Planning of Workplaces with Multiple Kinematically Redundant Robots

Rainer Konietschke

2007

Lehrstuhl für Realzeit-Computersysteme
Technische Universität München

Planning of Workplaces with Multiple Kinematically Redundant Robots

Rainer Konietschke

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: ....
Prüfer der Dissertation:
1. ....
2. ....

Die Dissertation wurde am .... bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am .... angenommen.

Abstract

This thesis provides new methods for planning and optimization of robotic workplaces, i.e. workplaces with assistance of a robotic system. The robots may work autonomously or in cooperation with man. The thesis covers cooperation between multiple robots and lays stress on robots with kinematic redundancy. As highly demanding application area, optimal design and preoperative planning for minimally invasive and open robotic surgery is chosen.

Optimization and planning of robotic workplaces are important instruments to cope with the increased complexity of today's robotic applications and to ensure safe operation. Algorithms and devices to facilitate and partially automatize planning and optimization are, however, barely existent so far – with currently available tools the user mostly has to resort to a trial and error approach. Especially for complicated tasks requiring e.g. several robots cooperating in an unstructured environment, it is very improbable that a good (if at all sufficient) setup of the robotic workplace can be found this way. Therefore, the inclusion of algorithms to automatize the planning procedure as presented in this thesis is the evident next step to be taken.

Closed form solutions for inverse kinematics and singularities provide the core of a reliable workplace optimization system and are developed in this thesis for serial kinematically redundant robots. Unlike state of the art methods, the presented inverse kinematics computation does not suffer from algorithmic singularities. The thesis describes an accordingly developed software library for inverse kinematics and shows both a planning procedure involving the medical robot KineMedic, and a real-time application, providing inverse kinematics for Cartesian control of the robotic system Justin with a computation time of lower than 0.6 ms for all 18 considered joints.

Based on the closed form solutions, the thesis presents a complete procedure for workplace optimization and robot synthesis that uses a two-step algorithm based on Genetic Algorithms and a subsequent Sequential Quadratic Programming method. The thesis develops several optimization criteria and demonstrates the performance of the methods with a preoperative planning procedure as well as with the kinematic design optimization of the KineMedic system.

The algorithms implemented in this thesis help the human during the decision taking procedure, by e.g. providing a preselection of (according to the chosen criteria) good solutions or by carrying out an optimization in a certain subspace while leaving the determination of the remaining parameters to the human. The thesis facilitates the transfer of a chosen setup into the real environment using a handheld contact-free surface-based registration procedure with an overall worst case error of 3 mm and a new handheld device to automatically project relevant structures into the real environment.

Zusammenfassung

Diese Dissertation stellt neue Methoden zur Planung und Optimierung robotischer Arbeitsplätze vor. Das robotische System arbeitet dabei entweder autonom oder in Kooperation mit dem Menschen. Die Arbeit behandelt auch die Kooperation mehrerer Roboter in unstrukturierten Umgebungen, und legt besonderes Gewicht auf robotische Systeme mit Redundanz. Sie erörtert Anwendungen innerhalb des anspruchsvollen Gebiets der minimal invasiven und offenen Chirurgierobotik im Bereich der Designoptimierung und präoperativen Planung.

Methoden zur Optimierung und Planung robotischer Arbeitsplätze helfen, die hohe Komplexität heutiger Robotikanwendungen in den Griff zu bekommen und sorgen für sicheren Betrieb. Algorithmen und Geräte, die den Optimierungs- und Planungsprozess vereinfachen und teilweise automatisieren, sind allerdings bis jetzt kaum vorhanden, der Benützer ist bei derzeit verwendeten Hilfsmitteln auf systematisches Ausprobieren angewiesen. Insbesondere bei komplizierten Aufgaben, welche etwa den Einsatz mehrerer kooperierender Roboter in unstrukturiertem Umfeld erfordern, ist es unwahrscheinlich, dass auf diese Weise ein guter (oder gar ausreichender) Aufbau des robotischen Arbeitsplatzes erreicht werden kann. Der Einbezug von Algorithmen zur Automatisierung der Planungsprozedur wie in dieser Arbeit vorgestellt ist daher der nächste Schritt.

Geschlossene Lösungen der Inverskinematik und der Singularitätsberechnung bilden den Kern einer zuverlässigen Arbeitsplatzoptimierung, sie werden in dieser Arbeit für redundante, serielle Roboter neu entwickelt. Anders als Methoden des aktuellen Forschungsstands leidet die entwickelte Berechnungsmethode für die Inverskinematik nicht unter algorithmischen Singularitäten. Die Arbeit präsentiert die entstandene Software-Bibliothek zum einen im Rahmen der Designoptimierung des medizinischen Roboters KineMedic, zum anderen in einer Echtzeitanwendung zur kartesischen Regelung des robotischen Systems Justin mit Rechenzeiten unter 0.6 ms für alle 18 betrachteten Gelenke.

Die Arbeit stellt dann, basierend auf den geschlossenen Lösungen, die gesamte Prozedur zur Arbeitsplatz- und Designoptimierung vor. Die Optimierung ist in einem zweistufigen Verfahren realisiert und verwendet zunächst genetische Algorithmen und im zweiten Schritt ein Sequential Quadratic Programming Verfahren. Die Arbeit entwickelt zahlreiche Optimierungskriterien und zeigt die Performanz des gesamten Verfahrens durch eine präoperative Planungsprozedur und die Designoptimierung des KineMedics-Roboters.

Die in dieser Arbeit entwickelten Algorithmen unterstützen den Menschen in der Entscheidungsfindung, indem sie entweder eine Vor-Auswahl von – hinsichtlich der gewählten Kriterien optimierter – Lösungen treffen, oder eine Optimierung in einem Unterraum des Problems durchführen und die restlichen Parameter der Bestimmung durch den Bediener überlassen. Die Arbeit vereinfacht die Übertragung des gewählten Aufbaus in das reale Umfeld durch eine neue Registrierungsmethode, basierend auf einer handgehaltenen und kontaktlosen Datenerfassung mit einer Gesamtgenauigkeit von – im schlimmsten Fall – 3 mm und durch ein neuartiges handgeführtes Gerät, welches automatisch die relevanten Planungsdaten in die reale Umgebung Patienten projiziert.

Acknowledgments

I received a lot of support from many sides while writing this Ph.D. thesis at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR), and I am deeply grateful for that. Firstly, I would like to thank Prof. G. Hirzinger for supervising my thesis and for his great efforts in providing the best possible research conditions. It is a great privilege to work in the Institute of Robotics and Mechatronics. I also like to thank Prof. G. Färber from the Institute for Real-Time Computer Systems at the Technical University of Munich for co-supervising the Ph.D. thesis and offering invaluable advice and support on many occasions.

The great team spirit in the institute and in the medical group helped me a lot while writing this thesis. Many thanks to my colleagues. It was really interesting and exciting to work with you.

Some experiments of the work were assisted by student projects and master theses – I am thankful for these contributions and enjoyed the collaboration a lot.

I would also like to thank the members of the service team of the institute. Always with a smile, they helped to overcome any bureaucratic obstacle.

Especially during the last phase of writing this thesis, I received a lot of support from Tobias, Mareike, Michael, Franziska, and my aunt Christina. Thank you for helping me by proofreading, discussing, and catering. Your support is apparent throughout the whole thesis. I would be proud to give something back to you some day. Many thanks to Astrid for painting the picture on the title page.

I am greatly thankful to be surrounded by nice and understanding friends. All of you encouraged me during the last years, which certainly must have been difficult once in a while. Thank you. To conclude, I would like to thank my parents and my family for the unceasing support, trust, and love.

Meiling, June 2007 Rainer Konietschke

Contents

Part I.

Introduction

The beginning is half the whole
Aristotle

Planning of how to achieve the whole may be considered as the first step, the beginning. It therefore has, according to Aristotle, a considerable influence on the final outcome of a task. In a robotic workplace as shown in Fig. 0.1, a task is carried out by a robotic system, possibly in cooperation with a human. Adequate planning is crucial in this case to fully exploit the potential of these workplaces and to guarantee error-free operation.

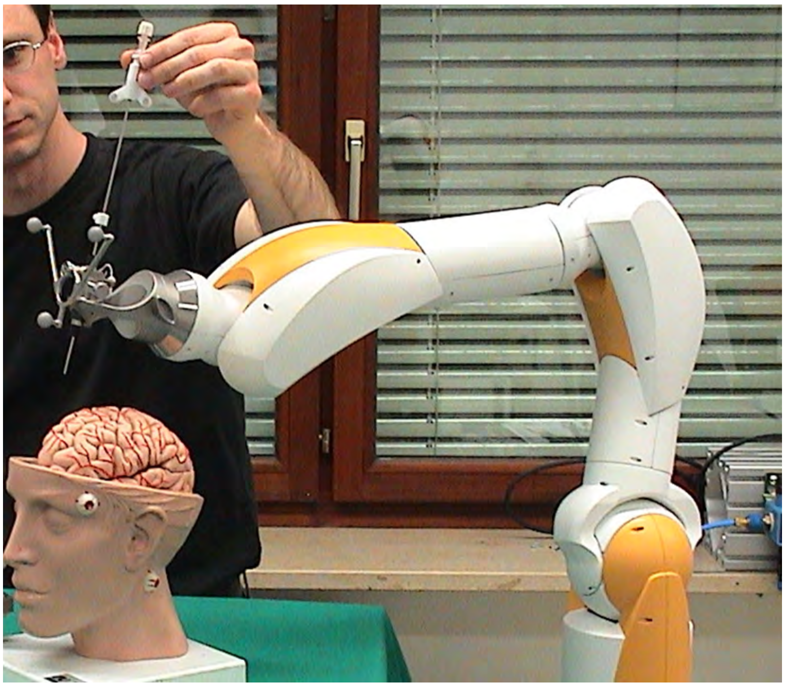


Figure 0.1.: A robotic workplace in the medical context: The medical robot KineMedic aims at combining the strengths of human and robot.

This thesis presents new algorithms and methods for planning and optimization of robotic workplaces in order to proceed from currently manual approaches to procedures that include algorithms to assist the user and to automatize some steps of the procedure.

The aim of the following chapters is to motivate the presented research and to give an outline of the thesis structure. Therefore, the next section provides a brief overview of the considered research topic, the field of robotic workplaces. Starting from the state of the art, some – up to now – open potentials of planning procedures are then presented. Chapter 2 summarizes the goals and presents the according thesis structure.

1. Introduction to Planning and Optimization of Robotic Workplaces

Robotics comprises a wide range of machines, all of them capable of performing physical tasks. Robots[1] take on many different forms, ranging from humanoid, which mimic the human form and way of moving, to industrial, whose appearance is dictated by the function to be performed. Robots can be grouped generally as mobile robots (e.g. autonomous vehicles), manipulator robots (e.g. industrial robots) and self reconfigurable robots, which can conform themselves to the task at hand.

Since the implementation of the first industrial robot Unimate in 1961 [76], robots have been mainly used in industry for repetitive, fast and accurate execution of tasks such as pick and place, screwing, cutting, or welding. With the use of sensors, the robot can react to the environment, for example to measure distances, forces, temperatures etc., and tasks can be executed according to the sensor data. A robotic system can be split into the basic components *actuators*, *sensors* and *robot control* (see Fig. 1.1):

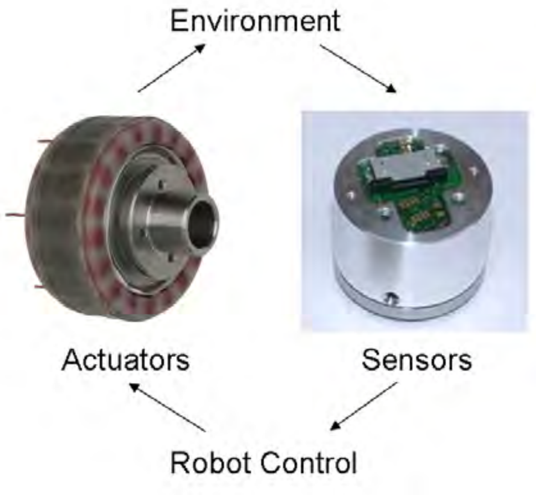


Figure 1.1.: Components of a robotic system.

[1] The word "robot" is also used in a general sense to mean any machine which mimics the actions of a human (bio mimicry), in the physical sense or in the mental sense. It comes from the Czech and Slovak word robota, labor or work (also used in the sense of a serf). The word robot first appeared in Karel Capek's science fiction play R.U.R. (Rossum's Universal Robots) in 1921.

An actuator is the mechanism by which an agent acts upon an environment. The agent can be either an artificial intelligent system or any other autonomous being (human, other animal, etc). In the field of robotics, this may be e.g. a grasping mechanism to move parts. In engineering, actuators are a subdivision of transducers. They are devices which transform an input signal (mainly an electrical signal) into motion. Electrical motors, pneumatic actuators, hydraulic pistons, relays, comb drive, and piezoelectric actuators are some examples of such actuators. Motors are mostly used when circular motions are needed, but can also be used for linear applications by transforming circular to linear motion with a bolt and screw transducer. On the other hand, some actuators are intrinsically linear, such as piezoelectric actuators.

A sensor is a physical device or biological organ that detects, or senses, a signal or physical condition and chemical compounds. Most sensors are electrical or electronic, although other types exist. A sensor is a type of transducer. Sensors are either directly indicating (e.g. a mercury thermometer or electrical meter) or are paired with an indicator (perhaps indirectly through an analog to digital converter, a computer and a display) so that the value sensed becomes readable by humans. In addition to other applications, sensors are heavily used in medicine, industry and robotics. Since a significant change involves an exchange of energy, sensors can be classified according to the type of energy transfer that they detect (thermal, electromagnetic, mechanical, chemical, optical and radiation).

Robot control is the theory of how to model and control robots. Basic concepts of robot control comprise position control on joint level and on Cartesian level, and impedance control. An important issue is the use of a real-time operating system to guarantee a fixed control rate. In many applications, high rates are needed to allow for fast reaction and stability, as e.g. in haptic feedback. Calculations performed in the control loop therefore have to be computationally efficient and deterministic.

