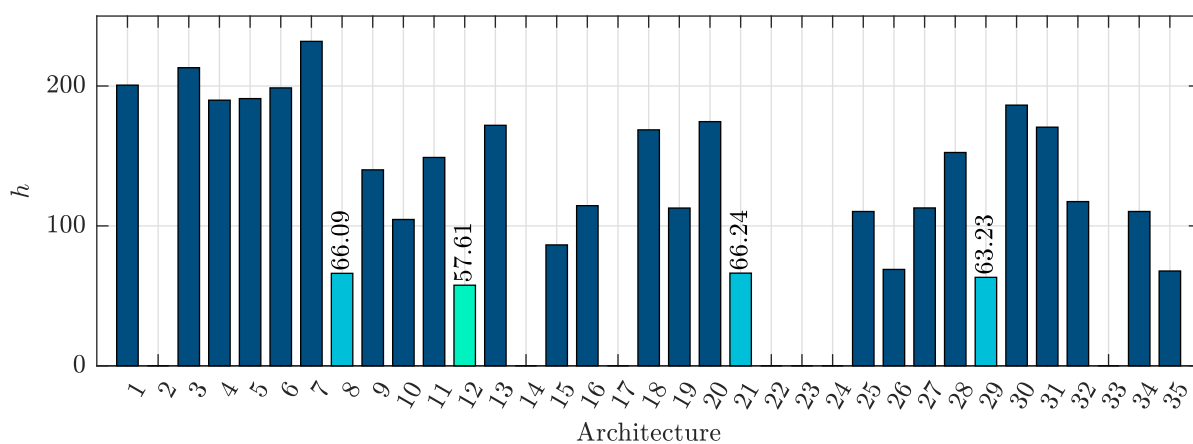


Figure 5.13: Optimization results for the suitable architectures. Architectures 1 to 17 are z -oriented, architectures 18 to 26 are y -oriented, architectures 27 to 35 are x -oriented

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
R	q_1	0.652	-0.116	1.570
P	0.000	q_2	0.004	1.570
R	q_3	0.101	0.104	0.000
P	-0.947	q_4	0.000	1.543

$${}^{(w)}\mathbf{r}_0 = (-0.155, -0.154, -0.054)^T \text{ in m}$$

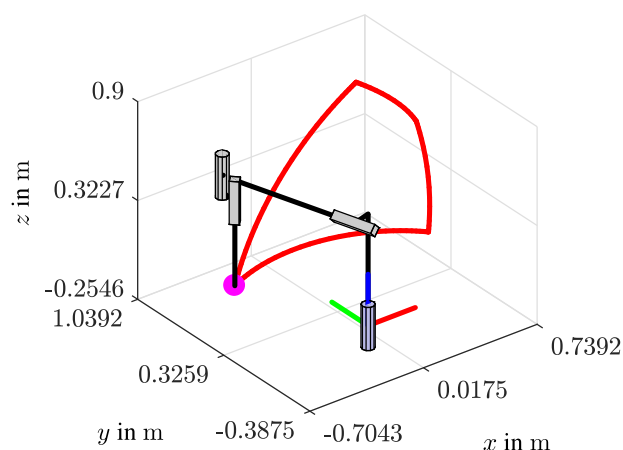


Figure 5.14: Manipulator with the minimum objective function of architecture 12 (RPRP)

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
P	0.000	q_1	0.000	1.570
P	1.570	q_2	0.240	1.570
R	q_3	0.000	0.207	0.000
P	-1.633	q_4	0.000	0.7398

$${}^{(w)}\mathbf{r}_0 = (-0.479, 0.342, 0.492)^T \text{ in m}$$

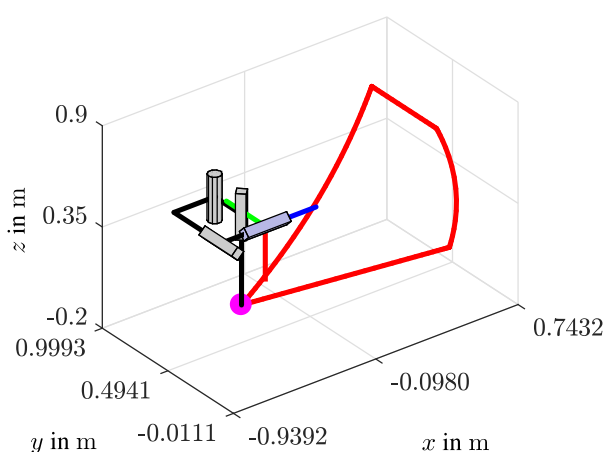


Figure 5.15: Manipulator with the minimum objective function of architecture 29 (PPRP)

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
R	q_1	0.008	0.073	-1.570
P	-1.352	q_2	0.264	1.570
P	1.570	q_3	0.105	1.570
P	0.663	q_3	-0.000	-2.439

$${}^{(w)}\mathbf{r}_0 = (0.133, 0.434, -0.309)^T \text{ in m}$$

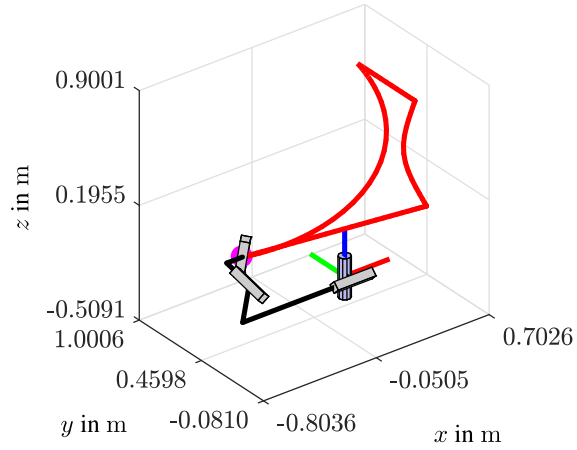


Figure 5.16: Manipulator with the minimum objective function of architecture 8 (RPPP)

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
P	1.570	q_1	0.000	1.570
P	1.570	q_2	0.100	1.570
R	q_3	0.093	0.100	0.000
P	-1.496	q_4	0.000	-0.030

$${}^{(w)}\mathbf{r}_0 = (-0.345, 0.170, 0.467)^T \text{ in m}$$

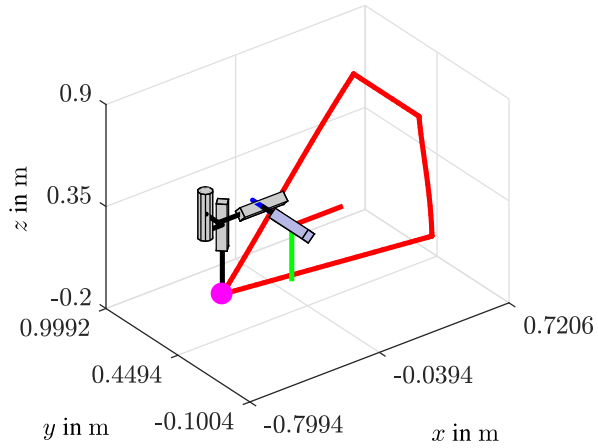


Figure 5.17: Manipulator with the minimum objective function of architecture 21 (PPRP)

The four architectures with the best performance in figure 5.13 are highlighted. For these manipulators the sensitivity analysis was performed as explained in section 4.6 with 1000 samples. The histograms with the variation of the objective function are shown in figures 5.18 to 5.21. As in the last section, the plots at the left side show the values of the objective function for the manipulators that fulfill all constraints. The plots at the right side show the number of manipulators fulfilling the constraints (labeled as E_{total}) as well as the manipulators that violate any constraint (labeled with the corresponding penalty value h_{p3} to h_{p0}).

Considering only manipulators fulfilling all constraints, the architectures 12, 29 and 21 (figures 5.18(a), 5.19(a) and 5.21(a)) exhibit a similar dispersion and, therefore, a similar sensitivity to changes in the geometric parameters. In the other hand, architecture 8 present higher sensitivity to geometric modifications.

From the sensitivity analysis for the architectures 12 and 21 (figures 5.18(b) and 5.21(b)) it can be seen that the most of the samples present collision when the optimal geometric parameters are modified. However, this condition can be avoided in a subsequent detailed design stage. For its

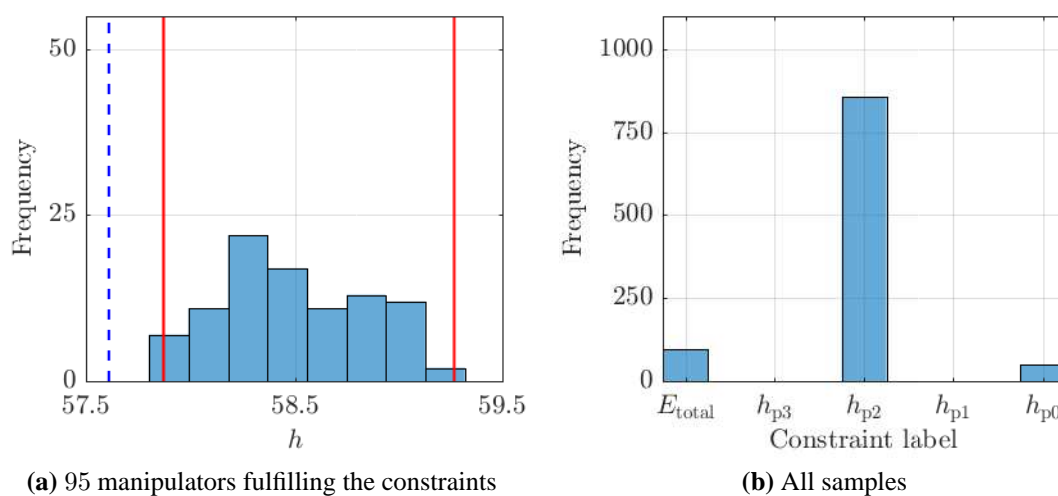


Figure 5.18: Frequency distribution of h after the Monte Carlo simulation. Architecture 12 (PPRP)

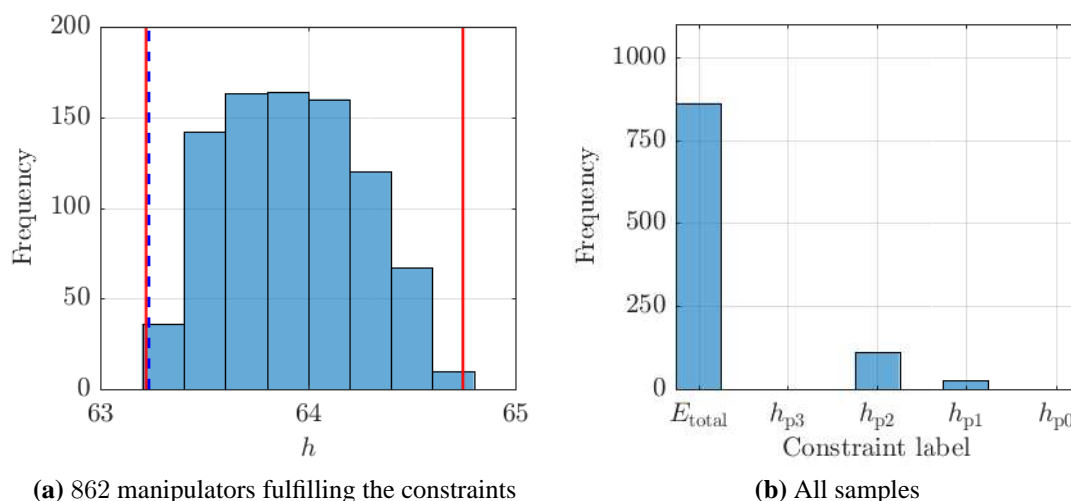


Figure 5.19: Frequency distribution of h after the Monte Carlo simulation. Architecture 29 (PPRP)

part, when the optimal manipulator of architecture 8 is modified, many of these modifications can lead to local conditioning index values under the establish threshold. It means, the manipulator is closer to a singularity as allowed. In the case of architecture 29, most of the samples fulfill the constraints. Only a small set present problems due to collisions and a few due to the local conditioning index.

5.2 Mechanisms with 6 DOF

In order to demonstrate the generality of the proposed approach, a more complex task (table 5.2) is considered in the following example, in which the minimization of the manipulator size will

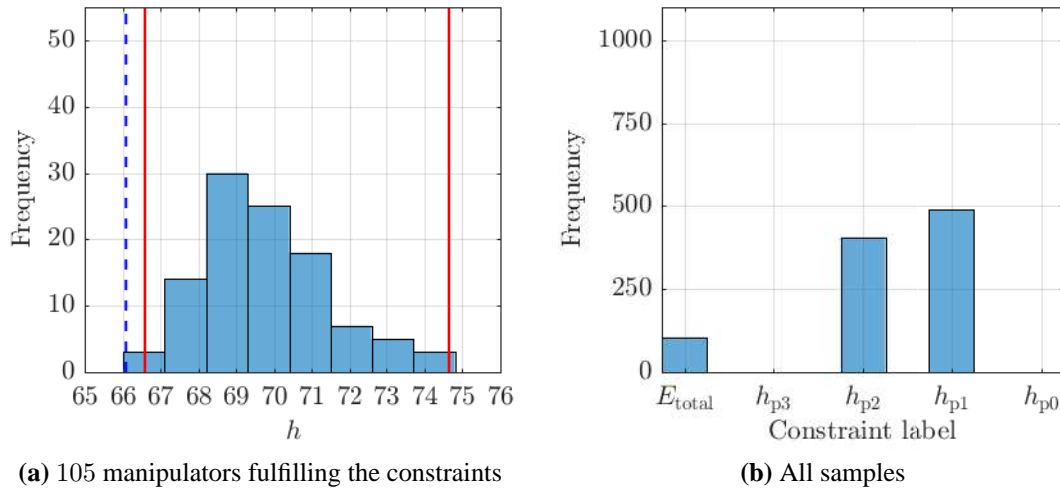


Figure 5.20: Frequency distribution of h after the Monte Carlo simulation. Architecture 8 (RPPP)

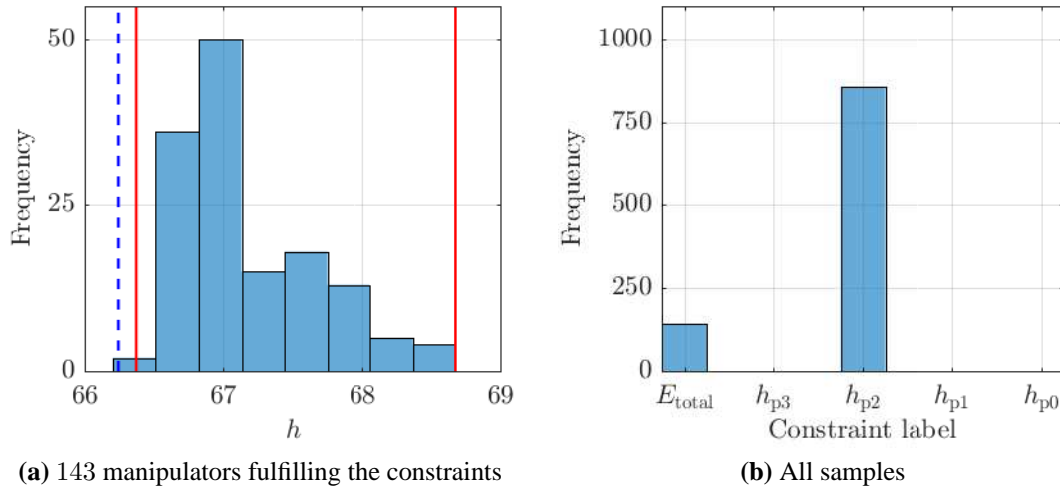


Figure 5.21: Frequency distribution of h after the Monte Carlo simulation. Architecture 21 (PPRP)

be used as main requirement. A minimum value for the local conditioning index as well as the absence of collision during the task execution are chosen as additional requirements.

