

# Combined Structural and Dimensional Synthesis of Serial Robot Manipulators

D. Ramirez, J. Kotlarski and T. Ortmaier

**Abstract** The present work addresses the unification of the structural and dimensional synthesis of serial robot manipulators. Unlike related publications, the approach does not utilize simplifications of the kinematic structures. In addition, it is also capable of generating manipulators with up to six degrees of freedom. The approach reduces the computational effort by automatically generating the task-suitable architectures together with their corresponding optimisation parameters. Since the kinematics modelling is one of the main challenges, an algorithm for the numerical solution of the inverse kinematics is introduced. Finally, in order to demonstrate the capability of the method, the optimal manipulator for a pick and place operation is found using kinematic performance indices and particle swarm optimisation.

**Keywords** Serial manipulators · Robot design · Structural synthesis · Dimensional synthesis · Denavit-Hartenberg parameters

## 1 Introduction

The robot design process currently involves two isolated procedures: the structural synthesis and the dimensional synthesis (Merlet 2005). The first aims to find the number, type, and orientation of the robot joints in order to generate a manipulator

---

D. Ramirez (✉) · J. Kotlarski · T. Ortmaier  
Institute of Mechatronic Systems, Leibniz Universität Hannover, Hannover, Germany  
e-mail: danielandres.ramirez@imes.uni-hannover.de; danielramirez@usantotomas.edu.co;  
andresr36@gmail.com

J. Kotlarski  
e-mail: jens.kotlarski@imes.uni-hannover.de

T. Ortmaier  
e-mail: tobias.ortmaier@imes.uni-hannover.de

D. Ramirez  
Universidad Santo Tomás, Bogotá, Colombia

with a desired number of degrees of freedom (DOF). For this purpose, several methods like the Lie groups (Caro et al. 2010) and the screw theory (Kuo and Dai 2010) have been employed. In the dimensional synthesis, the optimal dimensions of the robot links are determined in order to accomplish a given task (Carbone et al. 2007). For this optimisation, many kinds of performance indices (Patel and Sobh 2015) as well as optimisation algorithms (Carbone et al. 2007; Lara-Molina et al. 2010) have been proposed.

Although the methods for the structural and dimensional synthesis are well established, the architectures produced as a result of the structural synthesis are not considered in the dimensional synthesis. Instead, the optimisation of the links dimensions is performed only for one structure, which is usually chosen a priori. Thus, the obtained manipulator that results from the dimensional synthesis is not necessarily the best mechanism for the given task.

Integrated synthesis of planar mechanisms has been developed for rigid body guidance in (Luu and Hayes 2012; Pucheta and Cardona 2011). Furthermore, dual quaternions have been used (Perez-Gracia and McCarthy 2006) in the synthesis of spatial mechanisms from a prescribed set of finite positions. Some authors have tried to unify the two aforementioned procedures for task-specific serial manipulators (defining the task as one or several paths). However, these attempts consider only manipulators with three DOF (Kucuk and Bingul 2006), simplified architectures (e.g. the last three axes intersecting at one point) (Patel and Sobh 2014, 2015), or a limited set of link geometries (i.e. the possible components are predefined modules) (Rubrecht et al. 2011).

The present work introduces a combined (structural and dimensional) synthesis of serial robot manipulators which does not assume any structural simplification. It is able to synthesize manipulators with up to six DOF while taking prismatic (P) and revolute (R) joints into account. In order to reduce the computational effort, the generation of all task-suitable architectures as well as the extraction of the minimal optimisation parameters is performed as the first step of the approach. Then, a generic numerical solution of the inverse kinematics for serial manipulators is presented. The optimisation procedure for an exemplary pick and place task with four DOF is addressed in the fourth section. Finally, the method's capability is demonstrated by synthesizing a manipulator for the mentioned task.

## 2 Generation of Suitable Architectures and Extraction of the Optimisation Parameters

The generation of suitable architectures to perform a required task is summarized here. A detailed description of the algorithm was previously published (Ramirez et al. 2015).

The architecture and geometry of a serial manipulator are described for the combined synthesis through a symbolic matrix of Denavit-Hartenberg (DH) parameters.

The first part of the synthesis aims to find the parameters that can be used during the optimisation as variables and the parameters that have to be constant in order to provide the required end effector (EE) DOF.