The robotic workplace serves, similar to the workplace for humans, to fulfill a certain task or group of tasks and should be designed in a way as to provide optimal conditions according to reasonable criteria (e.g. cycle time, energy efficiency, safety or quality). Robotic workplaces may comprise several robotic systems, different other devices, the workpiece, and in particular may have strong interaction with humans. While robotic systems were mostly working separately from humans in the past, the interaction of humans and robots has become an important research topic in the last years. Tasks in service and medical robotics require close interaction between robots and human. Also in industrial applications the human-robot cooperation is advantageous, e.g. when teaching a robot by haptic interaction, i.e. by touching the robot and moving it according to the desired trajectory.

A main characteristic feature of the task execution at a robotic workplace is the operation mode of the robotic system. The main operation modes for a robotic system are classified in the next section.

1.1. Operation Modes and Human-Robot Cooperation

Robots may be controlled directly by a human, such as remotely-controlled bomb-disposal robots, robotic arms, or shuttles, or may act according to their own decision making ability, based on cognition and artificial intelligence. However, the majority of robots fall in-between these extremes, being controlled by pre-programmed computers. Such robots may include feedback loops to interact with their environment, but do not display actual intelligence.

Currently, the interaction of human and robot is prevented by safety measures in most industrial applications. However, potential advantages of close cooperation between human and robotic systems are evident and intensively investigated in research fields such as service robotics or medical robotics. One goal is to combine the strengths of human and robot. Humans are e.g. able to utilize qualitative information and to evaluate unclear information whereas robots provide high accuracy, various control schemes and can apply defined forces.

In the following paragraphs[2], robotic systems are classified according to their degree of autonomy into teleoperated systems, semi-autonomous systems with human robot cooperation, and completely autonomous systems[3].

1.1.1. Teleoperated Systems

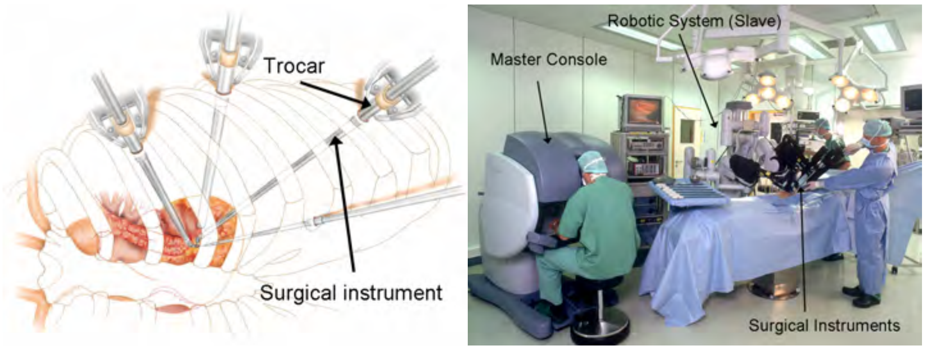


Figure 1.2.: In minimally invasive surgery, the operation field is reached through small incisions with long, thin instruments (left). In minimally invasive robotic surgery, these instruments are held by a robotic system and controlled by the surgeon through a master console (right). Pictures according to [41].

[2] Here, examples are taken out of the field of medical robotics, see a detailed description of such systems in [101].

[3] Clearly, the term autonomy (from Greek: Auto-Nomos – nomos meaning "law": one who gives oneself his own law) as used in philosophy implies the capacity of a rational individual to make an informed, uncoerced decision. Given this definition, current robotic systems are much closer to automatic (= self operating) machines than to autonomous machines. However, in the robotic community both terms *autonomous* and *automatic* are used nearly as synonyms.

Teleoperation indicates operation of a machine (usually a robot) at a distance. A prominent example of teleoperation is the minimally invasive robotically assisted surgery (MIRS) shown in Fig. 1.2. In MIRS applications, the field of operation is reached through small incisions[4] into the human body, using thin cylindrical instruments that are teleoperated by systems such as the daVinci$^{\text{TM}}$ [34] or ZEUS$^{\text{TM}}$ [88]. Certain autonomy functions like tremor filtering or motion compensation [77] can possibly be added. If a robotic system is used in teleoperation mode, the trajectories and actions of the system can be foreseen only very roughly beforehand. Therefore, for planning and optimization of the robotic workplace a considerable amount of tolerance with respect to the assumed nominal trajectories must be taken into account.

1.1.2. Human Robot Cooperation

In semi-autonomous systems, i.e systems with human robot cooperation, the task execution is shared between operator and robot, see e.g. the independent alignment of the robot-guided instrument with a located tumor [39] or the active constraint robot *AcroBot* for knee endoprothetics [43]. The goal of human robot cooperation in general is to combine the strengths of man and robot (see Table 1.1) in order to increase the performance of a certain task.

Table 1.1.: Strengths of man and robot (according to [26]).

Man	Robot
Good hand-eye coordination	High geometric accuracy
High skillfulness	No tremor or fatigue
Flexible and adaptable	Application of defined forces possible
Able to utilize qualitative information	Able to work in hazardous environments
Able to evaluate unclear information	Integration of different sensors and control strategies possible

Another example of human robot cooperation is the application shown in Fig. 1.3, using the KineMedic system to assist the surgeon as an *intelligent stand* in tasks such as biopsies and the navigated drilling of pedicle screws [81]. The robot can be re-positioned by simply touching and moving the robotic structure (see Fig. 1.3), and the surgeon is then assisted in guiding the robot to the preoperatively planned position by means of virtual fixtures[5]. Patient safety is increased through virtual safety barriers during robot removal. Experimental results suggest that these hands-on concepts clearly augment the quality of task execution [64].

[4] In this thesis, the location of the incisions is also denoted as *port* or *trocar point*. A trocar is a hollow cylinder with a sharply pointed end, that is used to introduce surgical instruments into body cavities (see Fig. 1.2 left).

[5] Virtual fixtures are used to reduce the Cartesian or joint space reachable by the robot. Volumes that the robot should not move into can be excluded this way, e.g. to prevent the robot from colliding with the environment.

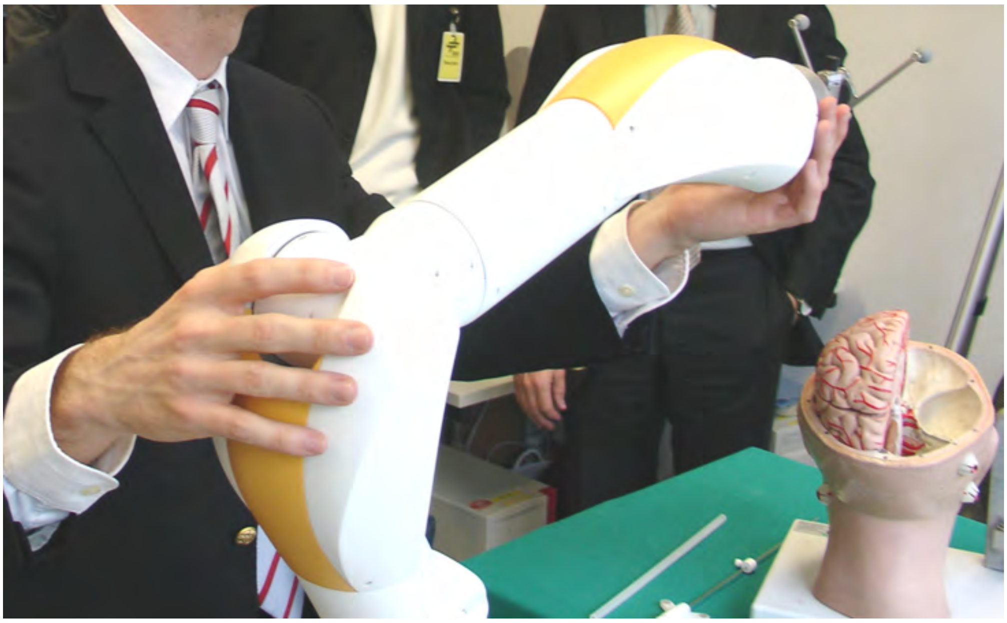


Figure 1.3.: Hands-on the medical robot KineMedic.

In industrial applications, human robot cooperation can be used to facilitate the teaching procedure: The operator moves the (gravity compensated, zero-torque controlled) robotic manipulator along the trajectories during the teaching procedure, and the robot will then repeat the trajectories with high accuracy and at scalable velocity autonomously, possibly including some optimization algorithms to e.g. smooth the trajectory, minimize torques or maximize the execution speed. Concepts to allow for safe interaction between human and robot are a current matter of research, see e.g. [3, 35, 112].

1.1.3. Autonomous Task Execution – Robotics and Cognition

Today, many automatic robotic systems are used in industry for e.g. positioning or pick-and-place tasks. Automatic systems are available in surgical robotics as well, providing e.g. fully automatic endoscope guidance via real-time image processing [107]. However, it has been reported that acceptance by the operator increases if the operator is closely included into the workflow and especially into the decision making [43].

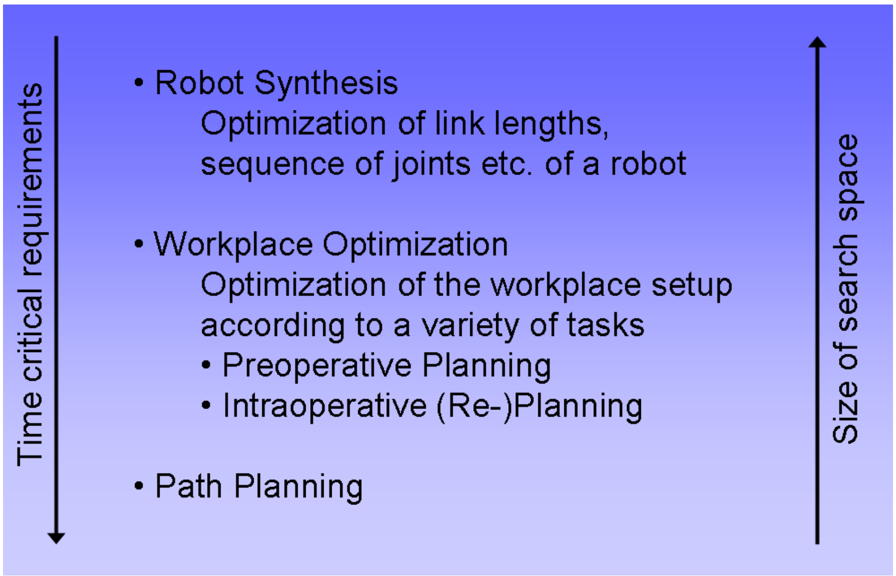
Cognitive abilities[6] and artificial intelligence as key features for autonomous robotic systems, however, are mainly subject to experimental platforms so far, see e.g. [6, 7, 20, 21, 103, 104, 108]. Especially, cognitive systems so far lack of reliability which is crucial when closely interacting with the operator.

[6] The term cognition (Latin: cognoscere, "to know") is used in several loosely related ways to refer to a faculty for the human-like processing of information, applying knowledge and changing preferences.

1.2. Potential of Planning and Optimization Methods for Robotic Workplaces

Customizing the robotic system according to the intended application is crucial for high performance. In the field of kinematics, optimal design of robotic systems is considered a major research topic with many unsolved problems [71]. In optimal kinematic design of robotic systems, one is concerned with finding e.g. the best sequence of joints, link lengths, base position, or joint limits according to a given task. Most of the unsolved problems described in [71] arise from the challenge of performing an optimization that involves multiple criteria and many parameters, resulting in a large search space. To handle multiple criteria, either weighting of the various criteria with respect to each other has to be performed, or methods such as interval analysis [36] can be applied in certain cases. Due to the large search space, computation time is a crucial issue when exhaustive (global) optimization has to be done. For neither of the mentioned problems a general solution is yet available.

Different planning and optimization tasks are described in the following. These tasks can be roughly put in order according to time-critical requirements and search space size as shown in Fig. 1.4.



Time critical requirements

- Robot Synthesis
  Optimization of link lengths, sequence of joints etc. of a robot
- Workplace Optimization
  Optimization of the workplace setup according to a variety of tasks
  - Preoperative Planning
  - Intraoperative (Re-)Planning
- Path Planning

Size of search space

Figure 1.4.: Optimization tasks in the context of robotics.

1.2.1. Robot Synthesis

Robot synthesis (optimal robot design) denotes the procedure of determining a suitable kinematic structure of a robot matching the required task or group of tasks. Unless the system belongs to the family of reconfigurable robots, a robot synthesis is usually done once during the design phase of the robot. According to the intended tasks, the sequence of joints, degrees of freedom (DoF) and geometric data such as link lengths have to be determined. Extensive simulation and mechanism analysis is reasonable since the search space is rather high. While many approaches exist for individual robotic structures (see e.g. [36, 83]), the methods presented in this thesis prove to be well suited to find optimal robot kinematics for a variety of robot synthesis problems (see e.g. [30, 47, 53] and Sect. 6.1).

1.2.2. Workplace Optimization

Given a robotic system and a set of well-defined tasks, the workplace optimization procedure determines optimal relative positions of the workpiece, the robotic system and other devices. Powerful tools to assist during the planning procedure are e.g. eM-Workplace [102] or KUKA.Sim [60]. Both tools, however, do not provide algorithms to optimize the workplace automatically. The user has to suggest a setup, and the software then allows for simulation. Inverse kinematics for non-redundant kinematic chains can be specified by the user, redundancy resolution is not yet available.

To help close these gaps, a main goal of this thesis therefore is

- to develop algorithms to allow for automatization of certain subtasks during planning and
- to provide methods to include redundancy resolution and thus to be able to plan on robots with redundant DoF.

An interesting and highly demanding application of workplace optimization is preoperative planning and intraoperative (re-)planning for MIRS. It was therefore chosen as testbed for the methods developed in this thesis and is presented in the next sections.