The task coordinates for this example are listed in table 5.2. They correspond to the four vertices of a square. The EE must be positioned on each point with a different orientation. For this task, 6DOF manipulators will be used. Therefore, the required motion vector is in this case

$$\xi_{\text{req}} = (\xi_{\text{req}_1}, \xi_{\text{req}_2}, \xi_{\text{req}_3}, \xi_{\text{req}_4}, \xi_{\text{req}_5}, \xi_{\text{req}_6})^T. \quad (5.15)$$

A total of 978 architectures were found as result of the structural synthesis. From this group, 326 are z -oriented architectures, 326 y -oriented, and the other 326 x -oriented. The number of architectures N_{ar}^j found for each joint sequence is listed in table 5.3. Since at least three R joints are required to achieve three rotational DOF of the EE, there are no architectures with less than

Table 5.2: Coordinates of the desired task (minimization of the robot size)

Pose	${}^{(W)}r_x$ in m	${}^{(W)}r_y$ in m	${}^{(W)}r_z$ in m	${}^{(W)}\psi_z$ in $^\circ$	${}^{(W')}\psi_y$ in $^\circ$	${}^{(W'')}\psi_z$ in $^\circ$
${}^{(W)}\mathbf{x}_{\text{pose}_1}$	-0.2	1.4	0.8	75	-105	0
${}^{(W)}\mathbf{x}_{\text{pose}_2}$	0.2	1.4	0.8	105	-90	0
${}^{(W)}\mathbf{x}_{\text{pose}_3}$	0.2	1.4	1.2	105	-75	0
${}^{(W)}\mathbf{x}_{\text{pose}_4}$	-0.2	1.4	1.2	75	-75	0

three R joints. To facilitate the analysis of the results, the architectures are grouped according to their joint sequence. The joint sequence of a manipulator indicates the order of the joints in this manipulator. For instance, the joint sequence for a manipulator whose first two joints are prismatic and the last four are revolute joints is PPRRRR. The number of architectures found for each joint sequence is listed in table 5.3.

Table 5.3: Number of architectures found for each joint sequence. The categories marked with (*) are chosen for the optimization

Joint sequence	N_{ar}^j	Joint sequence	N_{ar}^j	Joint sequence	N_{ar}^j	Joint sequence	N_{ar}^j
PPPPPP	0	PPPPPR	0	PPPPRP	0	PPPPRR	0
PPPRPP	0	PPPRPR	0	PPRRRP	0	PPRRRR	3
PPRPPP	0	PPRPPR	0	PPRPRP	0	PPRPRR	3
PPRRPP	0	PPRRPR	6	PPRRRP	6	PPRRRR	9*
PRPPPP	0	PRPPPR	0	PRPPRP	0	PRPPRR	9
PRPRPP	0	PRPRPR	21	PRRRRP	18	PRRRRR	21
PRRPPP	0	PRRPPR	15	PRRPRP	15	PRRPRR	24
PRRRPP	9	PRRRPR	24	PRRRRP	18	PRRRRR	18*
RPPPPP	0	RPPPPR	0	RPPPRP	0	RPPPRR	15
RPPRPP	0	RPPRPR	24	RPPRRP	24	RPPRRR	24
RPRPPP	0	RPRPPR	24	RPRPRP	30	RPRPRR	39
RPRRPP	24	RPRRPR	36	RPRRRP	36	RPRRRR	33
RRPPPP	0	RRPPPR	15	RRPPRP	15	RRPPRR	33
RRPRPP	15	RRPRPR	42	RRRRRP	42	RRRRRR	42
RRRPPP	3	RRRPPR	30	RRRPRP	36	RRRPRR	45
RRRRPP	27	RRRRPR	39	RRRRRP	36	RRRRRR	30*

Without loss of generality, the marked (*) architectures are exemplarily chosen to be considered within the CRS. The DH parameters of these architectures are shown in appendix A. As in the last examples, they are labeled with numbers. The chosen groups are the joint sequences RRRRRR (6R), PRRRRR (1P5R), and PPRRRR (2P4R). The first group corresponds to the most common architectures used in 6 DOF industrial robots. They have been previously studied in [Gog02], where they are classified in six families according to their singular configurations. Nevertheless,

additional comparison between these architectures could not be found in the literature. The group 1P5R can be understood as a 5R manipulator mounted on a rail. In the case of the 2P4R architectures, a 4R robot is mounted on two rails. Similar concepts are actually applied to increase the robot workspace [ABB17, Clo17b, Clo17a], however, in most cases for 6R robots.

Each architecture is optimized as explained in section 4.4. The threshold of the condition number and the link radius for the collision evaluation are respectively

$$\begin{aligned} \eta^* &= 0.1 \text{ and} \\ r_i &= r_j = 0.05 \text{ m for all links.} \end{aligned} \quad (5.16)$$

The minimum values of the objective function obtained after the optimization are visualized in figures 5.22 to 5.24. The correspondence between the labels and the direction of the first joint

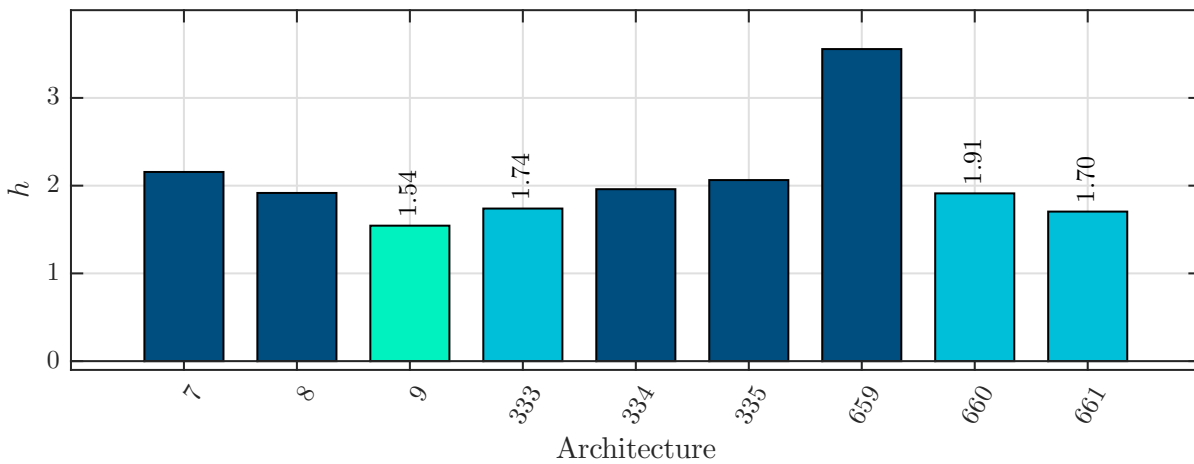


Figure 5.22: Optimization results for 2P4R architectures. Architectures 7 to 9 are z -oriented, architectures 333 to 335 are y -oriented, architectures 659 to 661 are x -oriented

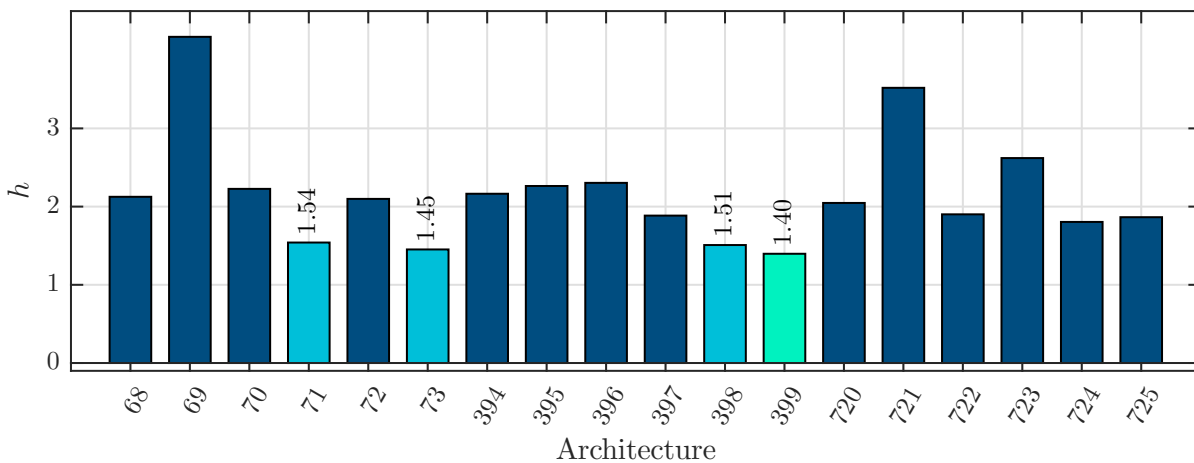


Figure 5.23: Optimization results for 1P5R architectures. Architectures 68 to 73 are z -oriented, 394 to 399 are y -oriented, 720 to 725 are z -oriented

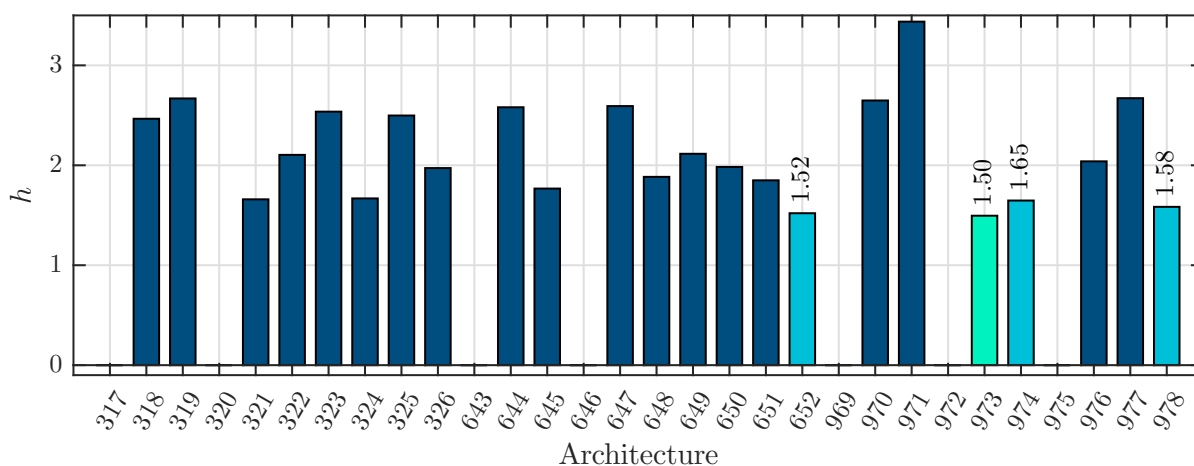


Figure 5.24: Optimization results for 6R architectures. Architectures 317 to 326 are z -oriented, 643 to 652 are y -oriented, 969 to 978 are x -oriented

axis for the architectures (see section 4.3) is presented in table 5.4.

Table 5.4: Orientation of the selected architectures

First joint axis orientation	figure 5.22	figure 5.23	figure 5.24
z -oriented architectures	7 - 9	68 - 73	317 - 326
y -oriented architectures	333 - 335	394 - 399	643 - 652
x -oriented architectures	659 - 661	720 - 725	969 - 978

Although all architectures are able to perform the task, for the architectures 317, 320, 643, 646, 696, 972, 975 (without bar in figure 5.24), the optimization algorithm could not find any combination of geometric parameters fulfilling the given constraints. The best results of each group are highlighted and their corresponding value h is presented in the figures. From the selected groups, the architecture with the best performance is the 399. The geometric parameters of the optimal manipulator is shown in figure 5.25.

The postprocessing was performed through a Monte Carlo simulation with 5,000 samples. As in the last two examples, the four manipulators with the best performance were considered, namely the architectures 399, 73, 973, and 398. Their sketches and DH parameters are presented in figures 5.25 to 5.28. The frequency distribution obtained from the Monte Carlo simulation is depicted in figures 5.29 to 5.32.

The plots on the left side show the frequency distribution of the objective function for manipulators fulfilling the optimization constraints. Considering the architecture 398 (figure 5.32(a)), although lower values were found through the Monte Carlo simulation, these are still higher than the minimum values of architecture 399.