The necessary DOF to execute a given task can be used to define the required motion vector  $\xi_{req}$  as

$$\xi_{req} = [\xi_{req_1}, \xi_{req_2}, \xi_{req_3}, \xi_{req_4}, \xi_{req_5}, \xi_{req_6}]^T, \tag{1}$$

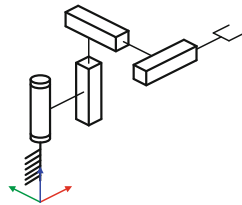
where  $\xi_{req_1}, \xi_{req_2},$  and  $\xi_{req_3}$  represent the translational DOF in direction of the axes  $x, y,$  and  $z$  while  $\xi_{req_4}, \xi_{req_5},$  and  $\xi_{req_6}$  represent the rotational DOF around the axes  $x, y,$  and  $z,$  respectively. It has to be noticed that  $\xi_{req}$  corresponds to the EE velocity of a serial manipulator  $\dot{\xi}$ :

$$\dot{\xi} = [\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6]^T = \begin{bmatrix} v_n \\ \omega_n \end{bmatrix} = J\dot{q}, \tag{2}$$

with  $J$  being the Jacobian matrix of the manipulator,  $v_n = [\xi_1, \xi_2, \xi_3]^T$  the EE linear velocity,  $\omega_n = [\xi_4, \xi_5, \xi_6]^T$  the EE angular velocity, and  $\dot{q}$  the joint rates vector. For a given task, if  $\xi_{req_i} = 0$  for any  $i = 1 \dots 6,$  then  $\xi_i$  should be also zero for the same  $i$  and for any  $\dot{q}$ . The suitability of a manipulator with respect to the task is evaluated using three conditions. Firstly, the null terms of  $\xi_{req}$  and  $\xi$  have to be the same. Secondly, the rank of the Jacobian matrix ( $\text{rank}(J)$ ) must be equal to the total number of required DOF. Thirdly, the independence of each EE DOF with respect to each DH parameter has to be verified. This evaluation is conducted in symbolic form in order to perform a global analysis. After the application of this conditions to the generated manipulators, it is possible to obtain a minimum set of architectures (i.e. without isomorphisms) to be considered during the subsequent optimisation.

Exemplarily, Fig. 1 shows a RPPP architecture, which is one of the 17 architectures that fulfil a required motion vector  $\xi_{req} = [\xi_{req_1}, \xi_{req_2}, \xi_{req_3}, 0, 0, \xi_{req_6}]^T$  (Ramirez et al. 2015). The corresponding DH parameters as well as the vector  $p$  with the optimisation parameters are also presented. In this case, the fixed parameters are  $\alpha_2 = \pi/2, \theta_3 = \pi/2,$  and  $\alpha_3 = \pi/2.$

**Fig. 1** Exemplary architecture generated for a task that requires three translational DOF and one rotational DOF around the z-axis (Ramirez et al. 2015)



R/P	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
R	$q_1$	$d_1$	$a_1$	$\alpha_1$
P	$\theta_2$	$q_2$	$a_2$	$\pi/2$
P	$\pi/2$	$q_3$	$a_3$	$\pi/2$
P	$\theta_4$	$q_4$	$a_4$	$\alpha_4$

$$p = [d_1, a_1, \alpha_1, \theta_2, a_2, a_3, \theta_4, a_4, \alpha_4]$$

### 3 Kinematics Modelling

A central problem of the combined structural-dimensional synthesis is the solution of the inverse kinematics for a general serial manipulator including R and P joints. In this paper, the inverse kinematics is solved using a numerical approach, which exploits the advantages of both the pseudoinverse and the transpose of the Jacobian matrix.