1.2.2.1. Preoperative Planning for Minimally Invasive Robotic Surgery

The use of teleoperated robots in the operating room (OR) for minimally invasive surgery (MIS) has been investigated closely in the past years both in abdominal and heart surgery [15, 23, 28, 33, 45, 70, 82, 85]. One key aspect necessary for a successful intervention is preoperative planning. Preoperative planning is done by the surgeon in order to prepare the intervention and to decide about the best access to the surgical site. In case of robotically assisted interventions the results of these decisions must be transferred also to the robotic equipment. Ports and robots must be placed so that no collisions between the robotic arms occur, the robotic system always provides sufficient dexterity, the view at the surgical site is optimal and the patient's trauma becomes minimal. The different criteria which must be met for an optimal port and robot placement can hardly be satisfied using an trial and error approach or just by experience. For that reason the use of optimization algorithms for preoperative planning seems to be necessary to ensure a high quality of interventions in the context of MIRS.

1.2.2.2. Intraoperative (Re-)Planning

State of the art planning procedures in MIRS are based on preoperative imaging data, hence requiring an accurate registration[7] between the preoperative data and the intraoperative situation to transfer the results of the planning procedure into the operating room. With registration usually a time-consuming step and a potential source of inaccuracy (due to e.g. insufflation[8], soft tissue displacement, breathing, and the surgical intervention itself), it would be advantageous to only use intraoperative data for planning. Thus planning takes place in the OR coordinate system, and no registration is necessary. Furthermore, if the OR setup or the task has to be modified due to e.g. complications in a previous step, it is necessary to adapt the plan to the current situation.

Both for preoperative and intraoperative planning, means for fast intraoperative replanning of a given task are highly desirable.

1.2.3. Path Planning

In path planning, one is usually interested in finding a good path from the current position to a goal position, optimizing certain criteria along the path such as distance from (possibly constantly moving) obstacles. Many custom algorithms were developed in the last years (see e.g. [5, 46, 59]), and even though many methods developed in this thesis are also relevant to path planning, it is not the main focus here.

1.3. Conclusion

Various application areas for planning and optimization methods in robot kinematics are shown in this chapter. Except for robot synthesis and path planning, the currently used procedures are mainly based on manual trial and error approaches. This can be justified for simple environments with only a few devices. For complicated tasks however, requiring e.g. several robotic systems, the manually determined setups of the robotic workplace may not perform sufficiently. Algorithms for optimization and automatization of the planning procedure can assist in this case and will be presented in this thesis.

[7] Registration in this context denotes the procedure of finding a transformation rule for transfer of the planning data into the coordinate system of the operating room. See Sect. 4.4 for further details.

[8] In order to expand a body cavity and increase workspace, or reduce obstruction during investigative surgery, an inert, non-toxic gas, such as carbon dioxide is often insufflated.

2. Contributions and Thesis Structure

As shown in the previous chapter, the optimization and planning of robotic workplaces holds a high potential for algorithms to facilitate the procedure. Clearly, it is neither possible to predict the task to be performed with complete certainty, nor is it reasonable to completely remove human reasoning from the planning procedure. However, algorithms can help the human during the decision-making procedure, by e.g. providing a preselection of (according to the chosen criteria) good solutions or by optimization in a certain subspace while leaving the determination of the remaining parameters to the human (see 1. Item below). Efficient optimization strategies (see 2. Item below) as well as fast and robust calculation of inverse kinematics (see 3. Item below) are crucial in this context.

Also the transfer of a chosen setup into the real environment can be assisted by algorithms, facilitating e.g. the registration (see 4. Item below) and positioning procedures (see 5. Item below).

Summarized, the main contributions of this thesis are addressed by the following items:

1. Workflow: Methods for the automatization of both workplace optimization and robot synthesis.

2. Optimization method: Efficient optimization strategy using Genetic Algorithms (GA) with a subsequent Sequential Quadratic Programming (SQP) method to determine a variety of optimal solutions.

3. Kinematics: Fast inverse kinematics for a variety of applications (workplace optimization, path planning, Cartesian control) and robot kinematics (including redundancy resolution, passive joints, constrained joints, serial and tree structures). Closed form solutions for determination of singular configurations.

4. Registration: Handheld contact-free surface-based registration with the DLR[1] 3D-Modeller.

5. Intraoperative Positioning: Development of a handheld device to automatically project relevant structures onto the patient surface.

[1] German Aerospace Center

Thesis Structure

The thesis is composed of four parts as shown in Fig. 2.1. The introduction and motivation is given in Part I. Part II presents the used theories and methods. Robot kinematics is revisited in Chapter 3 and leads to a robust knowledge about the inverse kinematics and singularities of some families of robot kinematics. Optimization and planning methods for robotically assisted interventions and for robot kinematics are addressed in Chapter 4.

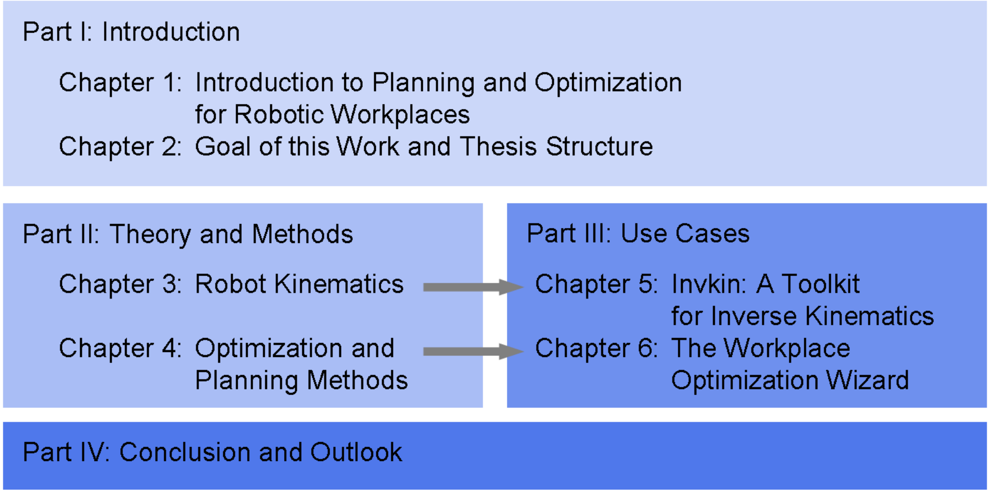

Part I: Introduction
Chapter 1: Introduction to Planning and Optimization for Robotic Workplaces
Chapter 2: Goal of this Work and Thesis Structure
Part II: Theory and Methods
Chapter 3: Robot Kinematics
Chapter 4: Optimization and Planning Methods
Part III: Use Cases
Chapter 5: Invkin: A Toolkit for Inverse Kinematics
Chapter 6: The Workplace Optimization Wizard
Part IV: Conclusion and Outlook

Figure 2.1.: Structure of the thesis.

Part III presents the use cases that illustrate the methods developed in Part II. In Fig. 2.1, the corresponding chapters are connected with arrows. In Chapter 5, the software library *InvKin* is described. It allows for fast calculation of a variety of kinematic structures, including redundancy resolution and tree structures. An implementation for planning procedures involving the DLR medical robot is presented, and real-time capabilities are shown through Cartesian control for the DLR robotic system Justin. Chapter 6 presents the conceived software suite *Workplace Optimizer Wizard*. Two applications are shown: Preoperative planning of robotically assisted minimally invasive surgeries and the kinematic design optimization of the medical robot KineMedic.

The thesis conclusion and an outlook are given in Part IV.

Part II.

Theory and Methods

This part presents the theories and methods used for the planning of workplaces with multiple kinematically redundant robots. It is subdivided into two chapters:

In Chapter 3, robot kinematics is developed. Special attention is thereby paid to kinematically redundant structures which are exemplified by the DLR medical robot KineMedic shown in Fig. 2.2. Inverse position kinematics is derived in closed form, thus allowing for solvability and fast computation. This is especially important in workplace optimization tasks where usually a large number of robot configurations has to be evaluated using the inverse kinematics to perform reachability checks. A closed form solution to determine all singular configurations of the DLR medical robot is then presented, which is crucial to guarantee a workspace free from singularities. Combining both the closed form solutions for inverse kinematics computation and determination of the singularities, a new concept for the computation of inverse position kinematics of kinematically redundant robots is derived. Eventually, the chapter is summarized and concluded.

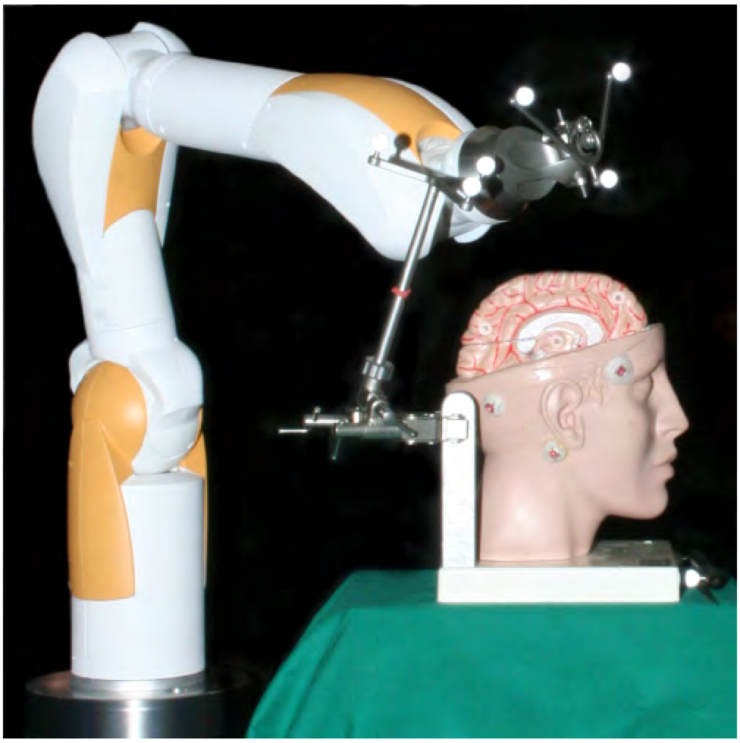


Figure 2.2.: The DLR medical robot KineMedic.

Based on the thorough analysis of robot kinematics in Chapter 3, Chapter 4 describes the methods necessary to set up a procedure for planning and optimization of robotic workplaces as well as for robot synthesis. Firstly, workflows of planning and optimization procedures are therefore determined. Optimization methods are then summarized and chosen according to the specific problem of workplace optimization and robot synthesis.

Relevant optimization criteria are derived, and new methods for registration and for assisting in the intraoperative positioning are presented. The chapter concludes with a short summary.

The results of this part are presented in various use cases in Part III. Namely, the performance of the inverse kinematics algorithms is shown, and the algorithms for robot synthesis are used to optimize the size of the KineMedic robot as well as of an actuated carrier. Furthermore, a workplace optimization is presented for a minimally invasive procedure with three KineMedic robots.

3. Robot Kinematics

This chapter deals with some important issues of robot kinematics found in kinematically redundant serial structures as exemplified in Sect. 3.1 with the DLR medical robot. The inverse position kinematics is revisited in Sect. 3.2, and three closed form solutions are derived for the DLR medical robot. Section 3.3 presents the determination of all singularities in closed form for a kinematically redundant robot. To this day, procedures to determine a closed form solution of the inverse position kinematics have been presented for various robotic structures [19, 58, 84, 96]. In case of redundancy usually a parametrization is necessary. If only a single parametrization is used for redundancy resolution, algorithmic singularities occur. To the author's knowledge, this problem has not yet been addressed and will be solved in Sect. 3.4. The chapter is concluded and summarized in Sect. 3.5.

3.1. The DLR Medical Robot KineMedic

Most of the theoretic results of the next sections are demonstrated by applying them to the DLR medical robot KineMedic which therefore is presented in the following according to [52]. The next section gives an overview of the mechanical and electrical design implementation of the robot, and Sect. 3.1.2 describes the kinematic structure of the robot in minimally invasive surgery applications.

3.1.1. General Description

The design objective of the robot was to build a compact, intuitively operated robot arm for a wide field of medical applications [52, 80]. Besides the classic pre-programmed and telemanipulated operation, the robot shall also be suited for semi autonomy where the surgeon can directly interact with the robot.

The medical robot has a slender design to comply with the very restricted space in the operating field and the tight interaction of the surgeon with the robot. A slender, compact arm reduces the space conflicts (e.g. risk of collision) and thus increases the acceptance of the robot by the surgeon. According to previous experience at the Institute of Robotics and Mechatronics in the field of light-weight robotics (LWR I-III [38]), the medical arm exhibits (a) joint redundancy: a flexible setup in the space-confined operation environment is ensured by 7 joints, (b) torque-controlled joints that enable direct haptic interaction, (c) a robot weight of approx. 10 kg that allows for an easy handling of the system and reduces the potential risk of injury by collision due to low inertia, and (d) safety of the system by means of sensor redundancy.

The joints of the medical robot consist of motors and gears, link side torque and position sensors, as well as motor side position sensors and safety brakes. On the one hand this raises the system safety by the use of redundant sensors, on the other hand the sensor values are needed for the used control algorithms [52]. A compact and slender joint grouping was derived: Whilst the lower joint unit has three intersecting axes (roll-pitch-pitch), the other two joint units have two intersecting axes each (pitch-roll), see joints $q_{1..7}$[1] in Fig. 3.1. The intersecting axes in the joints contribute to a simplified robot control as the inverse kinematics of the robot arm has a closed form solution.

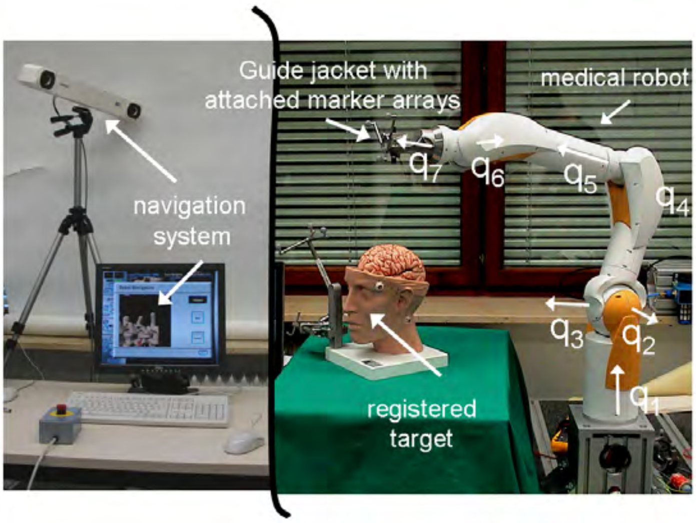


Figure 3.1.: The DLR medical robot KineMedic in a biopsy application [52]. Both the robot tool tip and the patient are tracked by a navigation system. The human cooperates with the robot while the guide jacket attached to the robot tip is positioned relative to the patient according to the preoperative planning. The biopsy needle is then inserted manually into the guide jacket, and the biopsy can be performed manually as well, while the robot is constantly observing the desired orientation of the biopsy needle.