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
P	0.179	q_1	0	3.009
R	q_2	-0.002	0.108	2.653
R	q_3	0.100	0	0.878
R	q_4	1.217	0	1.570
R	q_5	0	0.118	1.570
R	q_6	0	0.020	1.408

$${}^{(W)}\mathbf{r}_0 = (0.022, 0.199, 0.199)^T \text{ in m}$$

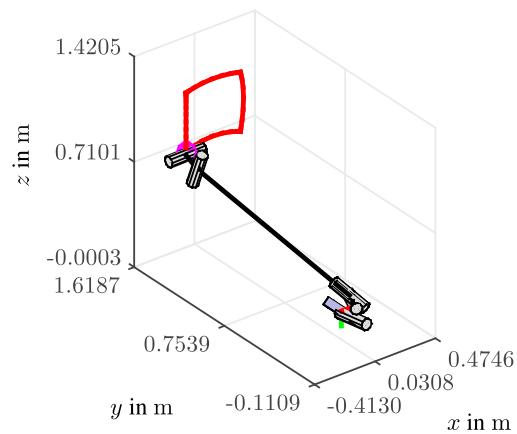


Figure 5.25: Manipulator with the minimum size index of architecture 399 (1P5R)

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
P	1.5147	q_1	0.015	1.125
R	q_2	0	0.227	-1.618
R	q_3	0.1	0	1.102
R	q_4	0.915	0	1.570
R	q_5	0	0.296	1.570
R	q_6	0	0	-2.978

$${}^{(W)}\mathbf{r}_0 = (0.051, 0.2, 0.2)^T \text{ in m}$$

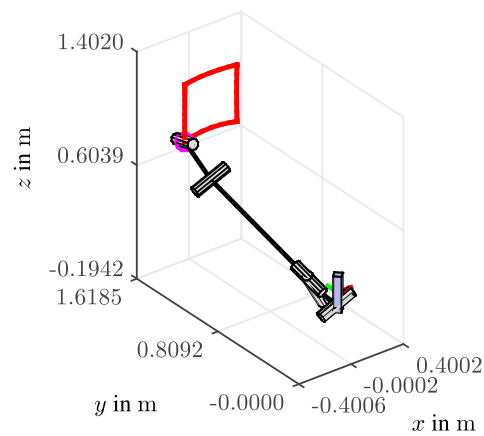


Figure 5.26: Manipulator with the minimum size index of architecture 73 (1P5R)

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
R	q_1	0	0	0.929
R	q_2	-0.003	0.133	1.570
R	q_3	-0.005	0.766	0
R	q_4	0	0.765	0
R	q_5	0	0.119	1.570
R	q_6	0	0.002	0.585

$${}^{(W)}\mathbf{r}_0 = (0.003, 0.198, 0.199)^T \text{ in m}$$

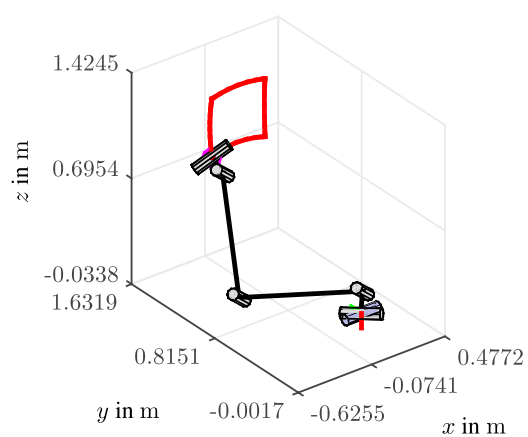


Figure 5.27: Manipulator with the minimum size index of architecture 973 (6R)

R/P	θ_i in rad	d_i in m	a_i in m	α_i in rad
P	-0.065	q_1	0.003	2.459
R	q_2	-0.002	0.984	-2.629
R	q_3	0.102	-0.005	1.570
R	q_4	0.108	0	1.570
R	q_5	-0.003	0.485	0
R	q_6	0.001	0.015	2.6275

$${}^{(w)}\mathbf{r}_0 = (0, 0.2, 0.2)^T \text{ in m}$$

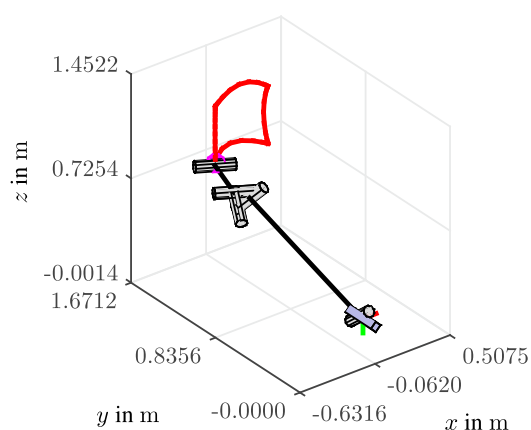
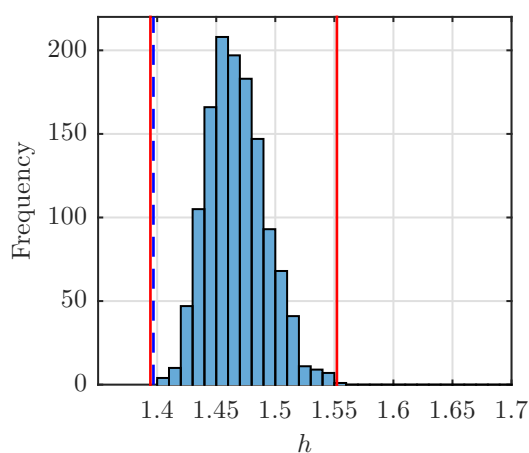
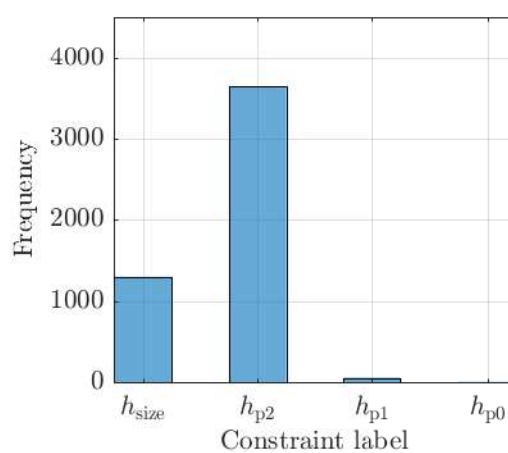


Figure 5.28: Manipulator with the minimum size index of architecture 398 (1P5R)

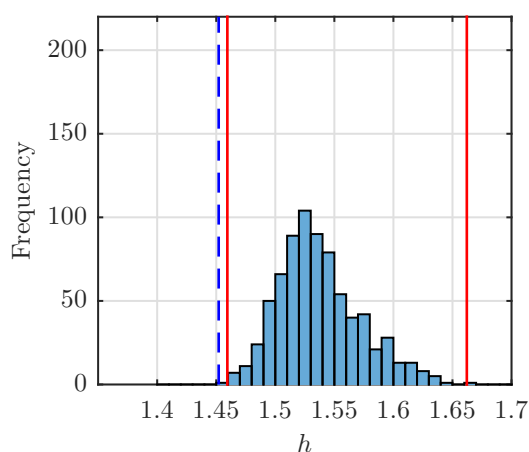


(a) 268 manipulators fulfilling the constraints

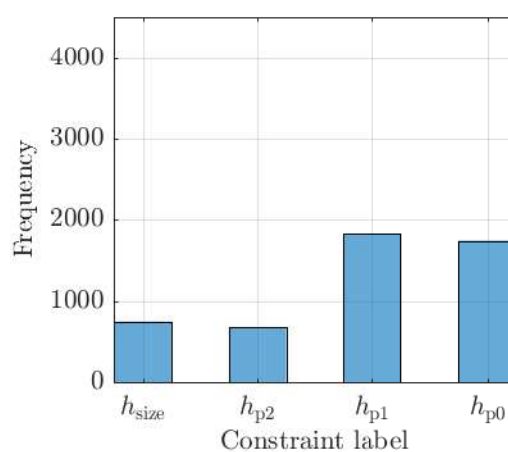


(b) All samples

Figure 5.29: Frequency distribution of h after the Monte Carlo simulation. Architecture 399 (2P4R)



(a) 168 manipulators fulfilling the constraints



(b) All samples

Figure 5.30: Frequency distribution of h after the Monte Carlo simulation. Architecture 73 (1P5R)

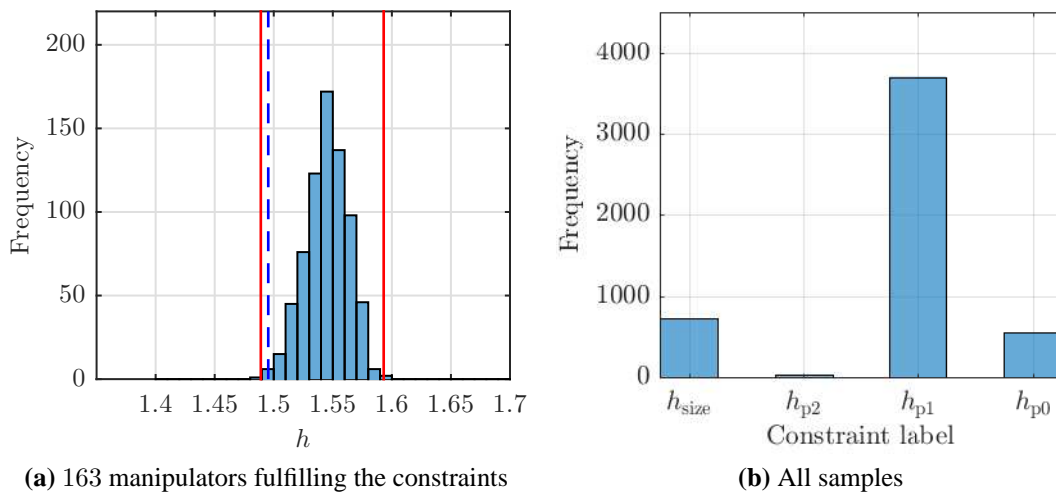


Figure 5.31: Frequency distribution of h after the Monte Carlo simulation. Architecture 973 (6R)

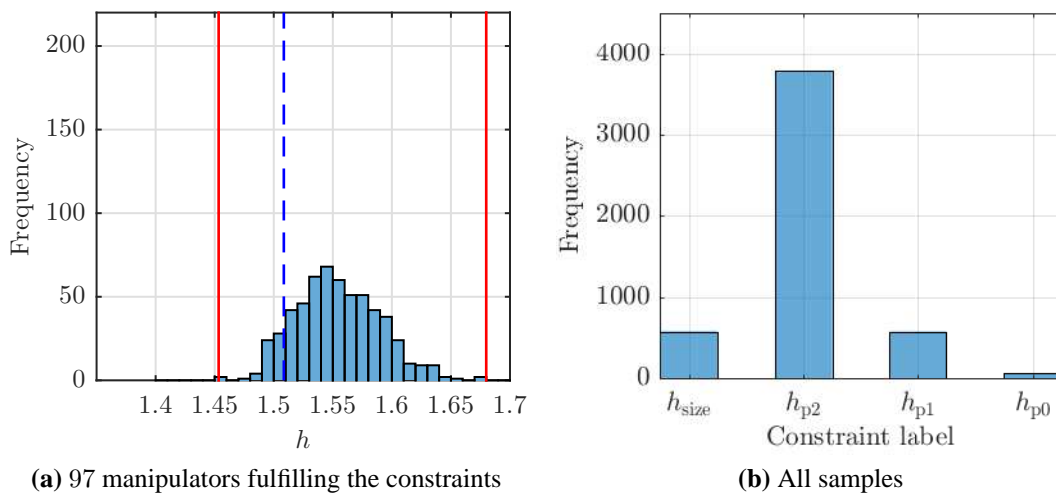


Figure 5.32: Frequency distribution of h after the Monte Carlo simulation. Architecture 398 (1P5R)

The plots on the right side show the distribution of all 5,000 samples. The labels correspond to the penalty values used for the constraints violation:

- h_{size} – manipulators fulfilling all constraints,
- h_{p2} – collisions are detected in the path execution,
- h_{p1} – local conditioning index is lower than allowed,
- h_{p0} – one or more poses of the task can not be reached,

From figure 5.29(b) and figure 5.32(b) it can be inferred that the principal inconvenience regarding the variation of the geometric parameters in architectures 399 and 398 is the collision between the links. However, this issue mainly depends on the final construction and could be solved with an adequate design of the links and joints. Architecture 973 (figure 5.31(b)), for its part, mainly exhibits dexterity problems when its geometric parameters are modified. Since the conditioning

index is a kinematic characteristic, it can not be modified in a detailed design stage. In the case of architecture 73 (figure 5.30(b)), there is not a clear predominant effect due to variations in its geometry.

From the four presented architectures, the 399 presents the lowest deviation of h and the largest amount of manipulators that fulfill the constraints despite the variation of its geometry. Furthermore, it does not present inconvenience due dexterity or reachability.

6 Summary and future work

The increasing development of task-specific robots with new architectures has been favored by the offer of modular construction systems, modular software solutions, and developments in rapid prototyping technologies. In the design of new manipulators two aspects have to be taken into account, namely the architecture and the dimensions of the robot. Existing methods are able to address each of these issues separately. However, they cannot be connected in a single synthesis method. The structural synthesis addresses the generation of architectures fulfilling a set of DOF. Meanwhile, the dimensional synthesis is dedicated to find the optimal dimensions for a single pre-defined structure. Since the performance of several structures is not considered in the dimensional synthesis, the obtained manipulator is not necessarily optimal for the required task.

The goal of the present work is the development of an efficient combined (structural-dimensional) robot synthesis for serial manipulators, which is able to consider all suitable architectures in the synthesis of task-specific manipulators. Furthermore, the method allows for comparing the performance of several architectures with respect to the task requirements. Executable code is automatically generated for the kinematic and dynamic modeling of each architecture. Hence, the combined robot synthesis is able to optimize task-specific manipulators with respect to kinematic and dynamic performance.

6.1 Summary

The first step of the CRS is the generation of all architectures that exhibit the required DOF. The robot architectures and their geometry are described using the DH notation. Therefore, identical architectures represented by different DH parameters (isomorphisms) are identified and grouped. The isomorphisms detection is based on the dependency between the direction of the EE-DOF and the direction in which each DH parameter is measured. Thanks to this procedure, the number of architectures decreases drastically. This allows the combination of the structural and geometric synthesis due to the reduction of the search space in the optimization. Furthermore, the geometric parameters that can be modified without affecting the EE-DOF are determined for each architecture. These are used as optimization parameters in the subsequent dimensional synthesis.