The kinematics of a serial manipulator can be expressed as the relationship between the joint coordinates vector  $\mathbf{q}$  and the Cartesian coordinates vector (position and orientation) of the EE  $\mathbf{x}_{EE}$ :

$$\mathbf{x}_{EE} = \mathbf{f}(\mathbf{q}) . \quad (3)$$

The inverse kinematics problem consists of calculating  $\mathbf{q}$  from a given  $\mathbf{x}_{EE}$ :

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x}_{EE}) . \quad (4)$$

Most widespread numerical methods to solve (4) employ either the pseudoinverse or the transpose of the Jacobian  $\mathbf{J}$ . The first case corresponds to the Newton-Raphson method to solve nonlinear kinematic equations (Schwarz and Köckler 2011). The approach approximates the solution of the inverse kinematics  $\mathbf{q}$  to the vector  $\mathbf{q}_{p+1}$  using:

$$\mathbf{q}_{p+1} = \mathbf{q}_p + \mathbf{J}(\mathbf{q}_p)^\dagger \mathbf{K} \Delta \mathbf{x}_p , \quad (5)$$

with

$$\Delta \mathbf{x}_p = \mathbf{x}_{EE} - \mathbf{f}(\mathbf{q}_p) , \quad (6)$$

$$\mathbf{J}^\dagger = \mathbf{J}^\top (\mathbf{J}\mathbf{J}^\top)^{-1} , \quad (7)$$

and  $\mathbf{K}$  being a diagonal matrix to adjust the step size of the algorithm.

At the beginning of the algorithm  $\mathbf{q}_p$  corresponds to a given initial joint coordinates vector  $\mathbf{q}_0$ , which can be arbitrarily chosen. Though the approach usually has a high convergence rate, it diverges when  $\mathbf{J}(\mathbf{q}_p)$  is singular.

Another numerical method to solve the inverse kinematics corresponds to the gradient method for the solution of a system of nonlinear equations (Schwarz and Köckler 2011). This utilizes the transpose of  $\mathbf{J}$  in a similar way as (5):

$$\mathbf{q}_{p+1} = \mathbf{q}_p + \mathbf{J}(\mathbf{q}_p)^\top \mathbf{K} \Delta \mathbf{x}_p . \quad (8)$$

Since the inversion of the Jacobian matrix is not necessary, this approach does not present problems with singularities. However, the convergence is slower in comparison to the Newton-Raphson method.

In order to take advantage of both approaches, a combined numerical solution of the inverse kinematics is proposed. This begins by using (5) and calculates the condition number  $\kappa$  of the Jacobian matrix  $\mathbf{J}(\mathbf{q}_p)$  at each step  $p$ . If  $\kappa \geq \kappa_{\max}$ , the algorithm switches to use (8) until  $\kappa < \kappa_{\max}$  and  $\mathbf{q}_p$  converges again. Then, Eq. (5) is used again to continue the process. The threshold  $\kappa_{\max}$  for the maximal allowed value of the condition number is given arbitrarily. Numerous tests showed that the algorithm is able to overcome singularities during the process and even find solutions near to singular poses with an appropriated convergence rate for  $200 < \kappa_{\max} < 600$ . The process is performed until  $\Delta \mathbf{x}_p < \epsilon$  for a given tolerance  $\epsilon$ . The proposed algorithm is presented in Algorithm 1. Detailed features of the algorithm are not discussed here due to space limitations.

---

**Alg. 1** Algorithm for the numerical solution of the inverse kinematics problem

---

**Require:**  $\mathbf{x}_{EE}$ ,  $\mathbf{q}_0$ ,  $\epsilon$ ,  $\kappa_{\max}$

```

 $\mathbf{q}_p \leftarrow \mathbf{q}_0$ 
repeat
   $\kappa \leftarrow \text{cond}(\mathbf{J}(\mathbf{q}_p))$ 
   $\Delta \mathbf{x}_p \leftarrow \mathbf{x}_{EE} - \mathbf{f}(\mathbf{q}_p)$ 
  if  $\kappa < \kappa_{\max}$  and  $\Delta \mathbf{x}_p < \Delta \mathbf{x}_{p-1}$  then
     $\mathbf{q}_{p+1} \leftarrow \mathbf{q}_p + \mathbf{J}(\mathbf{q}_p)^\dagger \mathbf{K} \Delta \mathbf{x}_{EE}$  {(from (5))}
  else
     $\mathbf{q}_{p+1} \leftarrow \mathbf{q}_p + \mathbf{J}(\mathbf{q}_p)^\text{T} \mathbf{K} \Delta \mathbf{x}_{EE}$  {(from (8))}
  end if
   $\mathbf{q}_p \leftarrow \mathbf{q}_{p+1}$ 
until  $\Delta \mathbf{x}_p < \epsilon$ 
return  $\mathbf{q}_p$ 