In all robot joints a special motor developed by the DLR (DLR RoboDrive [62]) is used which was optimized for application in robotics with respect to its weight and electrical losses. In contrast to the established industrial robots, the power electronics of the motors are located directly in the robotic arm and not in an external control unit. This has advantages for the electromagnetic compatibility (EMC): the EMC-problematic cable currents of the motors are generated near the motors and no long transmission cables through the whole robot arm are necessary. The integrated power electronics

[1] In this thesis, the notation $q_{k...k+j}$ with $k, j \in \mathbb{N}$, $k \geq 0$, and $j \geq 2$ represents the joint angle set $\{q_k, q_{k+1}, ..., q_{k+n}\}$. Likewise, the notation $q_{k,k+1}$ denotes the joint angle set $\{q_k, q_{k+1}\}$.

in combination with the fieldorientated control allow an optimal use of the specialized motors.

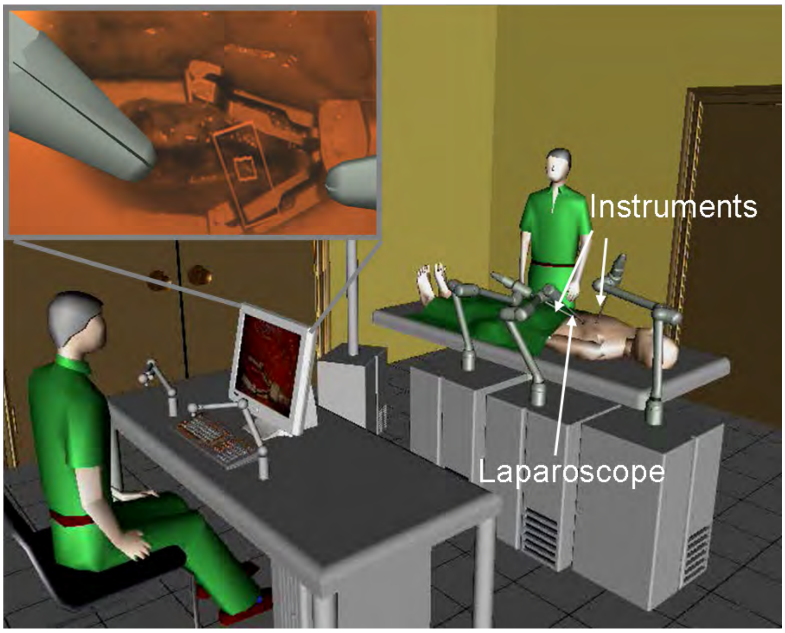


Figure 3.2.: In the DLR MIRS setup, one or several robots hold instruments and are used for manipulating, while an additional robot holds a stereoscopic laparoscope to provide vision to the operator.

3.1.2. Kinematic Structure

The DLR medical robot is designed for both open surgery (see e.g. Fig. 3.1 for a biopsy application) and minimally invasive surgeries as shown in Fig. 3.2.

Table 3.1.: DH parameters of the considered robot

i	a_{i-1} [mm]	α_{i-1}	d_i[mm]	θ_i
1	0	0°	0	q_1
2	0	-90°	0	$-90^\circ + q_2$
3	0	90°	0	q_3
4	310	-90°	0	$90^\circ + q_4$
5	0	90°	385	q_5
6	0	-90°	0	q_6
7	0	90°	364	$90^\circ + q_7$
8	0	-90°	0	$-90^\circ + q_8$
9	0	-90°	0	q_9

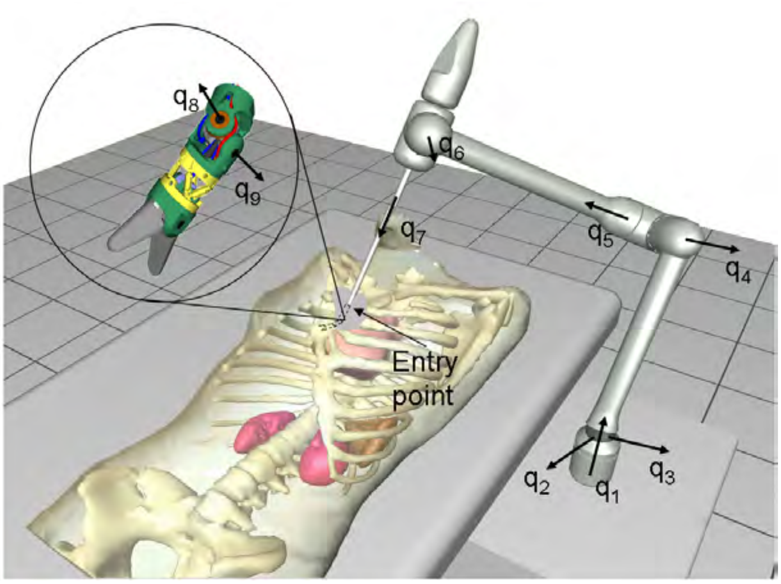


Figure 3.3.: The DLR medical robot in a minimally invasive application.

The kinematics in minimally invasive applications has the peculiarity of a fulcrum point where the surgical instrument enters into the human body (see Fig. 3.3). At that point, a constraint is imposed upon the system, resulting in a loss of two DoF. In order to regain full dexterity inside of the patient, an articulated instrument is used, adding two DoF (q_8 and q_9 in Fig. 3.3 and 3.4) to obtain full 6 Cartesian DoF (position and orientation) at the distal end of the instrument and thus allows for full manipulability inside of the patient. The robotic systems used in the DLR setup for MIRS therefore consist of the 7 DoF medical robot with an attached actuated instrument that offers two additional DoF. The anthropomorphic structure of the 7 DoF medical arm allows for a nullspace "elbow" motion outside of the patient with the position and orientation of the wrist frame $\{\mathbf{W}\}$ (see Fig. 3.4) imposed. The redundant DoF can thus be exploited to optimize a quality function that addresses e.g. manipulability, accuracy or to avoid singularities and collisions while the wrist frame position remains unchained. Alternatively, the surgical staff can position the robot inside the nullspace by haptic interaction. The kinematic structure of the considered robot with the attached actuated instrument and the used coordinate frames are shown in Fig. 3.4, the Denavit-Hartenberg (DH) parameters according to the convention of Craig [19] (see App. A) are given in Table 3.1.

3.2. Inverse Position Kinematics

Inverse kinematics provides methods to map the Cartesian space into the joint space of a kinematic chain. Inverse position kinematics searches for the joint angles $\mathbf{q}$ of the manipulator, given position and orientation $\mathbf{x} = \begin{pmatrix} x & y & z & \alpha & \beta & \gamma \end{pmatrix}^T$ of the robot tool

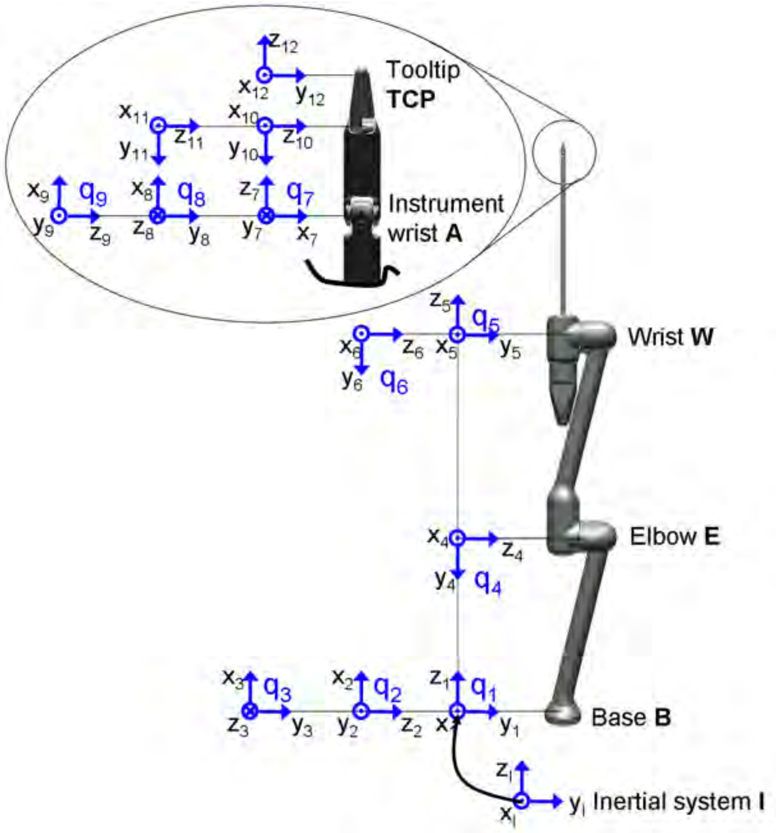


Figure 3.4.: Kinematic structure of the DLR medical robot with an actuated instrument.

center point (TCP) in the Cartesian space[2]. A manipulator will be called solvable if an algorithm exists that determines all possible sets of joint angles associated with a given TCP position and orientation [87]. In general, numerical methods do not provide solvability since these methods usually require an initial guess and provide a solution close to it, discarding all other solutions. This is sufficient for e.g. Cartesian control where one is only interested in a solution close to the previous step in time. In this case, numeric solutions may provide good solutions, and also methods based on linearization are very common. For linearization of a serial kinematic chain without redundancy, the inverse of the Jacobian can be used[3]:

$$\mathbf{q}_t = \mathbf{q}_{t-1} + \mathbf{J}^{-1} \left(\mathbf{x}_t - \mathbf{x}_{t-1} \right),$$

with $\mathbf{q}_{t,t-1}$ and $\mathbf{x}_{t,t-1}$ the joint position and Cartesian position at time t and $t-1$, respectively.

[2] The orientation is represented by the angles α, β and γ, using e.g. Euler or roll pitch yaw angles.

[3] Note that the Jacobian matrix depends on the chosen representation for the orientation α, β and γ.

In case of kinematic redundancy, the Jacobian matrix becomes non-square, and the pseudo-inverse Jacobian is used instead:

$$\mathbf{q}_t = \mathbf{q}_{t-1} + \mathbf{J}^{+}\left(\mathbf{x}_t - \mathbf{x}_{t-1}\right), \quad \text{with} \quad \mathbf{J}^{+} = \mathbf{J}^{\mathbf{T}}\left(\mathbf{J}\mathbf{J}^{\mathbf{T}}\right)^{-1}.$$

The pseudo-inverse Jacobian is obtained by minimizing the Euclidean norm of the joint velocities while maintaining a certain tool tip velocity $\dot{\mathbf{x}}$ [94].

In the scope of workplace optimization, an initial guess for the inverse kinematics can not be made in general. Therefore, numeric methods and methods based on integrating the inverse velocity kinematics either require many iterations or simply fail. The use of closed form solutions is very advantageous in this case, since these solutions are computationally very efficient and provide solvability independent from any initial guesses.

Calculation of inverse position kinematics in closed form involves solving a set of nonlinear equations. Therefore, no general algorithms are available. In many cases though it is possible to determine a closed form solution due to the special kinematic structure of a robot, relying on algebraic or geometric approaches. In the next section, the inverse kinematics of the actuated instrument of the DLR medical robot setup for MIS is calculated with a geometric approach, whereas the joint angles of the medical robot itself is calculated according to Pieper [84].

3.2.1. Closed Form Solution to the Inverse Position Kinematics of the DLR MIRS Setup

This section presents the calculation of the inverse position kinematics of the KineMedic in closed form. Note that in the following one joint of the KineMedic among $q_{1\ldots7}$ is chosen as nullspace parameter q_{null} and the remaining joint angles are grouped in the vector $\mathbf{q}_c \in \mathbb{R}^6$. The forward kinematics of the robotic system yields:

$$\mathbf{x} = h\left(\mathbf{q}_c, q_{\text{null}}, \mathbf{q}_{\text{instrument}}\right), \mathbf{q}_{\text{instrument}} \in \mathbb{R}^2, \tag{3.1}$$

with $\mathbf{q}_{\text{instrument}}$ a vector containing the joints $q_{8,9}$ of the instrument. To determine the optimal solution inside the nullspace, an optimization of the nullspace parameter q_{null} has to be included in the calculation as shown in Sect. 3.4. For determination of $\mathbf{q}_c$, Pieper's method [84] can be applied since three axes of the kinematic chain intersect in one point[4].

The problem can be broken down into the determination of (i) the inverse kinematics of the 2 DoF surgical instrument, used in a minimally invasive application (Sect. 3.2.1.1) and (ii) the general 7 DoF DLR medical arm (Sect. 3.2.1.2). Thereby the occurring systems of equation are solvable in closed form. Three alternative solutions for the computation of the inverse kinematics for the arm are given.

[4] The parameter q_{null} can be chosen arbitrarily among $q_{1,..3,5,..7}$. Only $q_{\text{null}} = q_4$ is not possible, see Eq. 3.8 on page 27 and the remark thereafter.

As follows, s_i denotes $\sin(q_i)$ and c_i denotes $\cos(q_i)$. A homogeneous transformation matrix is written as ${}^{i}_{j}\mathbf{T}$ and describes the position and orientation of the frame $\{j\}$ relative to the frame $\{i\}$ (see App. A for further details).