Besides the geometric parameters of each architecture, the position and orientation of the robot base is considered. The position is evaluated through three position coordinates. Robots with different base orientation are considered, however, as different architectures due to the modification of the EE-DOF orientation. Considering the modified position and orientation of the manipulator

base, the inverse kinematic is calculated using a numeric approach for a general serial manipulator. Unlike current methods, the orientation error is calculated using the axis-angle representation. This provides a meaningful expression and avoids overdetermined equation systems. The numeric algorithm is based on the Newton-Raphson method. Hence, in order to overcome singularities of the JACOBIAN matrix during the algorithm execution, both the transpose and the inverse are employed according to the distance to a singularity.

The forward kinematics and the dynamics are calculated, in turn, in symbolic form individually for each architecture. The determination of the closed-form equations allows for the automatic generation of MATLAB® and C code. Thanks to the automatic modeling of each architecture, several performance indices can be efficiently evaluated and each architecture can be optimized.

Due to the characteristics of the optimization problem, the PSO algorithm was chosen to solve it. The optimization is performed with respect to a main requirement and additional requirements are considered as constraints. The constrained optimization problem is handled through static penalty values for each type of constraint.

The capability of the CRS is demonstrated through three validation examples. In the former, 4 DOF manipulators are synthesized with respect to kinematic performance. The local conditioning index in the task's poses is used as main requirement. This index can be considered a metric of the distance to a singularity. As a further requirement, the robot must execute the task without collisions. In the second example, the synthesis is performed for 4 DOF manipulators considering dynamic performance. The energy consumption during the task execution is chosen as objective function. Threshold values for the local conditioning index as well as the absence of collisions are used as constraints. Compared with the evaluation of kinematic performance, the calculation of the objective function is in this case significantly more complex due to the consideration of the robot dynamics. The considerable difference between the performance of the optimized architectures verifies the relevance of the manipulator structure in the robot design. In a third example, 6 DOF manipulators are synthesized and optimized with respect to their size. As further requirements (constraints), the manipulator cannot be in a singular configuration when it reaches each required pose. Additionally, the robot must execute the task without collisions.

For the three examples, the sensitivity of the architectures with the best performance is studied through a Monte Carlo simulation. This determines the deviation of the objective function when the optimization parameters are varied with an uniform distribution. Moreover, it is possible to investigate which constraint becomes critical when the geometric parameters are changed. The sensitivity analysis allows for identify the most robust solutions for the required task as well as determining whether the critical constraint can (or cannot) be handled in subsequent design stages. Collisions between the links, for instance, can be handled in a subsequent step, since they greatly depend on the links final geometry. Meanwhile, singular configurations depend on the joint axes arrangement which cannot be modified without changing the kinematic structure. The comparison

between several architectures with respect to their performance as well to their sensitivity is one of the main features of the CRS and represents an innovation against existing works.

Summarizing, the present work presents a new methodology for the synthesis of task-specific robot manipulators. Thanks to a new structural synthesis and the automatic individual modeling of the manipulators, all suitable architectures can be considered in the optimization process. The strategies and methods developed and successfully validated in this work allow for the performance evaluation as well as the sensitivity analysis of several architectures simultaneously with low effort for the designer.

6.2 Future work

The combined robot synthesis provides a powerful tool for the conceptual design of serial robots. Its results can be employed in the automatic generation of CAD-based initial prototypes as can be observed in [Tho16], where the automatic CAD generation of different kinematic structures was explored (see figure 6.1). This kind of tools can significantly accelerate the conceptual design stage. Alternatively, the methods can be used to design tasks optimal robots from existing

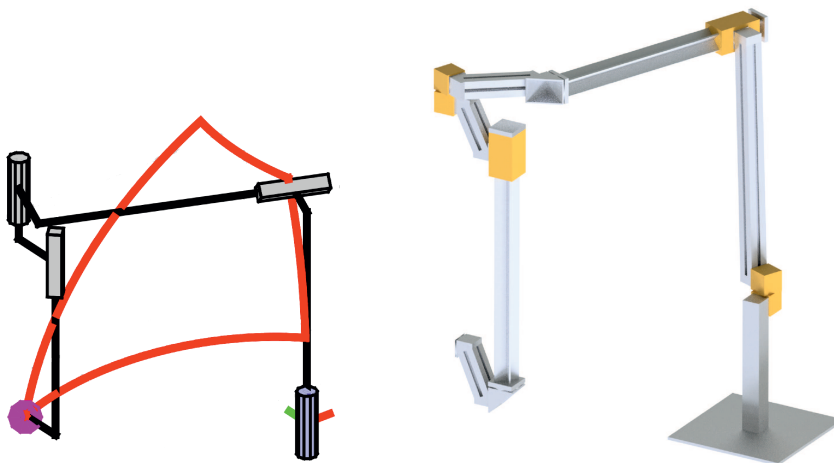


Figure 6.1: Automatic CAD generation of kinematic structures [Tho16]

construction modules or to choose the most suitable robot for a task from a set of predefined robots. The automatic generated code can be used in a quick setup of the control and an easy implementation of several control strategies.

The extension of the CRS to parallel mechanisms is a meaningful research subject. The synthesis of parallel mechanisms is known as a challenging topic due not only to the modeling but also to the large number of possible architectures. However, parallel structures offer several advantages against serial mechanisms as e. g. higher stiffness and velocity.

A Architectures with 6 DOF

A.1 2P4R Architectures

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1
P	θ_2	q_2	a_2	α_2	P	θ_2	q_2	a_2	α_2	P	θ_2	q_2	a_2	α_2
R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	α_3
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	$\pi/2$
R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 7 (b) Architecture 8 (c) Architecture 9

Figure A.1: Architectures 7 to 9

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1
P	θ_2	q_2	a_2	α_2	P	θ_2	q_2	a_2	α_2	P	θ_2	q_2	a_2	α_2
R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	α_3
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	$\pi/2$
R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 333 (b) Architecture 334 (c) Architecture 335

Figure A.2: Architectures 333 to 335

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1
P	θ_2	q_2	a_2	α_2	P	θ_2	q_2	a_2	α_2	P	θ_2	q_2	a_2	α_2
R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	α_3
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	$\pi/2$
R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 659 (b) Architecture 660 (c) Architecture 661

Figure A.3: Architectures 659 to 661

A.2 1P5R Architectures

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1
R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	$\pi/2$
R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	0
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$
R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	0
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 68 (b) Architecture 69

Figure A.4: Architectures 68 to 69

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1
R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	α_2
R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$
R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	0
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 70 (b) Architecture 71 (c) Architecture 72

Figure A.5: Architectures 70 to 72

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1
R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	$\pi/2$
R	q_3	d_3	a_3	α_3	R	q_3	d_3	a_3	0
R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	0
R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	$\pi/2$
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 73 (b) Architecture 394

Figure A.6: Architectures 73 and 394

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1
R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	α_2
R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$
R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	0
R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 395 (b) Architecture 396 (c) Architecture 397

Figure A.7: Architectures 395 to 397

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1
R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	α_2
R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	α_3	R	q_3	d_3	a_3	$\pi/2$
R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	$\pi/2$
R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	$\pi/2$
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	P	θ_6	q_6	a_6	α_6

(a) Architecture 398 (b) Architecture 399 (c) Architecture 719

Figure A.8: Architectures 398 to 719

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1
R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	$\pi/2$
R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	$\pi/2$
R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	0
R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	0
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 721 (b) Architecture 722

Figure A.9: Architectures 721 to 722

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1	P	θ_1	q_1	a_1	α_1
R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	α_2
R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	α_3
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	$\pi/2$
R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 723 (b) Architecture 724 (c) Architecture 725

Figure A.10: Architectures 723 to 725

A.3 6R Architectures

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	$\pi/2$	R	q_1	d_1	a_1	$\pi/2$	R	q_1	d_1	a_1	$\pi/2$
R	q_2	d_2	a_2	0	R	q_2	d_2	a_2	0	R	q_2	d_2	a_2	0
R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	$\pi/2$
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	0
R	q_5	d_5	0	$\pi/2$	R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	0
R	q_6	0	0	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 317 (b) Architecture 318 (c) Architecture 319

Figure A.11: Architectures 317 to 319

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	$\pi/2$	R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	α_1
R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	$\pi/2$
R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	0
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$
R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	0
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 320 (b) Architecture 321 (c) Architecture 322

Figure A.12: Architectures 320 to 322

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	α_1
R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	α_2
R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$
R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	0
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 323 (b) Architecture 324 (c) Architecture 325

Figure A.13: Architectures 323 to 325

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	$\pi/2$	R	q_1	d_1	a_1	$\pi/2$
R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	0	R	q_2	d_2	a_2	0
R	q_3	d_3	a_3	α_3	R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	0
R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$
R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	0	$\pi/2$	R	q_5	d_5	a_5	0
R	q_6	d_6	a_6	α_6	R	q_6	0	0	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 326 (b) Architecture 643 (c) Architecture 644

Figure A.14: Architectures 326 to 644

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	$\pi/2$	R	q_1	d_1	a_1	$\pi/2$	R	q_1	d_1	a_1	α_1
R	q_2	d_2	a_2	0	R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	$\pi/2$
R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	0
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	0
R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 645 (b) Architecture 646 (c) Architecture 647

Figure A.15: Architectures 645 to 647

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	α_1
R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	α_2
R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$
R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	0
R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 648 (b) Architecture 649 (c) Architecture 650

Figure A.16: Architectures 648 to 650

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	$\pi/2$
R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	0
R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	α_3	R	q_3	d_3	a_3	0
R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	0
R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	0	$\pi/2$
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	0	0	α_6

(a) Architecture 651 (b) Architecture 652 (c) Architecture 969

Figure A.17: Architectures 651 to 969

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	$\pi/2$	R	q_1	d_1	a_1	$\pi/2$	R	q_1	d_1	a_1	$\pi/2$
R	q_2	d_2	a_2	0	R	q_2	d_2	a_2	0	R	q_2	d_2	a_2	$\pi/2$
R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	0
R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	0
R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	0
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 970 (b) Architecture 971 (c) Architecture 972

Figure A.18: Architectures 970 to 972

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	α_1
R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	$\pi/2$
R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	$\pi/2$
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	0
R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	0
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 973 (b) Architecture 974 (c) Architecture 975

Figure A.19: Architectures 973 to 975

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	α_1
R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	α_2	R	q_2	d_2	a_2	α_2
R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	α_3
R	q_4	d_4	a_4	0	R	q_4	d_4	a_4	$\pi/2$	R	q_4	d_4	a_4	$\pi/2$
R	q_5	d_5	a_5	$\pi/2$	R	q_5	d_5	a_5	0	R	q_5	d_5	a_5	$\pi/2$
R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6	R	q_6	d_6	a_6	α_6

(a) Architecture 976 (b) Architecture 977 (c) Architecture 978

Figure A.20: Architectures 976 to 978

B Architectures with 4DOF

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	$\pi/2$	P	θ_1	q_1	a_1	$\pi/2$	P	θ_1	q_1	a_1	$\pi/2$
P	$\pi/2$	q_2	a_2	$\pi/2$	P	0	q_2	a_2	$\pi/2$	P	0	q_2	a_2	$\pi/2$
P	$\pi/2$	q_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	0
R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4	R	q_4	d_4	a_4	α_4

(a) Architecture 1 (b) Architecture 2 (c) Architecture 3

Figure B.1: Architectures 1 to 3

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	0	P	θ_1	q_1	a_1	0	P	θ_1	q_1	a_1	0
R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	0
P	$\pi/2$	q_3	a_3	$\pi/2$	P	0	q_3	a_3	$\pi/2$	R	q_3	d_3	a_3	$\pi/2$
P	θ_4	q_4	a_4	α_4	R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4

(a) Architecture 4 (b) Architecture 5 (c) Architecture 6

Figure B.2: Architectures 4 to 6

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	θ_1	q_1	a_1	0	R	q_1	d_1	a_1	α_1	R	q_1	d_1	a_1	0
R	q_2	d_2	a_2	0	P	θ_2	q_2	a_2	$\pi/2$	P	θ_2	q_2	a_2	$\pi/2$
R	q_3	d_3	a_3	0	P	$\pi/2$	q_3	a_3	$\pi/2$	P	0	q_3	a_3	$\pi/2$
R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4	R	q_4	d_4	a_4	α_4

(a) Architecture 7 (b) Architecture 8 (c) Architecture 9

Figure B.3: Architectures 7 to 9

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	$\pi/2$	R	q_1	d_1	a_1	0	R	q_1	d_1	a_1	$\pi/2$
P	0	q_2	a_2	$\pi/2$	P	θ_2	q_2	a_2	0	P	0	q_2	a_2	$\pi/2$
P	θ_3	q_3	a_3	0	R	q_3	d_3	a_3	$\pi/2$	R	q_3	d_3	a_3	0
R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4	P	θ_4	q_4	a_4	α_4

(a) Architecture 10 (b) Architecture 11 (c) Architecture 12

Figure B.4: Architectures 10 to 12

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	0	R	q_1	d_1	a_1	0	R	q_1	d_1	a_1	0
P	θ_2	q_2	a_2	0	R	q_2	d_2	a_2	0	R	q_2	d_2	a_2	$\pi/2$
R	q_3	d_3	a_3	0	P	θ_3	q_3	a_3	$\pi/2$	P	0	q_3	a_3	$\pi/2$
R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4	P	θ_4	q_4	a_4	α_4

(a) Architecture 13 (b) Architecture 14 (c) Architecture 15

Figure B.5: Architectures 13 to 15

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
R	q_1	d_1	a_1	0	R	q_1	d_1	a_1	0
R	q_2	d_2	a_2	0	R	q_2	d_2	a_2	0
P	θ_3	q_3	a_3	0	R	q_3	d_3	a_3	0
R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4