```

---

## 4 Optimisation Procedure

Due to the reasonable number of task-suitable architectures found through the generation of suitable architectures (Sect. 2), each of them can be optimised. Afterwards, the optimal configuration will be the one with the best performance of all architectures. The optimisation can be conducted for many kinds of requirements, e.g. accuracy, manipulability, size of the workspace, dynamical performance, etc. Exemplarily, the synthesis of a serial manipulator for a pick and place task will be introduced. Since the robot size influences other performance characteristics like stiffness and energy consumption, it will be used as optimisation criterion. Furthermore, the accuracy in the start pose ( $\mathbf{x}_{EE,s}$ ) and end pose ( $\mathbf{x}_{EE,e}$ ) are used as additional requirements of the design process. The indices that are used to evaluate the performance of the manipulator regarding these requirements as well as the optimisation algorithm are explained in the next sections.

## 4.1 Performance Indices

For the considered pick and place task, the robot size should be minimized maintaining a predefined accuracy in the start and end poses.

Several indices can be found in literature to evaluate the robot size. In this paper, the robot size index  $f_{\text{size}}$  is based on the structural length index (Waldron 2008), which compares the sum of the link lengths  $a_i$  and the joint offsets  $d_i$  to the volume of the reachable workspace. In our approach, the distance between the start and end points  $L_{\text{task}}$  will be used instead:

$$f_{\text{size}} = \frac{\sum_{i=1}^n |a_i| + |d_i|}{L_{\text{task}}}. \quad (9)$$

Since the condition number of  $\mathbf{J}$  ( $\kappa(\mathbf{J})$ ) is generally employed to analyse the error of the EE due to errors in the actuators, the local conditioning index (*LCI*) (Angeles and López-Cajún 1992) provides a good measure of the accuracy of the manipulator. However, in order to avoid the dimensional inhomogeneities of  $\mathbf{J}$ , the dimensionally homogeneous Jacobian matrix  $\tilde{\mathbf{J}}$  (Ranjbaran et al. 1995) must be used instead. The *LCI* is evaluated at the initial ( $\mathbf{q}_{\text{ini}}$ ) and end ( $\mathbf{q}_{\text{end}}$ ) point of the task. Then, the performance index corresponding to the accuracy of the manipulator is the minimum of these two values:

$$LCI_{\min} = \min \left( \kappa^{-1}(\tilde{\mathbf{J}}(\mathbf{q}_{\text{ini}})), \kappa^{-1}(\tilde{\mathbf{J}}(\mathbf{q}_{\text{end}})) \right). \quad (10)$$

## 4.2 Optimisation Problem

Exemplarily, the index  $f_{\text{size}}$  will be used as cost function and the minimum local conditioning index  $LCI_{\min}$  as constraint. Therefore, the optimal geometric parameters vector  $\mathbf{p}^*$  has to be found:

$$\mathbf{p}^* = \arg \min_{\mathbf{p}} (f_{\text{size}}(\mathbf{p})), \quad (11)$$

subject to

$$LCI_{\min} \geq LCI_{\min}^*, \quad (12)$$

where  $LCI_{\min}^*$  defines the lowest allowed value of  $LCI_{\min}$ . The parameters vector  $\mathbf{p}_{\min} \leq \mathbf{p} \leq \mathbf{p}_{\max}$  (lower bound  $\mathbf{p}_{\min}$ , upper bound  $\mathbf{p}_{\max}$ ) is obtained for each suitable architecture as explained in Sect. 2.

Since it is not possible to obtain an explicit expression for the cost function, a global optimisation method has to be used. In this case, the particle swarm optimisation algorithm (PSO) (Ebbesen et al. 2012) was chosen to solve the optimisation

problem. The optimisation process is performed for each suitable architecture. At the end of the process, the configuration (i.e. architecture including geometrical parameters) with the best performance is selected to perform the required task.