3.2.1.1. Determination of the Joint Angles $q_{8,9}$ of the Actuated Instrument

The joint angles of the instrument (q_8, q_9) are calculated with a geometric approach as follows: To calculate the joint angle q_9, the trocar point $\mathbf{t}$ is transformed into frame $\{9\}$ (see Fig. 3.4 and 3.5(right)):

$${}^{9}\mathbf{t} = \left({}^{I}_{TCP}\mathbf{T}\,{}^{TCP}_{9}\mathbf{T}\right)^{-1} \cdot {}^{I}\mathbf{t} = \begin{pmatrix} {}^{9}t_x \\ {}^{9}t_y \\ {}^{9}t_z \end{pmatrix}, \tag{3.2}$$

with ${}^{I}_{TCP}\mathbf{T}$ the desired tool tip frame written in the inertial coordinate system (COS) and ${}^{TCP}_{9}\mathbf{T}$ the frame $\{9\}$ written in the TCP COS. The frame ${}^{TCP}_{9}\mathbf{T}$ is known from the geometry of the instrument. The joint angle q_9 can then be calculated by projecting ${}^{9}\mathbf{t}$ into the x-y-plane (see Fig. 3.5(right)):

$$q_9 = -\text{arctan2}\left(-{}^{9}t_y, -{}^{9}t_x\right). \tag{3.3}$$

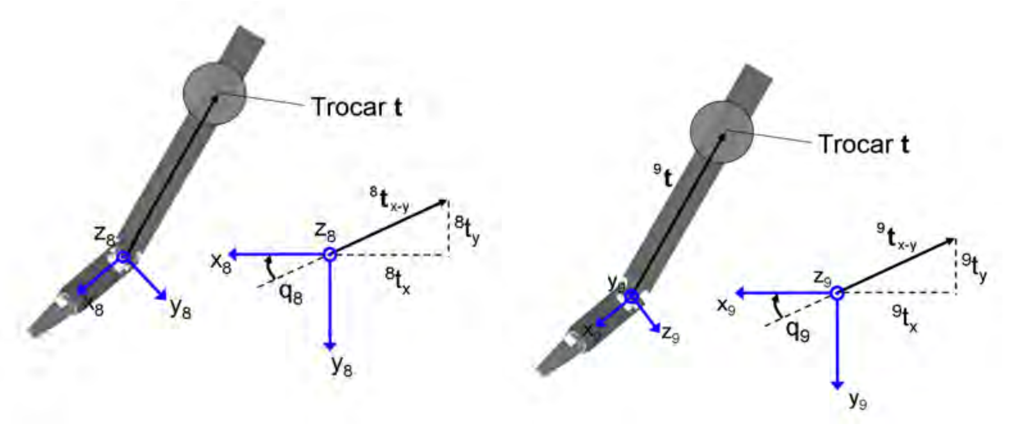


Figure 3.5.: Determination of the joint angles q_8 (left) and q_9 (right).

To determine the joint angle q_8, the trocar coordinates are transformed into frame $\{8\}$ (see Fig. 3.5(left)):

$${}^{8}\mathbf{t} = \left({}^{I}_{TCP}\mathbf{T}\,{}^{TCP}_{9}\mathbf{T}\,{}^{9}_{8}\mathbf{T}\right)^{-1} \cdot {}^{I}\mathbf{t} = \begin{pmatrix} {}^{8}t_x \\ {}^{8}t_y \\ {}^{8}t_z \end{pmatrix}. \tag{3.4}$$

The joint angle q_8 can then be calculated similar to (3.3):

$$q_8 = -\arctan2\left(-{}^8t_y, -{}^8t_x\right). \tag{3.5}$$

The solutions for q_8 and for q_9 are undefined in the case of ${}^9t_x = {}^9t_y = 0$ resp. ${}^8t_x = {}^8t_y = 0$. A singular configuration can be stated in these cases (see Sect. 3.3.1.2).

3.2.1.2. Determination of the Joint Angles $q_{1\ldots7}$ of the DLR Medical Arm

The rotation axes of the joints $q_{1\ldots3}$ as well as of the joints $q_{5\ldots7}$ intersect in one point. Thus, the inverse kinematics of the DLR medical arm can be solved in closed form in various ways (using Pieper's solution [84] to separate rotational DoF from translational DoF and thus simplify the equation system). Three solutions are given in the following. Naturally, the inverse kinematics of a redundant robot is solved (if solvable at all) by a manifold of joint angle sets. These sets span the nullspace. In the presented approaches the inverse kinematics is solved in closed form by presetting the nullspace parameter q_{null}. If this joint angle is defined as parameter of an optimization algorithm, an optimal joint set within the nullspace can be determined.

Solution 1 with $q_{\text{null}} = q_3$ resp. $q_{\text{null}} = q_2$

To determine the joint angles $q_{1,\ldots,4}$, the coordinates ${}^B_6\mathbf{p}$ of the frame $\{6\}$ in base coordinates are computed using the desired TCP frame (Eq. 3.6, where all components of ${}^B_6\mathbf{p}$ are known), and the coordinates ${}^B_5\mathbf{p}$ of the frame $\{5\}$ in base coordinates are computed using the forward kinematics (Eq. 3.7, where only three unknown joint angles appear). Since ${}^B_5\mathbf{p} = {}^B_6\mathbf{p}$, the joint angles $q_{1,\ldots,4}$ can then be calculated (Eq. 3.8–3.10 with $q_{\text{null}} = q_3$ and Eq. 3.8, 3.10, and 3.11 with $q_{\text{null}} = q_2$). The coordinates ${}^B_6\mathbf{p}$ yield

$$\begin{pmatrix} {}^B_6\mathbf{p} \\ 1 \end{pmatrix} = {}^I_B\mathbf{T}^{-1}\, {}^{\;\;I}_{TCP}\mathbf{T} \left({}^{\;\;6}_{TCP}\mathbf{T}\right)^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \tag{3.6}$$

with ${}^I_B\mathbf{T}$ the base frame and ${}^{\;\;I}_{TCP}\mathbf{T}$ the desired TCP frame (both relative to the inertial frame $\{\mathbf{I}\}$) and ${}^{\;\;6}_{TCP}\mathbf{T}$ the TCP frame relative to frame $\{6\}$. Note that the joint angle q_7 that appears in ${}^{\;\;6}_{TCP}\mathbf{T}$ can be chosen arbitrarily since it has no influence on the coordinates of ${}^B_6\mathbf{p}$.

The coordinates ${}^B_5\mathbf{p}$ are computed as a function of $q_{1,\ldots,4}$ using forward kinematics as follows:

$$\begin{pmatrix} {}^B_5p_x \\ {}^B_5p_y \\ {}^B_5p_z \end{pmatrix} = \begin{pmatrix} c_1k_2 - s_1s_3k_1 \\ s_1k_2 + c_1s_3k_1 \\ c_2c_3k_1 - d_5s_2s_4 \end{pmatrix}, \tag{3.7}$$

$$\text{with } k_1 = (d_5 c_4 + a_3), \quad \text{and} \quad k_2 = s_2 c_3 k_1 + d_5 c_2 s_4.$$

Equating ${}^{B}_{5}\mathbf{p} = {}^{B}_{6}\mathbf{p}$, the joint angles $q_{1,2,4}$ can be calculated with $q_{\text{null}} = q_3$:

$$q_4 = \pm \arccos\left(\frac{r^2 - \left({a_3}^2 + {d_5}^2\right)}{2\, a_3 d_5}\right), \tag{3.8}$$

$$\text{with} \quad r^2 = {}^{B}_{5}{p_x}^2 + {}^{B}_{5}{p_y}^2 + {}^{B}_{5}{p_z}^2\,,$$

$$q_2 = 2 \,\arctan\left(\frac{k_3 \pm \sqrt{k_3^2 + k_4^2 - k_5^2}}{k_4 + k_5}\right), \tag{3.9}$$

with

$$k_3 = -d_5 s_4, \quad k_4 = c_3\left(d_5 c_4 + a_3\right),$$

$$k_5 = {}^{B}_{5}p_z\,,$$

and

$$q_1 = \text{arctan2}\left({}^{B}_{5}p_y, {}^{B}_{5}p_x\right) - \text{arctan2}\left(k_2, s_3 k_1\right). \tag{3.10}$$

It is remarkable that the elbow joint angle q_4 is not dependent on the preset joint angle q_3. Thus, angle q_4 remains constant during any nullspace motion and can therefore not be chosen as nullspace parameter q_{null}.

The solution for the joint angles $q_{1,2,4}$ exists if (see Eq. 3.8)

$$\left|\frac{r^2 - \left({a_3}^2 + {d_5}^2\right)}{2\, a_3 d_5}\right| \leq 1$$

and if (due to the root in Eq. 3.9)

$$k_3^2 + k_4^2 - k_5^2 \geq 0\,.$$

Alternatively, the joint angle q_2 can be chosen as nullspace parameter q_{null}. In this case the joint angle q_4 can again be computed according to Eq. 3.8, and the joint angle q_3 yields:

$$q_3 = \pm \arccos\left(\frac{{}^{B}_{5}p_z + d_5 s_2 s_4}{c_2\left(d_5 c_4 + a_3\right)}\right). \tag{3.11}$$

The joint angle q_1 is calculated with Eq. 3.10. The solution for the joint angle q_3 exists if

$$\left| \frac{{}^{B}_{5}p_z + d_5 s_2 s_4}{c_2 \left(d_5 c_4 + a_3\right)} \right| \leq 1.$$

Note that both Eq. 3.9 and 3.11 may become unsolvable due to a bad choice for the joint angle q_{null}.

Eventually, the joint angles $q_{5\ldots7}$ have to be determined. Therefore, the orientation of frame {7} with respect to frame {4} is computed using (i) the desired TCP frame (Eq. 3.12, with all components of the matrix ${}^{4}_{7}\mathbf{R}$ known) and (ii) the forward kinematics (Eq. 3.13, where only three unknown joint angles appear), and then equated.

With ${}^{4}_{I}\mathbf{R}$ the orientation of the inertial frame relative to frame {4} (computed by evaluation of the joint angles $q_{1\ldots4}$), the frame ${}^{4}_{7}\mathbf{R}$ yields:

$${}^{4}_{7}\mathbf{R} = {}^{4}_{I}\mathbf{R}\,{}^{I}_{TCP}\mathbf{R}\,{}^{TCP}_{7}\mathbf{R} = \begin{pmatrix} {}^{4}_{7}r_{11} & {}^{4}_{7}r_{12} & {}^{4}_{7}r_{13} \\ {}^{4}_{7}r_{21} & {}^{4}_{7}r_{22} & {}^{4}_{7}r_{23} \\ {}^{4}_{7}r_{31} & {}^{4}_{7}r_{32} & {}^{4}_{7}r_{33} \end{pmatrix}. \tag{3.12}$$

On the other hand, ${}^{4}_{7}\mathbf{R}$ can be written as a function of $q_{5\ldots7}$:

$${}^{4}_{7}\mathbf{R} = \begin{pmatrix} -c_5c_6s_7 - s_5c_7 & -c_5c_6c_7 + s_5s_7 & c_5s_6 \\ -s_6s_7 & -s_6c_7 & -c_6 \\ -s_5c_6s_7 + c_5c_7 & -s_5c_6c_7 - c_5s_7 & s_5s_6 \end{pmatrix}. \tag{3.13}$$

Equating Eq. 3.12 and Eq. 3.13, the joint angles $q_{5\ldots7}$ can be computed:

$$q_6 = \operatorname{arctan2}\left(\pm\sqrt{{}^{4}_{7}r_{21}{}^{2} + {}^{4}_{7}r_{22}{}^{2}}, -{}^{4}_{7}r_{23},\right), \tag{3.14}$$

$$q_5 = \operatorname{arctan2}\left(\frac{{}^{4}_{7}r_{33}}{s_6}, \frac{{}^{4}_{7}r_{13}}{s_6}\right),$$

and

$$q_7 = \operatorname{arctan2}\left(-\frac{{}^{4}_{7}r_{21}}{s_6}, -\frac{{}^{4}_{7}r_{22}}{s_6}\right).$$

In case

$$s_6 = 0 \rightarrow q_6 = k\pi, k \in \mathbb{N},$$

the joint angles $q_{5,7}$have identical rotation axes and can be chosen arbitrarily according to

$$q_5 + q_7 = \text{arccos}\left(r_{31}\right).$$

Altogether 8 solution sets with individual nullspace each are obtained due to the plus-minus signs in Eq. 3.8, Eq. 3.9 and Eq. 3.14.