(a) Architecture 16 (b) Architecture 17

Figure B.6: Architectures 15 to 17

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	0	q_1	a_1	$\pi/2$	P	$\pi/2$	q_1	a_1	$\pi/2$	P	0	q_1	a_1	$\pi/2$
P	θ_2	q_2	a_2	$\pi/2$	P	$\pi/2$	q_2	a_2	$\pi/2$	P	θ_2	q_2	a_2	0
P	0	q_3	a_3	$\pi/2$	P	θ_3	q_3	a_3	0	R	q_3	d_3	a_3	$\pi/2$
R	q_4	d_4	a_4	α_4	R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4

(a) Architecture 18 (b) Architecture 19 (c) Architecture 20

Figure B.7: Architectures 18 to 20

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	$\pi/2$	q_1	a_1	$\pi/2$	P	0	q_1	a_1	$\pi/2$	P	0	q_1	a_1	$\pi/2$
P	$\pi/2$	q_2	a_2	$\pi/2$	P	θ_2	q_2	a_2	0	R	q_2	d_2	a_2	0
R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	0	P	θ_3	q_3	a_3	$\pi/2$
P	θ_4	q_4	a_4	α_4	R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4

(a) Architecture 21 (b) Architecture 22 (c) Architecture 23

Figure B.8: Architectures 21 to 23

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	0	q_1	a_1	$\pi/2$	P	0	q_1	a_1	$\pi/2$	P	0	q_1	a_1	$\pi/2$
R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	0	R	q_2	d_2	a_2	0
P	0	q_3	a_3	$\pi/2$	P	θ_3	q_3	a_3	0	R	q_3	d_3	a_3	0
P	θ_4	q_4	a_4	α_4	R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4

(a) Architecture 24 (b) Architecture 25 (c) Architecture 26

Figure B.9: Architectures 24 to 26

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	0	q_1	a_1	$\pi/2$	P	$\pi/2$	q_1	a_1	$\pi/2$	P	0	q_1	a_1	$\pi/2$
P	θ_2	q_2	a_2	$\pi/2$	P	$\pi/2$	q_2	a_2	$\pi/2$	P	θ_2	q_2	a_2	0
P	0	q_3	a_3	$\pi/2$	P	θ_3	q_3	a_3	0	R	q_3	d_3	a_3	$\pi/2$
R	q_4	d_4	a_4	α_4	R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4

(a) Architecture 27 (b) Architecture 28 (c) Architecture 29

Figure B.10: Architectures 27 to 29

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	$\pi/2$	q_1	a_1	$\pi/2$	P	0	q_1	a_1	$\pi/2$	P	0	q_1	a_1	$\pi/2$
P	$\pi/2$	q_2	a_2	$\pi/2$	P	θ_2	q_2	a_2	0	R	q_2	d_2	a_2	0
R	q_3	d_3	a_3	0	R	q_3	d_3	a_3	0	P	θ_3	q_3	a_3	$\pi/2$
P	θ_4	q_4	a_4	α_4	R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4

(a) Architecture 30 (b) Architecture 31 (c) Architecture 32

Figure B.11: Architectures 30 to 32

R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i	R/P	θ_i	d_i	a_i	α_i
P	0	q_1	a_1	$\pi/2$	P	0	q_1	a_1	$\pi/2$	P	0	q_1	a_1	$\pi/2$
R	q_2	d_2	a_2	$\pi/2$	R	q_2	d_2	a_2	0	R	q_2	d_2	a_2	0
P	0	q_3	a_3	$\pi/2$	P	θ_3	q_3	a_3	0	R	q_3	d_3	a_3	0
P	θ_4	q_4	a_4	α_4	R	q_4	d_4	a_4	α_4	P	θ_4	q_4	a_4	α_4

(a) Architecture 33 (b) Architecture 34 (c) Architecture 35

Figure B.12: Architectures 33 to 35

C Specifications of the elements used in the dynamic optimization

Table C.1: Rotational modules [SchunkPR1]

Reference	GPA01-2S 06IN41	GPA01-2S 09HN41
$\tau_{L,peak}^*$ in N	70	170
$\tau_{M,peak}^*$ in N	3.5	11
$\tau_{L,nom}^*$ in N	40	126
$\tau_{M,nom}^*$ in N	1.5	3.8
$\omega_{G,peak}^*$ in rad/s	14.24	10.47
$\omega_{M,peak}^*$ in rad/s	419	419
ρ_i	28	35
m_i in kg	7	15
${}_{(M_i)}I_{zz}^{(M_i)}$ in kg m ²	0.000042	0.00023
I_{zz} in kg m ²	0.028233	0.0564

Table C.2: Linear modules [SchunkLDN1]

Reference	PLS 110 / MCS06I41	PLS 110 / MCS09H41
$\tau_{L,peak}^*$ in N	1099.56	3455.76
$\tau_{M,peak}^*$ in N	3.5	11
$\tau_{L,nom}^*$ in N	471.24	1193.80
$\tau_{M,nom}^*$ in N	1.5	3.8
$\omega_{G,peak}^*$ in rad/s	1.33	1.33
$\omega_{M,peak}^*$ in rad/s	419	419
ρ_i	314.16	314.16
m_i in kg	3	7
${}_{(M_i)}I_{zz}^{(M_i)}$ in kg m ²	0.000042	0.00023
I_{zz} in kg m ²	0.023	0.04

Table C.3: Standard aluminum profile [SMT]

Characteristic	Value
Area moment of inertia about x -axis in m^4	134.06×10^{-8}
Area moment of inertia about y -axis in m^4	134.06×10^{-8}
Linear density in kg/m	5.33

Bibliography

Individual topics of this thesis were published during the work in the institute of mechatronic systems. In total six international conference contributions [RKO13, RKO14, RKO15b, RKO15a, RKO16, RKO17] were published as first author. Although this is own content, the references to these publications are neither omitted nor separately marked in the text. The same applies for supervised student works [Plu15, Plu16, Tho16].

- [ABB17] *ABB Wind Turbine Manufacturing*: <http://www.abb.com/cawp/seitp202/470db65362e52de1c12578000060a9c8.aspx>. Accessed in October 2017.
- [AC00] ANGELES, J., CHABLAT, D.: On Isotropic Sets of Points in the Plane. Application to the Design of Robot Architectures. In: LENARČIČ, J. (editor), STANIŠIĆ, M. M. (editor): *Advances in Robot Kinematics*. Springer Netherlands, 2000.
- [ade17] *Adept Technology*: <https://www.adept.de>. Accessed in October 2017.
- [AHSA09] ALTUZARRA, O. , HERNANDEZ, A. , SALGADO, O. , ANGELES, J: Multiobjective Optimum Design of a Symmetric Parallel Schönflies-Motion Generator. *Journal of Mechanical Design*, 131(3):031002–031002–11, 2009.
- [AK06] AYDIN, Y., KUCUK, S.: Quaternion Based Inverse Kinematics for Industrial Robot Manipulators with Euler Wrist. In: *Proceedings of the IEEE International Conference on Mechatronics*, pages 581–586, 2006.
- [ALC92] ANGELES, J., LÓPEZ-CAJÚN, C.: Kinematic Isotropy and the Conditioning Index of Serial Robotic Manipulators. *The International Journal of Robotics Research*, 11(6):560–571, 1992.
- [Ang85] ANGELES, J.: On the Numerical Solution of the Inverse Kinematic Problem. *The International Journal of Robotics Research*, 4(2):21–37, 1985.
- [Ang04] ANGELES, J.: *Fundamentals of Robotic Mechanical Systems : Theory , Methods , and Algorithms*. Springer International Publishing, 2004.
- [Ang06] ANGELES, J.: Is there a Characteristic Length of a Rigid-Body Displacement. *Mechanism and Machine Theory*, 41:884–896, 2006.
- [APBH13] ANGERER, A. , PFURNER, M. , BRANDSTÖTTER, M. , HOFBAUR, M.: Modulare serielle Roboter. *e & i Elektrotechnik und Informationstechnik*, 130(2):48–53, 2013.

- [BAH15] BRANDSTÖTTER, M. , ANGERER, A. , HOFBAUR, M.: The Curved Manipulator (Cuma-Type Arm): Realization of a Serial Manipulator with General Structure in Modular Design. In: *Proceedings of the 14th IFToMM World Congress*, 2015.
- [Bax17] *Rethink Robotics*: <http://www.rethinkrobotics.com/de/baxter/>. Accessed in October 2017.
- [Ber02] BERGH, F. van d.: *An Analysis of Particle Swarm Optimizers*, Department of Computer Science, University of Pretoria, Pretoria, South Africa, Ph. D. Thesis, 2002.
- [BH15] BRANDSTÖTTER, M., HOFBAUR, M.: Placing the Kinematic Behavior of a General Serial 6R Manipulator by Varying its Structural Parameters. In: *Proceedings of the Austrian Robotics Workshop*, 2015.
- [BK05] BUSS, S., KIM, J.-S.: Selectively Damped Least Squares for Inverse Kinematics. *Journal of Graphics, GPU, and Game Tools*, 10(3):37–49, 2005.
- [BM03] BALLANTYNE, G., MOLL, F.: The Da Vinci Telerobotic Surgical System: The Virtual Operative Field and Telepresence Surgery. *Surg lin N Am*, (83):1293–1304, 2003.
- [BM08] BIAGIOTTI, L., MELCHIORRI, C.: *Trajectory Planning for Automatic Machines and Robots*. Springer Berlin Heidelberg, 2008.
- [Bon15] BONGARDT, B.: *Analytic Approaches for Design and Operation of Haptic Human-Machine Interfaces*, Universität Bremen, Bremen, Germany, Ph. D. Thesis, 2015.
- [Bor16] BORCHARD, J.-H.: *Entwicklung und Aufbau von Sstemen für die roboterassistierte Single-Port-Laparoskopie*, Leibniz Universität Hannover, Hanover, Germany, Ph. D. Thesis, 2016.
- [Bra16] BRANDSTÖTTER, M.: *Adaptable Serial Manipulators in Modular Design*, Institute of Automation and Control Engineering, University for Health Sciences, Medical Informatics and Technology, Hall in Tirol, Austria, Ph. D. Thesis, 2016.
- [Bus09] BUSS, S.: *Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods*. Available on: <https://www.math.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf>. 2009.
- [Car17] *Care-O-bot 4*: <http://www.care-o-bot-4.de/>. Accessed in October 2017.
- [CB17] CAPUTANO, M., BELLICOSO, D.: *DAMAROB: Dario and Marcos Robotics Symbolic Toolbox for Matlab*. Available in <http://www.damarob.altervista.org/>. Accessed in October 2017.
- [CBCP13] CARBONARI, L. , BATTISTELLI, M. , CALLEGARI, M. , PALPACELLI, M.-C.: Dynamic Modelling of a 3-CPU Parallel Robot Via Screw Theory. *Mechanical Sciences*, 4:185–197, 2013.

- [CBH08] CAMPOS, A. , BUDDE, C. , HESSELBACH, J.: A Type Synthesis Method for Hybrid Robot Structures. *Mechanism and Machine Theory*, 43(8):984–995, 2008.
- [CBLW16] CORVES, B. , BRINKER, J. , LORENZ, M. , WAHLE, M.: Design Methodology for Translational Parallel Manipulators Exhibiting Actuation Redundancy. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 230(3):425–436, 2016.
- [CCGC07] CASTEJON, C. , CARBONARE, G. , GARCIA, J. , CECCARELLI, M.: A Multiobjective Optimization for Designing Service Robots. In: *Proceedings of The 12th IFToMM World Congress*, 2007.
- [Che98] CHEDMAIL, P.: Optimization of Multi-DOF Mechanisms. In: ANGELES, J. (editor), ZAKHARIEV, E. (editor): *Computational Methods in Mechanical Systems: Mechanism Analysis, Synthesis, and Optimization*. Springer Berlin Heidelberg, 1998.
- [CKPA10] CARO, S. , KHAN, W. , PASINI, D. , ANGELES, J.: The Rule-Based Conceptual Design of the Architecture of Serial Schoenflies-Motion Generators. *Mechanism and Machine Theory*, 45(2):251–260, 2010.
- [Clo17a] *Cloos Horizontal Stroke*: <http://www.cloos.de/de-de/produkte/qirox/roboterpositionierer/horizontalhub-zur-montage-an-c-stander-hochbahn/qr-rp-hl-5kn-gl-20kn/>. Accessed in October 2017.
- [Clo17b] *Cloos Vertical Stroke*: <http://www.cloos.de/de-de/produkte/qirox/roboterpositionierer/vertikalhub-zur-montage-an-hochbahn/qr-rp-vl-5kn-c/>. Accessed in October 2017.
- [COC07] CARBONE, G. , OTTAVIANO, E. , CECCARELLI, M.: An Optimum Design Procedure for Both Serial and Parallel Manipulators. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 221(7):829–843, 2007.
- [COC08] CARBONE, G. , OTTAVIANO, E. , CECCARELLI, M.: Optimality Criteria for the Design of Manipulators. In: *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics*, pages 768–773, 2008.
- [Cor11] CORKE, P.: *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer-Verlag Berlin Heidelberg, 2011.
- [Cor17] CORKE, P.: *Robotics Toolbox*: www.petercorke.com/Robotics_Toolbox.html. Accessed in October 2017.
- [CR96] CHEDMAIL, P., RAMSTEIN, E.: Robot Mechanism Synthesis and Genetic Algorithms. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3466–3471, 1996.