### 5 Exemplary Results

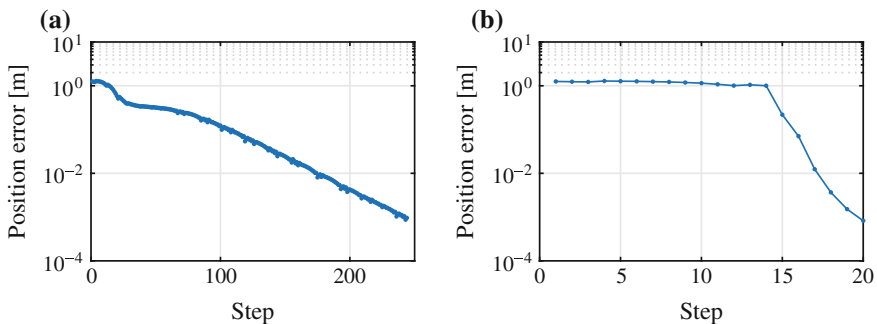
In order to demonstrate the capability of the proposed approach, the results of the synthesis for the aforementioned task are presented. This task requires three translational DOF and one rotational DOF around the vertical axis. The corresponding required motion vector is  $\xi_{req} = [\xi_{req_1}, \xi_{req_2}, \xi_{req_3}, 0, 0, \xi_{req_6}]^T$ . The required start and end poses of the EE ( $x_{EE,s}$  and  $x_{EE,e}$ ) are described using the position and orientation of the EE coordinate system (CS)<sub>EE</sub> with respect to the robot base coordinate system (CS)<sub>0</sub>. Table 1 gives the coordinates  $x, y, z$  and  $\varphi$  of the start and end poses.

The optimisation process was performed as explained in Sect. 4 for every suitable architecture (Ramirez et al. 2015). In each iteration of the optimisation, the inverse kinematics as introduced in Sect. 3 is computed to determine the joint coordinates corresponding to  $x_{EE,s}$  and  $x_{EE,e}$ . Afterwards, a point to point (PTP) interpolation is calculated in the joint space using a polynomial function of degree five (Biagiotti and Melchiorri 2008) in order to ensure the task feasibility.

The improvement in the numerical solution of the inverse kinematics is visualised in Fig. 2. As an example, for an RRRP architecture, the numerical inverse kinematics was performed with  $q_0 = [\pi/4, 0, \pi/4, 0.05]^T$  and  $x_{EE} = x_{EE,s}$ . Since  $q_0$

**Table 1** Start and end poses of the required task

Pose	$x$ [m]	$y$ [m]	$z$ [m]	$\varphi$ [rad]
$x_{EE,s}$	-0.5	0	0	$-\pi/2$
$x_{EE,e}$	0.5	0.6	0.7	0



**Fig. 2** Evolution of the position error during the solution of the inverse kinematics **a** Using the transpose of  $J$  **b** Using the solution proposed in Algorithm 1

**Table 2** Architectures with the best performance after the optimisation procedure

Architecture	RPPP	RPRP	RPPR	PRPR	RRRP	PPPR
$f_{\text{size}}$	0.9613	1.0089	1.0881	1.0881	1.1212	1.2460
$LCI_{\text{min}}$	0.3613	0.1768	0.3163	0.2776	0.1	0.3077

**Table 3** DH parameters of the optimised manipulator

R/P	$\theta_i$ [rad]	$d_i$ [m]	$a_i$ [m]	$\alpha_i$ [rad]
R	$q_1$	0.318	0.227	0.5665
P	-1.9369	$q_2$	-0.058	$\pi/2$
P	$\pi/2$	$q_3$	0	$\pi/2$
P	-2.8224	$q_4$	0	0.9230

corresponds to a singular pose, the proposed algorithm (Fig. 2b) switches to use (8), but after step 13, it uses (7) for rapid convergence. Using only  $\mathbf{J}^T$  the convergence of the algorithm is slower (Fig. 2a). The represented position error corresponds to the norm of the first three terms of  $\Delta \mathbf{x}_p$ .

The architectures with the highest performance after the optimisation (RPPP, RPRP, RPPR, PRPR) and its performance indices are reported in Table 2. These results were obtained using 100 particles (swarm size) for each architecture and  $LCI_{\text{min}}^* = 0.1$ . The optimisation of every architecture was carried out several times with similar results.

The DH parameters of the best manipulator (RPPP) are shown in Table 3. As can be seen, the dimensions of the manipulator are reasonable for the given task. The capability of the method is demonstrated comparing the indices of the serial architectures that are usually employed for pick and place operations, namely PPPR (Gantry robot) and RRRP (Scara robot). The performance obtained for these manipulator are clearly poorer than the manipulator obtained with our method as reported in Table 2.