Solution 2 with joint $q_{\textbf{null}} = q_7$

In case $q_{\text{null}} = q_7$, a solution to the inverse kinematics in line with the above shown method can be developed. First the joint angles $q_{4..6}$ of the robotic arm are calculated by equating the coordinates ${}^6_B\mathbf{p}$ using (i) the desired TCP frame (Eq. 3.15) and (ii) the forward kinematics (Eq. 3.16). The coordinates ${}^6_B\mathbf{p}$ yield with (i):

$$\begin{pmatrix} {}^6_B\mathbf{p} \\ 1 \end{pmatrix} = \left({}^B_I\mathbf{T}\,{}^I_{TCP}\mathbf{T}\,{}^{TCP}_{6}\mathbf{T}\right)^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{3.15}$$

On the other hand, one can calculate ${}^6_B\mathbf{p}$ as a function of the joint angles $q_{4,\ldots,6}$:

$$\begin{pmatrix} {}^6_Bp_x \\ {}^6_Bp_y \\ {}^6_Bp_z \end{pmatrix} = \begin{pmatrix} a_3c_6c_5s_4 - s_6\left(-a_3c_4 - d_5\right) \\ a_3c_6c_5s_4 - s_6\left(-a_3c_4 - d_5\right) \\ a_3s_5s_4 \end{pmatrix}. \tag{3.16}$$

Equating Eq. 3.15 and 3.16, the joint angles $q_{4..6}$ can be computed as follows:

$$q_4 = \pm\arccos\left(\frac{r - \left({a_3}^2 + {d_5}^2\right)}{2\,a_3d_5}\right) \tag{3.17}$$

$$\text{with} \quad r^2 = {}^6_B{p_x}^2 + {}^6_B{p_y}^2 + {}^6_B{p_z}^2\,,$$

$$q_5 = \begin{cases} \arcsin\left(\frac{{}^6_Bp_z}{a_3s_4}\right) \\ \pi - \arcsin\left(\frac{{}^6_Bp_z}{a_3s_4}\right) \end{cases}, \tag{3.18}$$

and

$$q_6 = \operatorname{arctan2}\left({}^6_Bp_y, {}^6_Bp_x\right) - \operatorname{arctan2}\left(-a_3c_4 - d_5, a_3s_4c_5\right). \tag{3.19}$$

The solution for the joint angles $q_{4,5,6}$ exists if (due to Eq. 3.17):

$$\left|\frac{r - \left({a_3}^2 + {d_5}^2\right)}{2\,a_3d_5}\right| \leq 1$$

and (due to Eq. 3.18):

$$\left|\frac{{}^6_Bp_z}{a_3s_4}\right| \leq 1\,.$$

Eventually, the joint angles $q_{1..3}$ have to be determined. Therefore, the rotation matrix of the base frame relative to the frame {3} is computed using (i) the known rotation matrices ${}^{I}_{B}\mathbf{R}$, ${}^{I}_{6}\mathbf{R}$, and ${}^{6}_{3}\mathbf{R}$ (Eq. 3.20) and (ii) the forward kinematics (Eq. 3.21), and then equated. With (i), the rotation matrix ${}^{3}_{B}\mathbf{R}$ yields:

$$
{}^{3}_{B}\mathbf{R} = \left({}^{I}_{B}\mathbf{R}^{-1}\,{}^{I}_{6}\mathbf{R}\,{}^{6}_{3}\mathbf{R}\right)^{-1} = \begin{pmatrix} {}^{3}_{B}r_{11} & {}^{3}_{B}r_{12} & {}^{3}_{B}r_{13} \\ {}^{3}_{B}r_{21} & {}^{3}_{B}r_{22} & {}^{3}_{B}r_{23} \\ {}^{3}_{B}r_{31} & {}^{3}_{B}r_{32} & {}^{3}_{B}r_{33} \end{pmatrix}. \tag{3.20}
$$

On the other hand, ${}^{3}_{B}\mathbf{R}$ can be written as a function of $q_{1,...,3}$ as:

$$
{}^{3}_{B}\mathbf{R} = \begin{pmatrix} -c_3s_2c_1 - s_3s_1 & c_3s_2s_1 - s_3c_1 & c_3c_2 \\ -s_3s_2c_1 + c_3s_1 & s_3s_2s_1 + c_3c_1 & s_3c_2 \\ -c_2c_1 & c_2s_1 & -s_2 \end{pmatrix}, \tag{3.21}
$$

and the joint angles $q_{1..3}$ can be computed:

$$
q_2 = \text{arctan2}\left(-{}^{3}_{I}r_{33}, \pm\sqrt{{}^{3}_{I}r_{31}{}^{2} + {}^{3}_{I}r_{32}{}^{2}}\right), \tag{3.22}
$$

$$
q_3 = \text{arctan2}\left(\frac{{}^{3}_{I}r_{23}}{c_2}, \frac{{}^{3}_{I}r_{13}}{c_2}\right), \tag{3.23}
$$

and

$$
q_1 = \text{arctan2}\left(\frac{{}^{3}_{I}r_{32}}{c_2}, \frac{{}^{3}_{I}r_{31}}{c_2}\right). \tag{3.24}
$$

In case

$$
c_2 = 0 \rightarrow q_2 = \frac{\pi}{2} + k\pi, k \in \mathbb{N},
$$

the joint angles $q_{1,3}$have identical rotation axes and can be chosen arbitrarily according to

$$
q_1 - q_3 = \arccos(r_{22}).
$$

The complete nullspace can be calculated by varying the nullspace parameter q_{null}. Every solution with $q_{\text{null}} = q_2$ is connected to a solution with $q_{\text{null}} = q_3$ resp. $q_{\text{null}} = q_7$ through the nullspace, so that all solutions obtained with $q_{\text{null}} = q_2$ are identical to the solutions obtained with $q_{\text{null}} = q_3$ resp. $q_{\text{null}} = q_7$. This is due to the fact that the parametrization (corresponding to the choice of the nullspace parameter q_{null}) has no influence on the obtained solutions. Other parametrizations are possible like the definition of an *arm angle* [96]. The advantage of the *arm angle* method is that all joint angles can be expressed as a function of the *arm angle*, independent from the other joint angles. Thus for some special kinematic structures it is e.g. possible to compute a solution in closed form inside the nullspace that observes the joint limits.

Figure 3.6 shows all possible configurations of the robot to reach a certain tool tip position and orientation through a given trocar with $q_7 = -3.92°$.

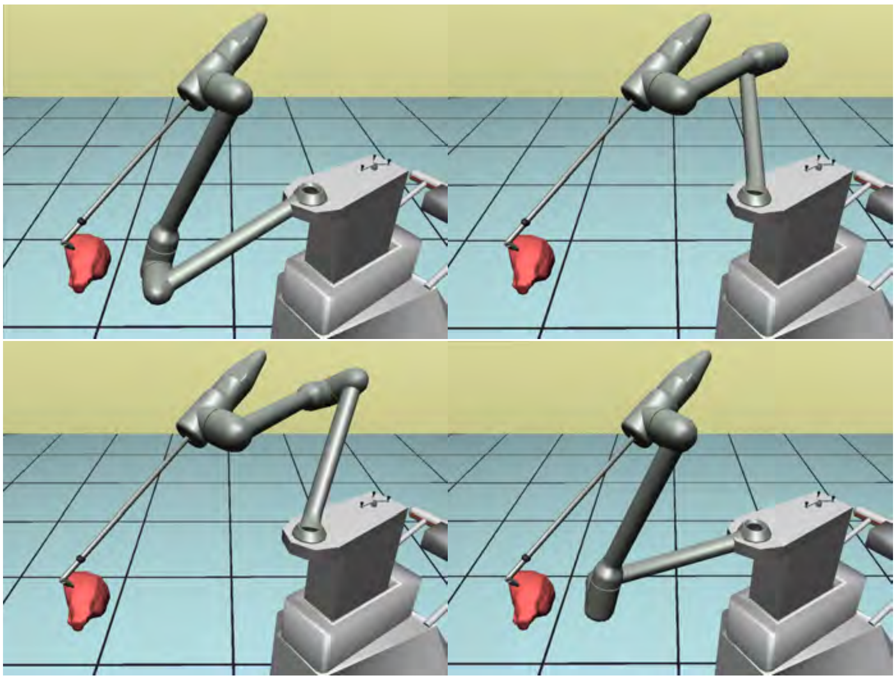


Figure 3.6.: The 8 possible robot configurations for a given tool tip position and orientation and a given entry point. Due to the intersection of the three consecutive axes $q_{5..7}$, each two configurations look similar and are not shown separately in the figure.

3.3. Singularities

In a singular configuration, the robot Jacobian[5] looses rank, i.e. for square Jacobians, the determinant $|\mathbf{J}| = 0$. For redundant manipulators with $\mathbf{J}$ being a non-square matrix, a loss of rank can be stated if $\left|\mathbf{J}\mathbf{J}^{\mathrm{T}}\right| = 0$. A manipulator with less degrees of freedom than the dimension of space it is supposed to move in is always singular.

The preoperative determination of the singular configurations of a robot is especially important in the case of teleoperation, where the exact path is not known in advance. Though singular configurations can be detected by monitoring certain manipulability measures as described e.g. in [54, 111], these measures are to the author's knowledge

[5] The robot Jacobian $\mathbf{J}$ relates the Cartesian space with the joint space on velocity level: $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$.

insufficient to signal vicinity of singular configurations. Since in most cases the behavior of robots near singularities is not very intuitive for the operator, it is highly desirable to restrict the workspace admissible to the operator to a space that does not contain any singularities or to control the robot in a way that singular configurations are avoided. This is facilitated if a closed form description of all singularities of the robot design is known, since in this case it becomes possible to use computationally cheap strategies for singularity avoidance in analogy with well known strategies for joint limit avoidance. Below, all singularities of the DLR medical robot are determined in closed form according to [50].

3.3.1. Singularities of the DLR Medical Robot in Minimally Invasive Applications

The design of robotic devices for medical applications is liable to exceptionally high requirements in terms of safety and reliability. A thorough analysis of the kinematic structure of the robot is important to ensure complete reachability as well as the absence of any singular configuration inside the desired workspace. The desired workspace is usually defined by the operator during a planning step, and serves to determine the optimal robot setup [1, 56]. In minimally invasive applications, the robot setup comprises the position and orientation of the robot base and the position of the entry point into the human body as well as any adjustable DH parameter (as for example adjustable instrument lengths). A closed form determination of all singularities is an important prerequisite for algorithms that identify configurations assuring the absence of singular configurations throughout a given workspace.

The determination in closed form of the singularities of the DLR medical arm and the attached surgical instrument is given in Sect. 3.3.1.1 and 3.3.1.2, respectively.

3.3.1.1. The Singular Configurations of the DLR Medical Arm

Written in the wrist frame $\{\mathbf{W}\}$ as shown in Fig. 3.4, the Jacobian $\mathbf{J}$ of the forward kinematics has the following form [111]:

$$\begin{pmatrix} \mathbf{v}_W \\ \omega_W \end{pmatrix} = \mathbf{J}\dot{\mathbf{q}} = \begin{pmatrix} \mathbf{J}_{11} & \mathbf{0} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{pmatrix} \dot{\mathbf{q}}, \tag{3.25}$$

with $\begin{pmatrix} \mathbf{v}_W \\ \omega_W \end{pmatrix}$ being the translational and rotational velocity of the wrist frame $\{\mathbf{W}\}$ and

$$\mathbf{J}_{11},\ \mathbf{J}_{21} \in \mathbb{R}^{3\times 4},\ \mathbf{J}_{22} \in \mathbb{R}^{3\times 3}\ . \tag{3.26}$$

A singular configuration occurs if the following determinant equals zero:

$$\left|\mathbf{J}\mathbf{J}^{\mathbf{T}}\right| = 0. \tag{3.27}$$

With the formula of Cauchy-Binet (see e.g. [32]), Eq. 3.27 can be transformed into a sum of squares of determinants:

$$\left|\mathbf{J}\mathbf{J}^{\mathbf{T}}\right| = \sum_{i=1}^{4} \left| \begin{pmatrix} \mathbf{J}_{11}^{i} & \mathbf{0} \\ \mathbf{J}_{21}^{i} & \mathbf{J}_{22} \end{pmatrix} \right|^2 + \sum_{i=1}^{3} \left| \begin{pmatrix} \mathbf{J}_{11} & \mathbf{0} \\ \mathbf{J}_{21} & \mathbf{J}_{22}^{i} \end{pmatrix} \right|^2 , \tag{3.28}$$

with $\mathbf{J}_{mn}^{i}$ the i-th submatrix (minor) obtained by suppressing column i of the matrix $\mathbf{J}_{mn}$. The terms of the first sum have a lower block triangular form and can be combined to:

$$\sum_{i=1}^{4} \left| \begin{pmatrix} \mathbf{J}_{11}^{i} & \mathbf{0} \\ \mathbf{J}_{21}^{i} & \mathbf{J}_{22} \end{pmatrix} \right|^2 = |\mathbf{J}_{22}|^2 \sum_{i=1}^{4} \left|\mathbf{J}_{11}^{i}\right|^2 = |\mathbf{J}_{22}|^2 \left|\mathbf{J}_{11}\mathbf{J}_{11}^{\mathbf{T}}\right| . \tag{3.29}$$

In the last step, the formula of Cauchy-Binet is applied inversely. Since the sum in Eq. 3.28 consists of squared summands, all of them have to equal zero in a singular configuration.

The following two simplifications are possible with consideration of the rank of the Jacobian:

1. Due to the special structure of $\mathbf{J}$, a sufficient condition for a singular configuration is:
$$\operatorname{rank}(\mathbf{J}_{11}) < 3\,. \tag{3.30}$$
2. For the remaining singular configurations (i.e., $\operatorname{rank}(\mathbf{J}_{11}) = 3$), a necessary condition is (due to Eq. 3.29):
$$\operatorname{rank}(\mathbf{J}_{22}) < 3\,. \tag{3.31}$$

Thus, the second sum of Eq. 3.28 has to be evaluated only for joint angles that cause $|\mathbf{J}_{22}|$ to be zero. The following singularities e_i can thus be determined, with $k \in \mathbb{N}$:

$$e_1: \quad q_4 = \pi k, \tag{3.32}$$

$$e_2: \quad q_2 = \frac{\pi}{2} + \pi k \wedge q_3 = \frac{\pi}{2} + \pi k, \tag{3.33}$$

$$e_3: \quad q_2 = \frac{\pi}{2} + \pi k \wedge q_4 = \pm \arccos\left(-\frac{a_3}{d_5}\right) + 2\pi k, \tag{3.34}$$

$$e_4: \quad q_2 = \frac{\pi}{2} + \pi k \wedge q_6 = \pi k, \text{ and} \tag{3.35}$$

$$e_5: \quad q_5 = \frac{\pi}{2} + \pi k \wedge q_6 = \pi k. \tag{3.36}$$

The singular configuration e_3 only appears if $\|a_3\| \leq \|d_5\|$. Details about the roots of the relevant determinants are given in App. B. As for the DLR medical robot, all singularities except for

$$e_1 : q_4 = \pi k \quad \text{and} \quad e_5 : q_5 = \frac{\pi}{2} + \pi k \wedge q_6 = \pi k$$

are outside of the joint limits.