- [Cra05] CRAIG, J.: *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall, 2005.
- [CTS04] COELLO, C. , TOSCANO, G. , SALAZAR, M.: Handling Multiple Objectives with Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, 2004.
- [Cyb17] *Cyberknife*: <https://www.cyber-knife.net/de/behandlung/technologie.html>. Accessed in October 2017.
- [Dai12] DAI, J.: Finite Displacement Screw Operators With Embedded Chasles Motion. *ASME. Journal of Mechanisms and Robotics*, 4(4):041002–1–041002–9, 2012.
- [Dai15] DAI, J.: Euler-Rodrigues Formula Variations, Quaternion Conjugation and Intrinsic Connections. *Mechanism and Machine Theory*, 92(Supplement C):144–152, 2015.
- [DGPC02] DI GREGORIO, R., PARENTI-CASTELLI, V.: Dynamic Performance Indices for 3-DOF Parallel Manipulators. In: LENARČIČ, J. (editor), THOMAS, F. (editor): *Advances in Robot Kinematics: Theory and Applications*. Springer Netherlands, 2002.
- [DK07] DOMBRE, E., KHALIL, W.: *Robot Manipulators: Modeling, Performance Analysis and Control*. John Wiley & Sons, 2007.
- [DLNK12] DEAN-LEON, E. , NAIR, S. , KNOLL, A.: User Friendly Matlab-Toolbox for Symbolic Robot Dynamic Modeling Used for Control Design. In: *Proceedings of the International Conference on Robotics and Biomimetics ROBIO*, 2012.
- [DLR17] *Deutsches Zentrum für Luft- und Raumfahrt, Robotik und Mechatronik Zentrum*: <http://www.dlr.de/rmc/rm/desktopdefault.aspx/tabid-11370/>. Accessed in October 2017.
- [EB09] EVERS, G., BEN, M.: Regrouping Particle Swarm Optimization: A New Global Optimization Algorithm with Improved Performance Consistency Across Benchmarks. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 3901–3908, 2009.
- [EKG12] EBBESEN, S. , KIWITZ, P. , GUZZELLA, L.: A Generic Particle Swarm Optimization Matlab Function. In: *Proceedings of the American Control Conference ACC*, pages 1519–1524, 2012.
- [ES98] EBERHART, R., SHI, Y.: Comparison Between Genetic Algorithms and Particle Swarm Optimization. In: PORTO, V. W. (editor) , SARAVANAN, N. (editor) , WAA-GEN, D. (editor) , EIBEN, A. E. (editor): *Proceedings of the 7th International Conference on Evolutionary Programming*. Springer Berlin Heidelberg, 1998.

- [Eve09] EVERS, G.: *An Automatic Regrouping Mechanism to Deal with Stagnation in Particle Swarm Optimization*, University of Texas-Pan American, Edinburg, USA, Master Thesis, 2009.
- [Fes17] *Festo Molecubes*: https://www.festo.com/rep/en_corp/assets/pdf/Molecubes_en.pdf. Accessed in October 2017.
- [Fis10] FISCHER, M.: *Anwendungsspezifische Kinematikentwicklung unter Einsatz modularer Prototypen*, Technische Universität München, Munich, Germany, Ph. D. Thesis, 2010.
- [Fra17] *Franka Emika*: <https://www.franka.de/>. Accessed in October 2017.
- [GB02] GALLANT, M., BOUDREAU, R.: The Synthesis of Planar Parallel Manipulators with Prismatic Joints for an Optimal, Singularity-Free Workspace. *Journal of Robotic Systems*, 19(1):13–24, 2002.
- [GBF85] GOLDENBERG, A. , BENHABIB, B. , FENTON, R.: A Complete Generalized Solution to the Inverse Kinematics of Robots. *IEEE Journal on Robotics and Automation*, 1(1):14–20, 1985.
- [GC00] GEN, M., CHENG, R.: *Genetic Algorithms and Engineering Optimization*. John Wiley & Sons, 2000.
- [Gfr00] GFRERRER, A.: Study's Kinematic Mapping — A Tool for Motion Design. In: LENARČIČ, J. (editor), STANIŠIĆ, M. M. (editor): *Advances in Robot Kinematics*. Springer Netherlands, 2000.
- [Gla10] GLAZUNOV, V.: Design of Decoupled Parallel Manipulators by Means of the Theory of Screws. *Mechanism and Machine Theory*, 45(2):239–250, 2010.
- [Gog02] GOGU, G.: Families of 6R Orthogonal Robotic Manipulators with Only Isolated and Pseudo-Isolated Singularities. *Mechanism and Machine Theory*, 37(11):1347–1375, 2002.
- [Gog04] GOGU, G.: Structural Synthesis of Fully-Isotropic Translational Parallel Robots Via Theory of Linear Transformations. *European Journal of Mechanics - A/Solids*, 23(6):1021–1039, 2004.
- [Gog05a] GOGU, G.: Chebychev Grübler Kutzbach's Criterion for Mobility Calculation of Multi-Loop Mechanisms Revisited Via Theory of Linear Transformations. *European Journal of Mechanics - A/Solids*, 24(3):427–441, 2005.
- [Gog05b] GOGU, G.: Evolutionary Morphology. In: TICHKIEWITCH, S. (editor) , TOLLENAERE, M. (editor) , RAY, P. (editor): *Advances in Integrated Design and Manufacturing in Mechanical Engineering*. Springer Netherlands, 2005.

- [Gog06a] GOGU, G.: Fully-Isotropic Parallel Manipulators with Five Degrees of Freedom. In: *Proceedings of the IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 1141–1146, 2006.
- [Gog06b] GOGU, G.: Fully-Isotropic Parallel Manipulators with Schönflies Motions and Complex Legs with Rhombus Loops. *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, (May):1147–1152, 2006.
- [Gog06c] GOGU, G.: Fully-Isotropic T3R2-Type Parallel Robotic Manipulators. In: *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics*, 1–6, 2006.
- [Gog07] GOGU, G.: Fully-Isotropic Three-Degree-of-Freedom Parallel Wrists. In: *Proceedings of the IEEE International Conference on Robotics and Automation Vol. 2*, 895–900, 2007.
- [Gog08] GOGU, G.: *Structural Synthesis of Parallel Robots. Part 1: Methodology*. Springer Netherlands, 2008.
- [Gog09a] GOGU, G.: *Structural Synthesis of Parallel Robots. Part 2: Translational Topologies with Two and Three Degrees of Freedom*. Springer Netherlands, 2009.
- [Gog09b] GOGU, G.: T2R1-Type Parallel Manipulators with Decoupled and Bifurcated Planar-Spatial Motion of the Moving Platform. In: *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2009.
- [Gog10] GOGU, G.: *Structural Synthesis of Parallel Robots. Part 3: Topologies with Planar Motion of the Moving Platform*. Springer Netherlands, 2010.
- [Gog12] GOGU, G.: *Structural Synthesis of Parallel Robots. Part 4: Other Topologies with Two and Three Degrees of Freedom*. Springer Netherlands, 2012.
- [Gog14] GOGU, G.: *Structural Synthesis of Parallel Robots. Part 5: Basic Overconstrained Topologies with Schönflies Motions*. Springer Netherlands, 2014.
- [Gos92] GOSSELIN, C. M.: The Optimum Design of Robotic Manipulators Using Dexterity Indices. *Journal of Robotics and Autonomous Systems*, 9(4):213–226, 1992.
- [Gro15] GROH, F.: *Numerische Verfahren für Polynomgleichungssysteme mit Anwendungen in der Robotik*, Universität Stuttgart, Stuttgart, Germany, Ph. D. Thesis, 2015.
- [GWXQ09] GUO, Z. , WANG, K. , XU, Z. , QI, H.: Topological Design and Genetic Synthesis of the Variable Topology Parallel Mechanisms. In: *Proceedings of the ASME/IFTOMM International Conference on Reconfigurable Mechanisms and Robots ReMAR*, pages 221–228, 2009.
- [Hag11] HAGN, U.: *The Aspect of Versatility in the Design of a Lightweight Robot for Surgical Applications*, Leibniz Universität Hannover, Hanover, Germany, Ph. D. Thesis, 2011.

- [HC15] HUANG, W., CAMPBELL, M.: Automated Synthesis of Planar Mechanisms with Revolute, Prismatic and Pin-in-Slot Joints. In: *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2015.
- [HEKO14] HANSEN, C. , EGGERS, K. , KOTLARSKI, J. , ORTMAIER, T.: Task Specific Trajectory Profile Selection for Energy Efficient Servo Drive Movements. In: *Proceedings of the 31st International Symposium on Automation and Robotics in Construction and Mining ISARC*, pages 514–522, 2014.
- [Her99] HERVÉ, J.: The Lie Group of Rigid Body Displacements, a Fundamental Tool for Mechanism Design. *Mechanism and Machine Theory*, 34(5):719–730, 1999.
- [HHIAA06] HASAN, A. , HAMOUDA, A. , ISMAIL, N. , AL-ASSADI, H.: An Adaptive-Learning Algorithm to Solve the Inverse Kinematics Problem of a 6 D.O.F Serial Robot Manipulator. *Advances in Engineering Software*, 37(7):432–438, 2006.
- [HKO13] HANSEN, C. , KOTLARSKI, J. , ORTMAIER, T.: Experimental Validation of Advanced Minimum Energy Robot Trajectory Optimization. In: *Proceedings of the 16th International Conference on Advanced Robotics ICAR*, pages 1–8, 2013.
- [HKO14] HANSEN, C. , KOTLARSKI, J. , ORTMAIER, T.: Optimal Motion Planning for Energy Efficient Multi-Axis Applications. *International Journal of Mechatronics and Automation IJMA*, 4(3):147–160, 2014.
- [HIMO12] HANSEN, C. , ÖLTJEN, J. , MEIKE, D. , ORTMAIER, T.: Enhanced Approach for Energy-Efficient Trajectory Generation of Industrial Robots. In: *Proceedings of the 2012 IEEE International Conference on Automation Science and Engineering CASE*, pages 1–7, 2012.
- [HM05] HAO, F., MERLET, J. P.: Multi-Criteria Optimal Design of Parallel Manipulators Based on Interval Analysis. *Mechanism and Machine Theory*, 40(2):157–171, 2005.
- [HMKPG16] HASSANZADEH, N. , MOVASSAGH-KHANI, R. , PEREZ-GRACIA, A.: Design of a Dexterous Hand for a Multi-Hand Task. In: PARENTI-CASTELLI, V. (editor), SCHIEHLEN, W. (editor): *ROMANSY 21 - Robot Design, Dynamics and Control. Proceedings of the 21st CISM-IFTOMM Symposium, June 20-23, Udine, Italy*. Springer International Publishing, 2016.
- [HPS07] HUSTY, M. , PFURNER, M. , SCHRÖCKER, H.-P: A New and Efficient Algorithm for the Inverse Kinematics of a General Serial 6R Manipulator. *Mechanism and Machine Theory*, 42(1):66–81, 2007.
- [HSJ+91] HERVÉ, J. , SPARACINO, F. , JAL, Y. , JAP, T. , HERVÉ, J.: Star, a New Concept in Robotics. In: *Proceedings of the 5th International Conference on Advanced Robotics*, 1991.

- [HSKC12] HYUNSEOP, L. , SOONWOONG, H. , KYOOSIK, S. , CHANGSOO, H.: Comparative Study of Optimization Technique for the Global Performance Indices of the Robot Manipulator Based on an Approximate Model. *International Journal of Control, Automation and Systems*, 10(2):374–382, 2012.
- [Hun73] HUNT, K.: Constant-Velocity Shaft Couplings: A General Theory. *Journal of Engineering for Industry*, 95(2):455–464, 1973.
- [IFR17] *International Federation of Robotics*: <http://www.ifr.org/industrial-robots/products/>. Accessed in October 2017.
- [igu17] *IGUS Robolink*: http://www.igus.de/wpck/6155/roboLink_beta. Accessed in October 2017.
- [JHT10] JIANG, H.-Z. , HE, J.-F. , TONG, Z.-Z.: Characteristics Analysis of Joint Space Inverse Mass Matrix for the Optimal Design of a 6-DOF Parallel Manipulator. *Mechanism and Machine Theory*, 45(5):722–739, 2010.
- [JJ13] JORDEHI, A., JASNI, J.: Parameter Selection in Particle Swarm Optimisation: A Survey. *Journal of Experimental & Theoretical Artificial Intelligence*, 25(4):527–542, 2013.
- [JSF07] JAFARI, A. , SAFAVI, M. , FADAEI, A.: A Genetic Algorithm to Optimum Dynamic Performance of Industrial Robots in the Conceptual Design Phase. In: *Proceedings of the IEEE 10th International Conference on Rehabilitation Robotics*, 2007.
- [KA00] KARLIK, B., AYDIN, S.: An Improved Approach to the Solution of Inverse Kinematics Problems for Robot Manipulators. *Engineering Applications of Artificial Intelligence*, 13(2):159–164, 2000.
- [KAOH09] KOTLARSKI, J. , ABDELLATIF, H. , ORTMAIER, T. , HEIMANN, B.: Enlarging the Useable Workspace of Planar Parallel Robots Using Mechanisms of Variable Geometry. In: *Proceedings of the ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots*, pages 94–103, 2009.
- [KB05] KUCUK, S., BINGUL, Z.: Robot Workspace Optimization Based on a Novel Local and Global Performance Indices. In: *Proceedings of the IEEE International Symposium on Industrial Electronics ISIE*, 2005.
- [KB06] KUCUK, S., BINGUL, Z.: Comparative Study of Performance Indices for Fundamental Robot Manipulators. *Robotics and Autonomous Systems*, 54(7):567–573, 2006.
- [KCAP07] KHAN, W. , CARO, S. , ANGELES, J. , PASINI, D.: A Formulation of Complexity-Based Rules for the Preliminary Design Stage of Robotic Architectures. In: *Proceedings of the International Conference on Engineering Design*, 2007.