## 6 Conclusions

The paper presents an approach to combine the structural and dimensional synthesis of serial manipulators. Starting from a required task, every suitable architecture is considered during the dimensional optimisation process. In order to reduce the computational effort, the appropriate geometrical parameters of each architecture are deduced automatically. Due to an efficient numerical solution of the inverse kinematics, mechanisms with up to six DOF can be synthesized without any kinematic simplification. The benefits of considering all task-suitable architectures are observed



in the design of the manipulator for an exemplary pick and place operation. Here, the size of the robot was minimized and the accuracy was considered as a constraint during the optimisation. The obtained manipulator has a higher performance than the architectures typically employed for this kind of tasks.

## References

- Angeles, J., & López-Cajún, C. (1992). Kinematic isotropy and the conditioning index of serial robotic manipulators. *The International Journal of Robotics Research*, 11(6), 560–571.
- Biagiotti, L., & Melchiorri, C. (2008). *Trajectory Planning for Automatic Machines and Robots*. Berlin: Springer.
- Carbone, G., Ottaviano, E., & Ceccarelli, M. (2007). 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.
- Caro, S., Khan, W., Pasini, D., & Angeles, J. (2010). The rule-based conceptual design of the architecture of serial schoenflies-motion generators. *Mechanism and Machine Theory*, 45(2), 251–260.
- Ebbesen, S., Kiwitz, P., & Guzzella, L. (2012). A generic particle swarm optimization matlab function. In: *Proceedings of the American Control Conference (ACC 2012)* (pp. 1519–1524).
- Kucuk, S., & Bingul, Z. (2006). Comparative study of performance indices for fundamental robot manipulators. *Robotics and Autonomous Systems*, 54(7), 567–573.
- Kuo, C., & Dai, J. (2010) Structural synthesis of serial robotic manipulators subject to specific motion constraints. In: *Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. Montreal.
- Lara-Molina, F. A., Rosario, J. M., & Dumur, D. (2010) Multi-objective design of parallel manipulator using global indices. *The Open Mechanical Engineering Journal* 4 (2010).
- Luu, T. J., & Hayes, M. (2012). Latest Advances in Robot Kinematics, chap. Integrated Type and Dimensional Synthesis of Planar Four-Bar Mechanisms (pp. 317–324). Springer Netherlands.
- Merlet, J. (2005). Optimal design of robots. In: *Proceedings of Robotics: Science and Systems*. Cambridge, USA.
- Patel, S., & Sobh, T. (2014). Using task descriptions for designing optimal task specific manipulators. In: *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)* (pp. 755–760).
- Patel, S., & Sobh, T. (2015). Manipulator performance measures—a comprehensive literature survey. *Journal of Intelligent & Robotic Systems*, 77(3–4), 547–570.
- Patel, S., & Sobh, T. (2015). Task based synthesis of serial manipulators. *Journal of Advanced Research*, 6(3), 479–492.
- Perez-Gracia, A., & McCarthy, J. M. (2006). 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.
- Pucheta, M. A., & Cardona, A. (2011). Topological and dimensional synthesis of planar linkages for multiple kinematic tasks. *Multibody System Dynamics*, 29(2), 189–211.
- Ramirez, D., Kotlarski, J., & Ortmaier, T. (2015). Automatic generation of a minimal set of serial mechanisms for a combined structural - geometrical synthesis. In: *Proceedings of the 14th IFToMM World Congress*.
- Ranjbaran, F., Angeles, J., Gonzalez-Palacios, M., & Patel, R. (1995). The mechanical design of a seven-axes manipulator with kinematic isotropy. *Journal of Intelligent and Robotic Systems*, 14(1), 21–41.
- Rubrecht, S., Singla, E., Padois, V., Bidaud, P., & de Broissia, M. (2011). Evolutionary design of a robotic manipulator for a highly constrained environment. In S. Doncieux, N. Bredeche, & J.

- Mouret (Eds.), *New Horizons in Evolutionary Robotics, Studies in Computational Intelligence* (Vol. 341, pp. 109–121). Berlin: Springer.
- Schwarz, H., & Köckler, N. (2011). *Numerische Mathematik*. Vieweg Teubner Verlag.
- Waldron, K.J. (2008). Design of arms. In: Dorf, R., Nof, S. (Eds.) *International Encyclopedia of Robotics: Applications and Automation* (Vol. 1), chap. A. Wiley.