The classical "wrist singularity" ($q_6 = \pi k$) that occurs in many 6 DoF kinematic chains (consider for example a kinematic chain obtained with joint q_3 held constant) only appears in conjunction with additional conditions (singularities $e_{4,5}$). To illustrate this, the pseudo-inverse $\mathbf{J}_a^+$ of the Jacobian $\mathbf{J}_a$ in the non-singular configuration $\mathbf{q}_a = (0, 0, 0, \pi/2, 0, 0, 0)^{\mathrm{T}}$ as shown left in Fig. 3.7 is considered. In the vicinity of a singular configuration, certain Cartesian velocity vectors cause exceedingly high joint velocities if transformed into joint space by the pseudo-inverse of the Jacobian. Written in frame $\{\mathbf{I}\}$, the pseudo-inverse $\mathbf{J}_a^+$ yields:

$$\mathbf{J}_a^+ = \mathbf{J}_a^{\mathbf{T}} \left(\mathbf{J}_a \mathbf{J}_a^{\mathbf{T}} \right)^{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{a_3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{a_3} & 0 & 0 & 0 & -\frac{d_5}{a_3} \\ -\frac{1}{a_3} & 0 & -\frac{1}{d_5} & 0 & 0 & 0 \\ 0 & \frac{1}{2a_3} & 0 & \frac{1}{2} & 0 & -\frac{d_5}{2a_3} \\ 0 & 0 & \frac{1}{d_5} & 0 & 1 & 0 \\ 0 & \frac{1}{2a_3} & 0 & \frac{1}{2} & 0 & -\frac{d_5}{2a_3} \end{pmatrix}. \tag{3.37}$$

With $d_5/a_3 \approx 1$, all joint velocities remain small for arbitrary rotations of the tool tip. Particularly, pure rotation around an axis b as shown left in Fig. 3.7 (perpendicular to the rotation axes of q_6 and q_7 and intersecting them), constituting the singular direction in case of a kinematic chain with joint q_3 held constant, leads to the following (reasonably small) joint velocities:

$$\dot{\mathbf{q}} = \mathbf{J}_a^+ \cdot (0, 0, 0, 0, 0, 1)^{\mathrm{T}} = (1, 0, -\frac{d_5}{a_3}, 0, -\frac{d_5}{2a_3}, 0, -\frac{d_5}{2a_3})^{\mathrm{T}}. \tag{3.38}$$

On the right of Fig. 3.7 the norm $||\mathbf{J}_{a'}^+(q_5) \cdot (0, 0, 0, 0, 0, 1)^{\mathbf{T}}||_2$ is shown as a function of the joint angle q_5 with all other angles remaining in configuration $\mathbf{q}_a$:

$$\mathbf{q}_{a'} = (0, 0, 0, \pi/2, q_5, 0, 0)^{\mathrm{T}}.$$

A singular configuration only occurs with joint $q_5 = \pi/2 + \pi k$ as determined above.

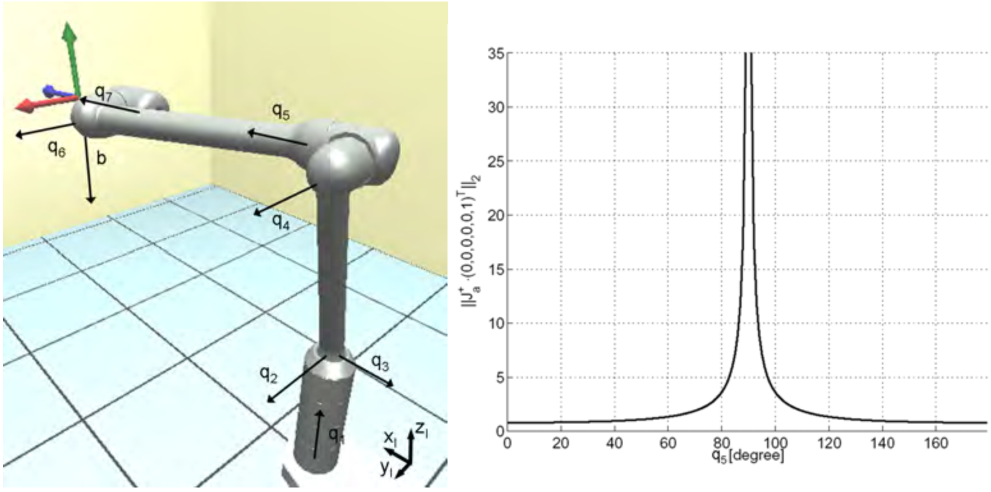


Figure 3.7.: In case of the considered robot, pure rotations around the axis b can still be performed even if $q_6 = 0$ (left). Only if also $q_5 = \pi/2 + \pi k$, a singular configuration occurs, see the norm $||\mathbf{J}_{a'}^{+}(q_5)\cdot(0,0,0,0,0,1)^{\mathbf{T}}||_2$ as a function of the joint angle $q_5$on the right.

Generalization to the Case of a Serial Robot with N-Fold Redundancy

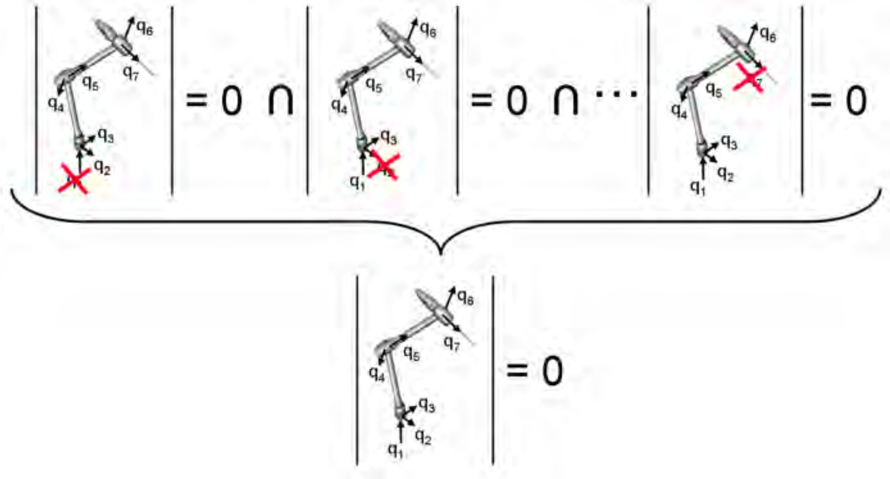


Figure 3.8.: The singularities of a serial redundant structure with $m+n$ joints of which n are redundant are identical with the intersection of the singularities of all those robotic structures obtained by fixing any possible set of n joints of the redundant structure. The figure shows the case $n = 1$.

The singular configurations of a general, n-fold redundant robot can be calculated by considering the roots of the following determinant:

$$\left|\mathbf{J}\mathbf{J}^{\mathbf{T}}\right| = \sum_{i=1}^{\frac{(m+n)!}{(n!)(m!)}} \left|\mathbf{J}_i\right|^2, \quad \mathbf{J} \in \mathbb{R}^{m\times(m+n)}, \tag{3.39}$$

with $\mathbf{J}_i$ representing all $\frac{(m+n)!}{(n!)(m!)}$ (different) matrices obtainable by suppressing n columns of the Jacobian $\mathbf{J}$. It can be seen from Eq. 3.39 that the singularities of a serial redundant structure with $m+n$ joints of which n are redundant are identical with the intersection of the singularities of all those robotic structures obtained by fixing any possible set of n joints of the redundant structure (see Fig. 3.8). It has to be noted, however, that already for the case of a 2-fold redundant robot with 8 DoF, $\frac{8!}{2!\cdot 6!} = 28$ minors have to be considered, each of which being usually a rather complex function of the joint angles $\mathbf{q}$.

3.3.1.2. Singularities of the Instrument in a Minimally Invasive Application

To analyze the singular configurations introduced by the fulcrum point and the two extra DoF of the instrument (see Fig. 3.9), the following Jacobian matrix is considered:

$$\begin{pmatrix} \mathbf{v}_9 \\ \omega_9 \end{pmatrix} = {}^6_9\mathbf{J}_v \begin{pmatrix} \dot{x}_6 \\ \dot{y}_6 \\ \dot{z}_6 \\ \dot{q}_7 \\ \dot{q}_8 \\ \dot{q}_9 \end{pmatrix}, \; {}^6_9\mathbf{J}_v = \begin{pmatrix} \frac{d_7'-d_7}{d_7'} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{d_7'-d_7}{d_7'} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{d_7'} & 0 & -c_7 & -s_7c_8 \\ 0 & 0 & 0 & -1 & 0 & s_8 \\ -\frac{1}{d_7'} & 0 & 0 & 0 & -s_7 & c_7c_8 \end{pmatrix}, \tag{3.40}$$

with $\mathbf{v}_9$ resp. ω_9 the translational and rotational velocity of frame $\{\mathbf{9}\}$ and $(\dot{x}_6,\dot{y}_6,\dot{z}_6)$ the translational velocities of frame $\{\mathbf{W}\}$. The determinant of ${}^6_9\mathbf{J}_v$ yields:

$$\left|{}^6_9\mathbf{J}_v\right| = -\frac{(d_7' - d_7)^2\, c_8}{d_7'^2}, \tag{3.41}$$

and a singular configuration can be stated when

$$c_8 = 0 \rightarrow q_8 = \frac{\pi}{2} + k\pi, k \in \mathbb{N}, \tag{3.42}$$

with the joint axes of q_7 and q_9 aligned. The singular configuration that occurs if

$$d_7' = d_7 \tag{3.43}$$

corresponds to a configuration where the fulcrum point is coincident with the origin of the instrument wrist frame $\{\mathbf{A}\}$. In this case, translations of the frame $\{\mathbf{A}\}$ are partly restricted by the constraint of the fulcrum point.

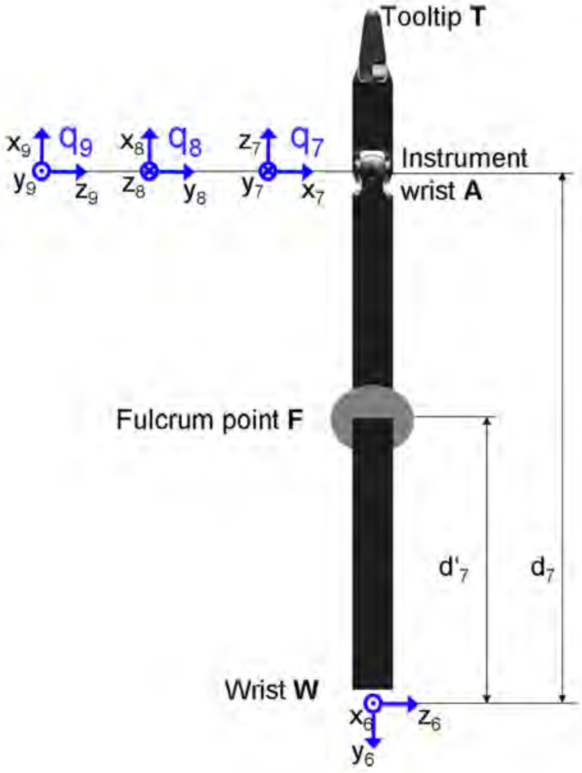


Figure 3.9.: Kinematic description of the articulated instrument in MIRS. The length $d_7^{'}$ signifies the distance between the wrist frame $\{\mathbf{W}\}$ and the fulcrum point.

3.3.1.3. Singularities and the Closed Form Solution to Inverse Kinematics for Redundant Structures

Section 3.2.1 shows different ways to calculate the inverse kinematics of a redundant manipulator in closed form. The basic idea is to treat the robot as a 6 DoF kinematic chain by considering some joints to be held constant (e.g. the nullspace parameter q_{null}). However, if the robot is reduced to a 6 DoF kinematic chain, the closed form solution also suffers from all the singularities of the respective 6 DoF kinematics. It is therefore important to choose the appropriate closed form inverse kinematics algorithm that, according to the desired Cartesian position or the initial joint position, is far from any singular configurations.

3.4. A New Concept for Calculating the Inverse Position Kinematics of Redundant Serial Manipulators

With the knowledge of singular configurations both in joint space and in Cartesian space (using forward kinematics), it is possible to reliably calculate parts of the inverse kinematics in closed form while optimizing the remaining redundant DoF. The dimension of the optimization problem and therefore the calculation time can thus be reduced considerably. Furthermore, the desired TCP position and orientation is reached even if the optimization problem is stuck in a local minimum (as opposed to if the TCP position is included in the algorithm as an optimization criterion).

For the DLR medical robot KineMedic, the proposed algorithm is as follows:

1. According to the initial joint angles, choose the closed form solution algorithm that has its singularities farthest away from the initial joint angles in terms of the Euclidean norm. The remaining joint angle q_{null} is subject to optimization.
2. Optimize q_{null} . Whenever a new q_{null} is evaluated, all joint angles have to be updated using the closed form inverse kinematics solution.

Calculation results of this method are shown and compared to state of the art methods in Chapter 5.

3.5. Summary and Conclusion

In the previous sections, stress is laid on closed form solutions for both the inverse position kinematics and the determination of singularities. Compared to iterative methods, closed form solutions require a thorough and challenging investigation of the robot kinematics, whereas iterative methods often allow for treating the robot kinematics as a "black box". Closed form solutions, however, offer the following advantages:

- Closed form inverse position kinematics
    - provides solvability, i.e. all possible solutions are obtained and
    - is computationally faster, especially if no initial solutions are available.
- Closed form determination of the singularities
    - allows for very easy and computationally fast checks of the current robot configuration with respect to singularities (it is not necessary to check the condition of the Jacobian) and
    - allows for algorithms that optimize a robotic setup so that no collisions occur within a given workspace.

For the DLR medical robot KineMedic, closed form solutions of both the inverse position kinematics and the determination of singularities can be given. For the calculation of the singularities of redundant kinematic structures, the use of the formula of Cauchy-Binet simplifies the equations considerably. The singularities of the KineMedic robot inside the joint limits yield:

$$e_1 : q_4 = \pi k \quad \text{and} \quad e_5 : q_5 = \frac{\pi}{2} + \pi k \wedge q_6 = \pi k \,.$$

The shown mathematic approaches can be applied to all other DLR 7 DoF robots shown in Fig. 3.10 as well.