- [KCZ12] KELAIAIA, R. , COMPANY, O. , ZAATRI, A.: Multiobjective Optimization of a Linear Delta Parallel Robot. *Mechanism and Machine Theory*, 50(0):159–178, 2012.
- [KD10] KUO, C., DAI, J.: Structural Synthesis of Serial Robotic Manipulators Subject to Specific Motion Constraints. In: *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2010.
- [KES01] KENNEDY, J. , EBERHART, R. , SHI, Y.: *Swarm Intelligence*. Morgan Kaufmann, 2001 (The Morgan Kaufmann Series in Artificial Intelligence).
- [KG06] K., Xianwen, GOSSELIN, C. M.: Type Synthesis of 4-DOF SP-Equivalent Parallel Manipulators: A Virtual Chain Approach. *Mechanism and Machine Theory*, 41(11):1306–1319, 2006.
- [KG07a] KONG, X., GOSSELIN, C.: Type Synthesis of 3-DOF Linear Translational Parallel Manipulators. In: *Proceedings of the 12th IFToMM World Congress*, pages 11–16, 2007.
- [KG07b] KONG, X., GOSSELIN, C.: *Type Synthesis of Parallel Mechanisms*. Springer Berlin Heidelberg, 2007.
- [KHO12] KOTLARSKI, J. , HEIMANN, B. , ORTMAIER, T.: Influence of Kinematic Redundancy on the Singularity-free Workspace of Parallel Kinematic Machines. *Frontiers of Mechanical Engineering*, 7(2):120–134, 2012.
- [Kir00] KIRCHNER, J.: *Mehrkriterielle Optimierung von Parallelkinematiken*, Technische Universität Chemnitz, Chemnitz, Germany, Ph. D. Thesis, 2000.
- [KK93] KIM, J., KHOSLA, P.: A Formulation for Task Based Design of Robot Manipulators. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, 1993.
- [KL85] KANE, T., LEVINSON, D.: *Dynamics, Theory and Applications*. McGraw Hill, 1985.
- [KLZ99] KONG, F. , LI, Q. , ZHANG, W.: An Artificial Neural Network Approach to Mechanism Kinematic Chain Isomorphism Identification. *Mechanism and Machine Theory*, 34(2):271–283, 1999.
- [KO93] KOHLI, D., OSVATIC, M.: Inverse Kinematics of General 6R and 5R,P Serial Manipulators. *ASME. Journal of Mechanical Design*, 115(4):922–931, 1993.
- [Kob15] KOBLER, J.-P.: *Ein knochenfixiertes, mechatronisches Assistenzsystem für die minimalinvasive Cochleaimplantat-Chirurgie*, Leibniz Universität Hannover, Hanover, Germany, Ph. D. Thesis, 2015.
- [Kon03] KONG, X.: *Type Synthesis and Kinematics of General and Analytic Parallel Mechanisms*, Université Laval, Quebec, Canada, Ph. D. Thesis, 2003.

- [Kon13] KONG, X.: Type Synthesis of 3-DOF Parallel Manipulators with Both a Planar Operation Mode and a Spatial Translational Operation Mode. *Journal of Mechanisms and Robotics*, 5(4):041015–041015–8, 2013.
- [Kot13] KOTLARSKI, J.: *Leistungssteigerung von Parallelrobotern basierend auf kinematischer Redundanz*, Leibniz Universität Hannover, Hanover, Germany, Ph. D. Thesis, 2013.
- [Kuk17a] KUKA LBR iiwa: <https://www.kuka.com/en-de/products/robot-systems/industrial-robots/lbr-iiwa>. Accessed in October 2017.
- [kuk17b] KUKA Robotics: <https://www.kuka.com>. Accessed in October 2017.
- [LA16] LÉGER, J., ANGELES, J.: Off-Line Programming of Six-Axis Robots for Optimum Five-Dimensional Tasks. *Mechanism and Machine Theory*, 100(Supplement C):155–169, 2016.
- [LH09] LEE, C., HERVÉ, J.: Type Synthesis of Primitive Schoenflies-Motion Generators. *Mechanism and Machine Theory*, 44(10):1980–1997, 2009.
- [LH10] LEE, C., HERVÉ, J.: Generators of the Product of Two Schoenflies Motion Groups. *European Journal of Mechanics - A/Solids*, 29(1):97–108, 2010.
- [LHH04] LI, Q. , HUANG, Z. , HERVÉ, J.: Type Synthesis of 3R2T 5-DOF Parallel Mechanisms Using the Lie Group of Displacements. *IEEE Transactions on Robotics*, 20(2):173–180, 2004.
- [LJ06] LAROCHELLE, A. P. P. Mand Murray M. P. Mand Murray, J., Angeles: A Distance Metric for Finite Sets of Rigid-Body Displacements via the Polar Decomposition. *ASME Journal of Mechanical Design*, 129(8):883–886, 2006.
- [LL05] LU, Y., LEINONEN, T.: Type Synthesis of Unified Planar-Spatial Mechanisms by Systematic Linkage and Topology Matrix-Graph Technique. *Mechanism and Machine Theory*, 40:1145–1163, 2005.
- [LLGCM10] LEGRIEL, J. , LE-GUERNIC, C. , COTTON, S. , MALER, O.: Approximating the Pareto Front of Multi-criteria Optimization Problems. In: *Proceedings of the 16th International Conference Tools and Algorithms for the Construction and Analysis of Systems TACAS*, 2010.
- [LLL00] LIU, M.-J. , LI, C.-X. , LI, C.-N.: Dynamics Analysis of the Gough-Stewart Platform Manipulator. *Transactions on Robotics and Automation*, 16(1):94–98, 2000.
- [LMRD10] LARA-MOLINA, F. A. , ROSARIO, J. M. , DUMUR, D.: Multi-Objective Design of Parallel Manipulator Using Global Indices. *The Open Mechanical Engineering Journal*, 4:37–47, 2010.

- [LMXL03] LIU, G. , MENG, J. , XU, J , LI, Z.: Kinematics Synthesis of Parallel Manipulators: a Lie Theoretic Approach. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2096–2100, 2003.
- [LWA17a] Schunk LWA 4D: http://de.schunk.com/sk_en/gripping-systems/series/lwa-4d/. Accessed in October 2017.
- [LWA17b] Schunk LWA 4P: https://schunk.com/de_en/gripping-systems/series/lwa-4p/. Accessed in October 2017.
- [LY17] LU, Y., YE, N.: Type Synthesis of Parallel Mechanisms by Utilizing Sub-Mechanisms and Digital Topological Graphs. *Mechanism and Machine Theory*, 109:39–50, 2017.
- [mar17] *Project AsRoBe (Assistive robot for people with physical disabilities)*: <http://asrobe.hs-weingarten.de/>. Accessed in October 2017.
- [MD06] MERLET, J. P., DANEY, D.: Legs Interference Checking of Parallel Robots Over a Given Workspace or Trajectory. In: *Proceedings of the IEEE International Conference on Robotics and Automation ICRA*, 2006.
- [MD08] MERLET, J. P., DANEY, D.: Appropriate Design of Parallel Manipulators. In: WANG, L (editor), XI, J (editor): *Smart Devices and Machines for Advanced Manufacturing*. Springer, 2008.
- [Mer97] MERLET, J. P.: Designing a Parallel Manipulator for a Specific Workspace. *International Journal of Robotics Research*, 16(4):545–556, 1997.
- [Mer05a] MERLET, J. P.: Jacobian, Manipulability, Condition Number, and Accuracy of Parallel Robots. *ASME. Journal of Mechanical Design*, 128(1):199–206, 2005.
- [Mer05b] MERLET, J. P.: Optimal Design of Robots. In: *Proceedings of Robotics: Science and Systems*, 2005.
- [Mer06] MERLET, J. P.: *Parallel Robots*. 2nd Edition. Springer-Verlag, 2006.
- [Mey00] MEYER, C.: *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics SIAM, 2000.
- [Mey01] MEYLAHN, A.: *Effiziente Algorithmen für die Steuerung von Werkzeugmaschinen mit Hexapod-Kinematik : Transformation, Kollisionsüberwachung und Freifahrerunterstützung*, Aachen, Germany, Ph. D. Thesis, 2001.
- [MHBJ13] MAZHOUD, I. , HADJ-HAMOU, K. , BIGEON, J. , JOYEUX, P.: Particle Swarm Optimization for Solving Engineering Problems: A New Constraint-Handling Mechanism. *Engineering Applications of Artificial Intelligence*, 26(4):1263–1273, 2013.
- [MLL07] MENG, J. , LIU, G. , LI, Z.: A Geometric Theory for Analysis and Synthesis of Sub-6 DOF Parallel Manipulators. *IEEE Transactions on Robotics*, 23(4):625–649, 2007.

- [MLSY09] MA, L. , LIU, A. , SHEN, H. , YANG, T.: A Method for Structure Synthesis of Reconfigurable Mechanisms Based on Genetic Optimization Algorithm. In: *Proceedings of the ASME IFToMM International Conference on Reconfigurable Mechanisms and Robots*, pages 148–152, 2009.
- [MS11] MCCARTHY, J., SOH, G.: *Geometric Design of Linkages*. Springer New York, 2011.
- [MSC⁺12a] MORENO, H. , SALTAREN, R. , CARRERA, I. , PUGLISI, L. , ARACIL, R.: Índices de Desempeño de Robots Manipuladores: una revisión del Estado del Arte. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 9(2):111–122, 2012.
- [MSC⁺12b] MORENO, H. , SALTAREN, R. , CARRERA, I. , PUGLISI, L. , ARACIL, R.: Performance Indices for Robotic Manipulators: a Review of the State of the Art. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 9(2):111–122, 2012.
- [MSPC16] MAZZOTTI, C. , SANCISI, N. , PARENTI-CASTELLI, V.: A Measure of the Distance Between Two Rigid-Body Poses Based on the Use of Platonic Solids. In: PARENTI-CASTELLI, V. (editor), SCHIEHLEN, W. (editor): *ROMANSY 21 - Robot Design, Dynamics and Control: Proceedings of the 21st CISM-IFTToMM Symposium, June 20-23, Udine, Italy*. Springer International Publishing, 2016.
- [NA89] NNAJI, B., ASANO, D.: Evaluation of Trajectories for Different Classes of Robots. *Robotics and Computer-Integrated Manufacturing*, 6(1):25–35, 1989.
- [Pau81] PAUL, R.: *Robot Manipulators: Mathematics, Programming and Control (Artificial Intelligence)*. MIT Press, 1981.
- [PB07] PEREZ, R., BEHDINAN, K.: Particle Swarm Approach for Structural Design Optimization. *Computers & Structures*, 85(19):1579–1588, 2007.
- [PC05] PUCHETA, M., CARDONA, A.: Type Synthesis of Planar Linkage Mechanisms with Rotoidal and Prismatic Joints. *Mecánica Computacional*, 24:2703–2730, 2005.
- [PC08] PUCHETA, M., CARDONA, A.: Synthesis of Planar Multiloop Linkages Starting from Existing Parts or Mechanisms: Enumeration and Initial Sizing. *Mechanics Based Design of Structures and Machines*, 36(4):364–391, 2008.
- [PC13] PUCHETA, M., CARDONA, A.: Topological and Dimensional Synthesis of Planar Linkages for Multiple Kinematic Tasks. *Multibody System Dynamics*, 29(2):189–211, 2013.
- [Pet08] PETERSSON, M.: *Design Optimization in Industrial Robotics: Methods and Algorithms for Drive Train Design*, Linköping University, Sweden, Linköping, Sweden, Ph. D. Thesis, 2008.
- [PGM06] PEREZ-GRACIA, A., MCCARTHY, J.: Kinematic Synthesis of Spatial Serial Chains Using Clifford Algebra Exponentials. *Proceedings of the Institution of Mechanical*

- Engineers, Part C: Journal of Mechanical Engineering Science*, 220(7):953–968, 2006.
- [Pie68] PIEPER, D.: *The Kinematics of Manipulators Under Computer Control*, Stanford, USA, Ph. D. Thesis, 1968.
- [PK93] PAREDIS, C., KHOSLA, P.: Kinematic Design of Serial Link Manipulators From Task Specifications. *The International Journal of Robotics Research*, 12(3):274–287, 1993.
- [PKPM⁺16] PERMIN, E. , KURILOVA-PALISAITIENE, J. , MANNHEIM, T. , BUHSE, K. , SCHMITT, R. , CORVES, B. , BJÖRKMAN, M.: Energy Efficiency Potentials of Different Robot Design Concepts. In: *Proceedings of the Robotics and Automated Production Lines* Vol. 840, pages 8–15, 2016 (Applied Mechanics and Materials).
- [PLC17] *PLCopen*: <http://www.plcopen.org/>. Accessed in October 2017.
- [Plu15] PLUMHOFF, S. (advisor: Ramirez D.: *Optimierung serieller Roboter unter Berücksichtigung unterschiedlicher Strukturen*, Institut für Mechatronische Systeme, Leibniz Universität Hannover, Hannover, Germany, Projektarbeit, 2015.
- [Plu16] PLUMHOFF, S. (advisor: Ramirez D.: *Optimierung serieller Roboter ohne vorangegangene Struktursynthese*, Institut für Mechatronische Systeme, Leibniz Universität Hannover, Hannover, Germany, Projektarbeit, 2016.
- [PO09] PETERSSON, M., ÖLVANDER, J.: Drive Train Optimization for Industrial Robots. *IEEE Transactions on Robotics*, pages 1419–1424, 2009.
- [Pro17] *Project JAMES (Joint Action for Multimodal Embodied Social Systems)*: <http://www.fortiss.org/forschung/projekte/james/>. Accessed in October 2017.
- [PS14] PATEL, S., SOBH, T.: Goal Directed Design of Serial Robotic Manipulators. In: *Proceedings of the Zone 1 Conference of the American Society for Engineering Education*, pages 1–6, 2014.
- [PS15a] PATEL, S., SOBH, T.: Manipulator Performance Measures - A Comprehensive Literature Survey. *Journal of Intelligent & Robotic Systems*, 77(3-4):547–570, 2015.
- [PS15b] PATEL, S., SOBH, T.: Task Based Synthesis of Serial Manipulators. *Journal of Advanced Research*, 6(3):479–492, 2015.
- [PS15c] PATEL, S., SOBH, T.: Using Task Descriptions for Designing Optimal Task Specific Manipulators. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, pages 3544–3551, 2015.
- [PSM⁺12] PUGLISI, L. , SALTAREN, R. , MORENO, H. , CÁRDENAS, P. , GARCIA, C. , ARACIL, R.: Dimensional Synthesis of a Spherical Parallel Manipulator Based on the Evaluation of Global Performance Indexes. *Robotics and Autonomous Systems*, 60(8):1037–1045, 2012.

- [RAF⁺07] RIZK, R. , ANDREFF, N. , FAUROUX, J. C. , LAVEST, J. , GOGU, G.: Precision Study of a Decoupled Four Degrees of Freedom Parallel Robot Including Manufacturing and Assembling Errors. In: TICHKIEWITCH, S. (editor) , TOLLENAERE, M. (editor) , RAY, P. (editor): *Advances in Integrated Design and Manufacturing in Mechanical Engineering*. Springer Netherlands, 2007.
- [RAGPP95] RANJBARAN, F. , ANGELES, J. , GONZÁLEZ-PALACIOS, M. A. , PATEL, R. V.: The Mechanical Design of a Seven-Axes Manipulator with Kinematic Isotropy. *Journal of Intelligent and Robotic Systems*, 14(1):21–41, 1995.
- [Rao09] RAO, S.: *Engineering Optimization: Theory and Practice*. John Wiley & Sons, 2009.
- [RDM⁺99] REICHENSPURNER, H. , DAMIANO, R. , MACK, M. , BOEHM, D. , GULBINS, H. , DETTER, C. , MEISER, B. , ELLGASS, R. , REICHART, B.: Use of the Voice-Controlled and Computer-Assisted Surgical System Zeus for Endoscopic Coronary Artery Bypass Grafting. *The Journal of Thoracic and Cardiovascular Surgery*, 118(1):11–16, 1999.
- [RHNT06] REFAAT, S , HERVÉ, J , NAHAVANDI, S , TRINH, H: Asymmetrical Three-DOFs Rotational-Translational Parallel-Kinematics Mechanisms Based on Lie Group Theory. *European Journal of Mechanics - A/Solids*, 25(3):550–558, 2006.
- [RJW06] ROOS, F. , JOHANSSON, H. , WIKANDER, J.: Optimal Selection of Motor and Gearhead in Mechatronic Applications. *Mechatronics*, 16(1):63–72, 2006.
- [RKL12] RAHMAN, T. , KROUGLICOF, N. , LYE, L.: Kinematic Synthesis of Nonspherical Orientation Manipulators: Maximization of Dexterous Regular Workspace by Multiple Response Optimization. *Journal of Mechanical Design*, 134(7):071009–071009–10, 2012.
- [RKO13] RAMIREZ, D. , KOTLARSKI, J. , ORTMAIER, T.: An Efficient Approach to Calculate the Workspace and Performance Indices of Serial Manipulators using a modified Quadtree/Octree Methodology. In: *DGR Tage 2013 (Deutsche Gesellschaft für Robotik)*, 2013.
- [RKO14] RAMIREZ, D. , KOTLARSKI, J. , ORTMAIER, T.: Automatic Generation of Serial Robot Architecture from Required Motion Directions. In: *Proceedings of the Symposium on Automated Systems and Technologies*, pages 55–64, 2014.
- [RKO15a] RAMIREZ, D. , KOTLARSKI, J. , ORTMAIER, T.: Automatic Generation of a Minimal Set of Serial Mechanisms for a Combined Structural - Geometrical Synthesis. In: *Proceedings of the 14th IFToMM World Congress*, 2015.
- [RKO15b] RAMIREZ, D. , KOTLARSKI, J. , ORTMAIER, T.: Automatic Generation of Serial Manipulators to Be Used in a Combined Structural Geometrical Synthesis. In: CORVES, Burkhard (editor) , LOVASZ, Erwin-Christian (editor) , HÜSING, Mathias

- (editor): *Mechanisms, Transmissions and Applications* Vol. 31. Springer International Publishing, 2015.
- [RKO16] RAMIREZ, D. , KOTLARSKI, J. , ORTMAIER, T.: Combined Structural and Dimensional Synthesis of Serial Robot Manipulators. In: PARENTI-CASTELLI, V. (editor), SCHIEHLEN, W. (editor): *ROMANSY 21 - Robot Design, Dynamics and Control. CISM International Centre for Mechanical Sciences (Courses and Lectures), vol 569*. Springer International Publishing, 2016.
- [RKO17] RAMIREZ, D. , KOTLARSKI, J. , ORTMAIER, T.: Combined Structural-Dimensional Synthesis of Robot Manipulators for Minimal Energy Consumption. In: SCHÜPPSTUHL, T. (editor) , FRANKE, J. (editor) , TRACHT, K. (editor): *Tagungsband des 2. Kongresses Montage Handhabung Industrieroboter*. Springer Vieweg, Berlin, Heidelberg, 2017.
- [ROS17] *ROS*: <http://www.ros.org/>. Accessed in October 2017.
- [RR93] RAGHAVAN, M., ROTH, B.: Inverse Kinematics of the General 6R Manipulator and Related Linkages. *ASME. Journal of Mechanical Design*, 115(3):502–508, 1993.
- [RSP⁺11] RUBRECHT, S. , SINGLA, E. , PADOIS, V. , BIDAUD, P. , BROISSIA, M. de: Evolutionary Design of a Robotic Manipulator for a Highly Constrained Environment. In: DONCIEUX, S. (editor) , BREDECHE, N. (editor) , MOURET, J.B. (editor): *New Horizons in Evolutionary Robotics*. Springer Berlin Heidelberg, 2011 (Studies in Computational Intelligence).
- [SC82] SALISBURY, J. K., CRAIG, J. J.: Articulated Hands: Force Control and Kinematic Issues. *The International Journal of Robotics Research*, 1(1):4–17, 1982.
- [Sch17a] *Schunk Handling Systems*: https://schunk.com/de_en/gripping-systems/category/gripping-systems/handling/. Accessed in October 2017.
- [Sch17b] *Schunk PR Rotary Units*: https://schunk.com/de_en/gripping-systems/series/pr/. Accessed in October 2017.
- [Sch17c] *Schunk Rotary Units*: https://schunk.com/de_de/greifsysteme/category/greifsysteme/handhabung/rotieren-schwenken-wenden/dreheinheiten-360-1/. Accessed in October 2017.
- [Sch17d] *Schunk Universal Linear Module LDN*: https://schunk.com/de_en/gripping-systems/series/ldn/. Accessed in October 2017.
- [SH93] SPARACINO, F., HERVÉ, J.: Synthesis of Parallel Manipulators Using Lie-Groups Y-STAR and H-ROBOT. In: *Proceedings of the IEEE/Tsukuba International Workshop on Advanced Robotics*, 75–80, 1993.
- [Sim13] SIMON, D.: *Evolutionary Optimization Algorithms*. John Wiley & Sons, 2013.

- [SK08] SICILIANO, B., KHATIB, O.: *Springer Handbook of Robotics*. Springer-Verlag Berlin Heidelberg, 2008.
- [SKK02] SHIAKOLAS, P. , KOLADIYA, D. , KEBRLE, J.: Optimum Robot Design Based on Task Specifications Using Evolutionary Techniques and Kinematic, Dynamic, and Structural Constraints. In: *Proceedings of the ASME International Mechanical Engineering Congress and Exposition, Dynamic Systems and Control*, 2002.
- [SL13] SUN, S., LIU, H.: Particle Swarm Algorithm: Convergence and Applications. In: YANG, Xin-She (editor) , CUI, Zhihua (editor) , XIAO, Renbin (editor) , GANDOMI, Amir H. (editor) , KARAMANOGLU, Mehmet (editor): *Swarm Intelligence and Bio-Inspired Computation*. Elsevier, 2013.
- [SS87] SCIAVICCO, L., SICILIANO, B.: Solving the Inverse Kinematic Problem for Robotic Manipulators. In: MORECKI, A. (editor) , BIANCHI, G. (editor) , KEDZIOR, K. (editor): *RoManSy 6: Proceedings of the Sixth CISM-IFTToMM Symposium on Theory and Practice of Robots and Manipulators*. Springer US, 1987.
- [SSPG14] SIMO-SERRA, E., PEREZ-GRACIA, A.: Kinematic Synthesis Using Tree Topologies. *Mechanism and Machine Theory*, 72:94–113, 2014.
- [SSVO09] SICILIANO, B. , SCIAVICCO, L. , VILLANI, L. , ORIOLO, G.: *Robotics. Modelling, Planning and Control*. 2nd Edition. Springer-Verlag London, 2009.
- [Ste12] STEWART, J.: *Multivariable Calculus*. 7th Edition. Brooks/Cole, CENGAGE Learning, 2012.
- [Tho16] THOMSEN, J. (advisor: Ramirez D.: *Automatisierte CAD-Modellierung synthetisierter Roboterkinematiken*, Institut für Mechatronische Systeme, Leibniz Universität Hannover, Hannover, Germany, Projektarbeit, 2016.
- [TK11] TOZ, M., KUCUK, S.: A Comparative Study for Computational Cost of Fundamental Robot Manipulators. In: *Proceedings of the IEEE International Conference on Industrial Technology*, 2011.
- [TKHO09] THANH, T. , KOTLARSKI, J. , HEIMANN, B. , ORTMAIER, T.: A New Program to Automatically Generate the Kinematic and Dynamic Equations of General Parallel Robots in Symbolic Form. In: *Proceedings of the 1st IFTToMM International Symposium on Robotics and Mechatronics*, 2009.
- [TPGP16] TAMIMI, A. , PEREZ-GRACIA, A. , PUCHETA, M.: Structural Synthesis of Hands for Grasping and Manipulation Tasks. In: LENARCIC, J. (editor), MERLET, J.P. (editor): *Advances in Robot Kinematics : Proceedings of the 15th international conference on Advances in Robot Kinematics*. 2016.
- [Tsa99] TSAI, L. W.: *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. John Wiley & Sons, 1999.

- [Tsa01] TSAI, L. W.: *Mechanism Design: Enumeration of Kinematic Structures According to Function*. Bd. 16. CRC, 2001.
- [Uni17] *Universal Robots*: <https://www.universal-robots.com/de/>. Accessed in October 2017.
- [VDI] *VDI-Richtlinie 2861 – Kenngrößen für Industrieroboter*.
- [VWT86] VIJAYKUMAR, R. , WALDRON, K. J. , TSAI, M. J.: Geometric Optimization of Serial Chain Manipulator Structures for Working Volume and Dexterity. *The International Journal of Robotics Research*, 5(2):91–103, 1986.
- [Wal07] WALLÉN, J.: On Robot Modelling Using Maple / Technical report from Automatic Control at Linköpings universitet. 2007. – Technical Report.
- [WBH15] WU, G. , BAI, S. , HJØRNET, P.: Parametric Optimal Design of a Parallel Schönflies-Motion Robot under Pick-and-Place Trajectory Constraints. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, 2015.
- [WC91] WANG, L., CHEN, C.: A Combined Optimization Method for Solving the Inverse Kinematics Problems of Mechanical Manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):489–499, 1991.
- [WE84] WOLOVICH, W., ELLIOTT, H.: A Computational Technique for Inverse Kinematics. In: *Proceedings of the 23rd IEEE Conference on Decision and Control*, pages 1359–1363, 1984.
- [Wei09] WEISE, T.: *Global Optimization Algorithms - Theory and Application*. 2nd Edition. it-weise.de, 2009.
- [Wer11] WERNER, D.: *Funktionalanalysis*. Springer Berlin Heidelberg, 2011.
- [WGLZ12] WU, W. , GUAN, Y. , LI, H. H. and Z. H. and Zhu , ZHANG, H.: Performance Analysis and Optimization of Serial Robots. In: *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, 2012.
- [WL89] WANG, K., LIEN, T.: Closed Form Solution for the Inverse Kinematics of a PUMA Robot Manipulator - II. Demonstration. *Robotics and Computer-Integrated Manufacturing*, 5(2):159–163, 1989.
- [YBPG14] YIHUN, Y. , BOSWORTH, K. , PEREZ-GRACIA, A.: Link-Based Performance Optimization of Spatial Mechanisms. *ASME. Journal of Mechanical Design*, 136(12):122303–122303–11, 2014.
- [YLJ⁺09] YANG, T.-L. , LIU, A.-X. , JIN, Q. , LUO, Y.-F. , SHEN, H.-P. , HANG, L.-B.: Position and Orientation Characteristic Equation for Topological Design of Robot Mechanisms. *Journal of Mechanical Design*, 131(2):021001–021001–17, 2009.
- [YLS⁺13] YANG, T.-L. , LIU, A.-X. , SHEN, H.-P. , LUO, Y.-F. , HANG, L.-B. , SHI, Z.-X.: On the Correctness and Strictness of the Position and Orientation Characteristic

- Equation for Topological Structure Design of Robot Mechanisms. *Journal of Mechanisms and Robotics*, 5(2):021009–021009–18, 2013.
- [YMPG⁺12] YIHUN, Y. , MIKLOS, R. , PEREZ-GRACIA, A. , REINKENSMeyer, D. , DENNEY, K. , WOLBRECHT, E.: Single Degree-of-Freedom Exoskeleton Mechanism Design for Thumb Rehabilitation. In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1916–1920, 2012.
- [Yos85] YOSHIKAWA, T.: Manipulability of Robotic Mechanisms. *The International Journal of Robotics Research*, 4(2):3–9, 1985.
- [Yum17] *ABB yumi*: <http://new.abb.com/future/de/yumi>. Accessed in October 2017.
- [ZBH11] ZHOU, L. , BAI, S. , HANSEN, M.: Design Optimization on the Drive Train of a Light-Weight Robotic Arm. *Mechatronics*, 21(3):560 –569, 2011.
- [ZBH12] ZHOU, L. , BAI, S. , HANSEN, M.: Integrated Dimensional and Drive-Train Design Optimization of a Light-Weight Anthropomorphic Arm. *Robotics and Autonomous Systems*, 60(1):113–122, 2012.
- [ZN10] ZHANG, X., NELSON, C.: Multiple-Criteria Kinematic Optimization for the Design of Spherical Serial Mechanisms Using Genetic Algorithms. In: *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 2: 34th Annual Mechanisms and Robotics Conference, Parts A and B*, 2010.
- [ZZLC09] ZHAN, Z. , ZHANG, J. , LI, Y. , CHUNG, H.: Adaptive Particle Swarm Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, (39):1362–1381, 2009.

Curriculum Vitae

Personal information

Name Ramirez Rodriguez, Daniel Andres
Birth 26 November 1981 in Duitama, Colombia
Family status married, one son
Nationality colombian

Professional experience

since Mar/17 Project engineer (mechanical design) at Yuanda Robotics GmbH
Jan/13 - Mar/17 Ph. D. student, Institute of Mechatronic Systems (imes), LUH
Oct/10 - Jun/12 Lecturer and junior researcher, Saint Tomas University,
Bogota, Colombia
Oct/04 - Aug/06 Development engineer, MAZDA Motor Corporation,
Bogota, Colombia

Studies

Aug/06 - Oct/10 Automation engineering (M. Sc.), National University of Colombia,
Bogota, Colombia
Feb/99 - Oct/04 Mechanical engineering (B. Sc.), National University of Colombia,
Bogota, Colombia