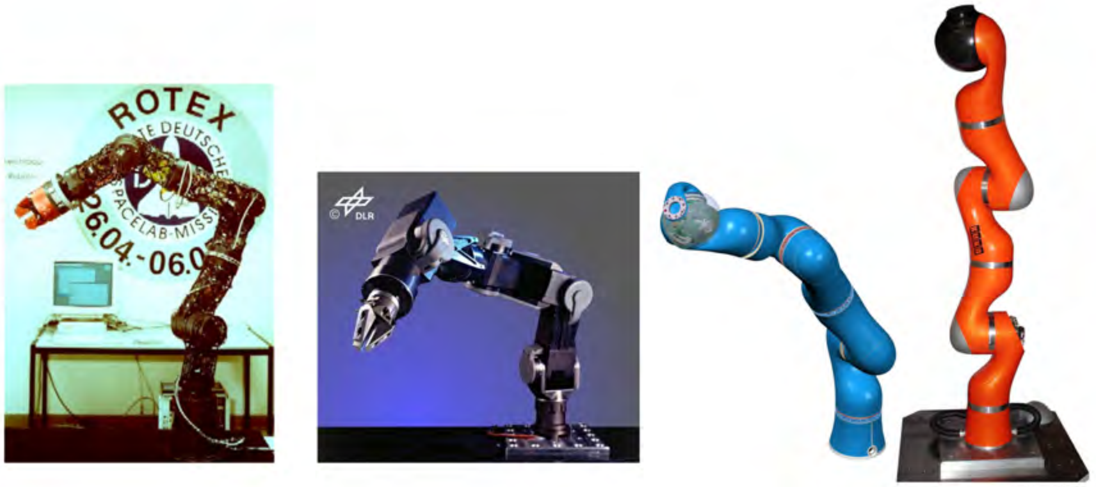


Figure 3.10.: The DLR Light-Weight Robot (LWR) generations (from left): LWR I (1991), LWR II (1998), LWR III (2002) and the commercialized version of the LWR III (2006).

An important issue when dealing with real robotic systems is the discrepancy between the assumption of certain characteristics in the DH parameters (e.g. ideally intersecting, perpendicular axes) and the slight variation in the actual implementation of the robotic system (due to e.g. manufacturing tolerances). A feasible approach to compensate for these variations is e.g. to perform a linearization as follows:

It is assumed that a forward kinematics function h_{ideal} based on the ideal DH parameters with a closed form solution exists. Furthermore, the (slightly varying) forward kinematics function h_{real} is acquired, e.g. by calibration of the robot.

With $\mathbf{x}_{\text{desired}}$ the desired TCP position and orientation, the joint angles $\mathbf{q}_{\text{ideal}}$ are calculated based on the closed form solution of:

$$\mathbf{x}_{\text{desired}} = h_{\text{ideal}}\left(\mathbf{q}_{\text{ideal}}\right).$$

By inserting $\mathbf{q}_{\text{ideal}}$ into h_{real}, a slightly varied TCP position and orientation $\mathbf{x}_{\text{real}}$,

$$\mathbf{x}_{\text{real}} = h_{\text{real}}\left(\mathbf{q}_{\text{ideal}}\right)$$

can be computed, and an error in TCP position and orientation $\Delta\mathbf{x}$ can be determined:

$$\Delta\mathbf{x} = \mathbf{x}_{\text{real}} - \mathbf{x}_{\text{desired}}.$$

By negatively compensating for $\Delta\mathbf{x}$, a corrected solution $\mathbf{q}_{\text{real}}$ that takes into account the real forward kinematics of the robot in a linearized way can be computed in closed form, based on the following equation:

$$\mathbf{x}_{\text{desired}} - \Delta\mathbf{x} = h_{\text{ideal}}\left(\mathbf{q}_{\text{real}}\right).$$

4. Optimization and Planning Methods for Robotic Applications

In this chapter, the necessary components to perform planning and optimization in robotic applications are described. Namely, the workflow is presented in the next section. Suitable optimization methods are given in Sect. 4.2, and Sect. 4.3 derives optimization criteria relevant to the problems aimed at. New methods for registration and for assisting in the intraoperative positioning are presented in Sect. 4.4 and 4.5, respectively. Section 4.6 concludes the chapter and gives a short summary.

4.1. Workflows

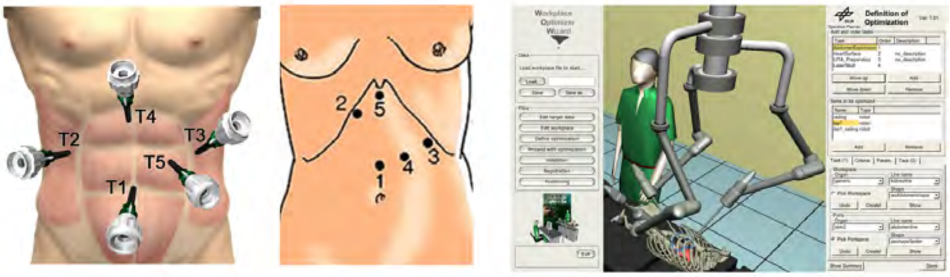


Figure 4.1.: Planning for surgical workplaces can be performed either based on experience and supported by atlases (left according to [74] and middle according to [11]), or a planning software can be used (right).

A workflow denotes the sequence of tasks necessary to fulfill a certain goal. In this section, workflows for both workplace optimization and robot synthesis are presented. In both domains, this thesis focuses on automatic algorithms to support the planning procedure. For many planning tasks, it may also be sufficient to rely on experience, using e.g. atlases [11] as shown in Fig. 4.1(left and middle) or interactive virtual reality planning tools as presented e.g. in [105]. In more complex planning tasks, and especially for robot synthesis, the search space becomes however exceedingly large, and optimization algorithms are therefore essential to keep the time requirements for the decision taking procedure low. Figure 4.1(right) shows the planning software developed in this thesis that uses optimization algorithms.

4.1.1. Workplace Optimization

The workflow in a procedure that uses algorithms to optimize a robotic workplace can be summarized as shown in Fig. 4.2: First the data of the target workpiece are acquired. This step basically consists of importing the relevant physical data of the workpiece from e.g. external software such as Pro/E[1] or Amira[2]. In the next step, the workplace is modeled preliminary around the workpiece: Robots are chosen, and all necessary equipment is placed according to the rough scheme the expert has in mind. Furthermore, the parameters of the task(s) to be performed have to be defined. Usually, these parameters comprise a set of TCP frames the robot has to reach throughout the task. Then optimization parameters and criteria have to be defined: The accessible parameters for optimization (e.g. robot base positions, position of the workpiece, robot segment lengths) are chosen and relevant optimization criteria are attributed. The optimization is then performed automatically. The final step in the preoperative phase consists of validating the found setup(s). In case the optimized workplace is not satisfactory, either the workplace is remodeled or the optimization problem is refined and the automatic optimization is performed once again. Once the optimized workplace is satisfactory, the preoperative phase is concluded. Intraoperatively, the workplace and workpiece are then registered with respect to each other and to the preoperative plan (see Sect. 4.4). Eventually, all items in the workplace have to be positioned according to the preoperative plan (see Sect. 4.5).

Alternatively, the whole planning procedure could take place intraoperatively. Figure 4.3 presents a fast procedure that uses intraoperative data to take the actual OR situation into account. Since in this case planning takes place in the OR coordinate system, no registration is necessary. A crucial issue is the acquisition of sufficiently accurate and detailed patient data to assist the surgeon in the planning phase.

4.1.2. Robot Synthesis

The optimization problems for robot synthesis are often very large and thus solving them is time-consuming. Therefore, it is reasonable to perform the optimization offline, e.g. on a cluster as shown in Fig. 4.4. The definition of all optimization parameters, criteria and tasks however can be advantageously supported by a graphical user interface.

[1] Pro/ENGINEER (commonly referred to as Pro/E) is a parametric feature-based three-dimensional Solid modeling CAD software created by Parametric Technology Corporation (PTC), http://www.ptc.com/ [109].

[2] Amira is a software for 3D visualization and volume modeling of mainly medical data. E.g., tools are available to transform tomographic data into volume or surface models, http://www.amiravis.com/.

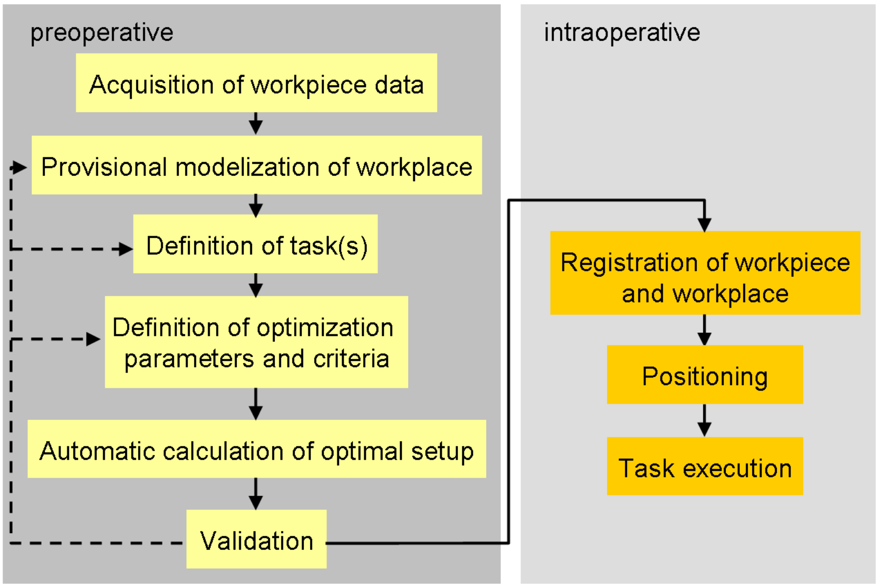

preoperative
Acquisition of workpiece data
Provisional modelization of workplace
Definition of task(s)
Definition of optimization parameters and criteria
Automatic calculation of optimal setup
Validation
intraoperative
Registration of workpiece and workplace
Positioning
Task execution

Figure 4.2.: Workflow for workplace optimization.

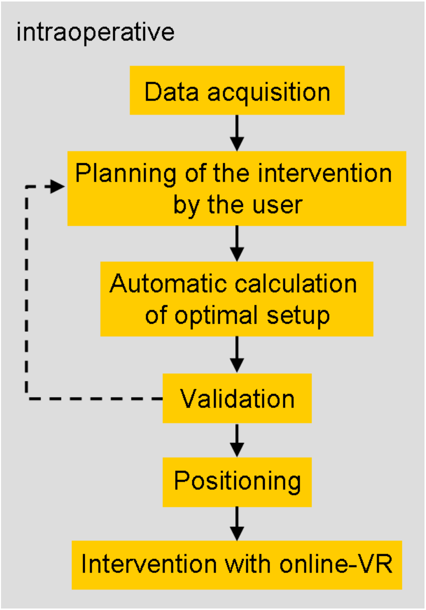

intraoperative
Data acquisition
Planning of the intervention by the user
Automatic calculation of optimal setup
Validation
Positioning
Intervention with online-VR

Figure 4.3.: Workflow for intraoperative planning.

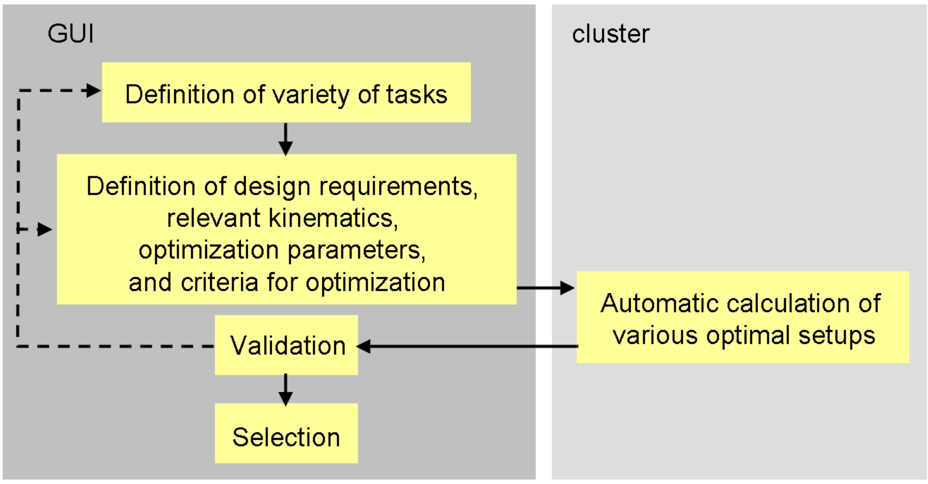

GUI
cluster
Definition of variety of tasks
Definition of design requirements, relevant kinematics, optimization parameters, and criteria for optimization
Automatic calculation of various optimal setups
Validation
Selection

Figure 4.4.: Workflow for robot synthesis.

4.2. Optimization Algorithms

An optimization problem can usually be written as follows, according to [90]:

$$\min \mathbf{f}(\mathbf{p}) = \min \left(f_1(\mathbf{p}), \ldots, f_k(\mathbf{p}), \ldots, f_{n_c}(\mathbf{p})\right), \quad \mathbf{p} \in \mathbb{R}^n, \tag{4.1}$$

$$\begin{aligned} g_j(\mathbf{p}) &= 0, \quad j = 1, \ldots, m_e \\ g_j(\mathbf{p}) &\geq 0, \quad j = m_{e+1}, \ldots, m \\ \mathbf{p}_l \leq \mathbf{p} &\leq \mathbf{p}_u \end{aligned} \tag{4.2}$$

The choice of an adequate algorithm for solving Eq. 4.2 mainly depends on:

1. The dimension of $\mathbf{f}(\mathbf{p})$:
   - (a): If $n_c > 1$, only a pareto-optimal surface can be determined[3].
   - (b): If $n_c = 1$, local minima exist and can be determined.
2. The existence of $g_j(\mathbf{p}), \mathbf{p}_l$, and $\mathbf{p}_u$:
   - (a): Unconstrained optimization: $\mathbf{p} \in \mathbb{R}^n$.
   - (b): Only equality constraints exist: $m_e = m$.
   - (c): Existence of inequality constraints: $m_e < m$, and/or existence of $\mathbf{p}_l, \mathbf{p}_u$.

[3] For a pareto-optimal solution, an improvement in one objective can only occur with the worsening of at least one other objective. Thus, instead of a unique solution to the problem, the solution to a multiobjective problem is a (possibly infinite) set of Pareto points, constituting the pareto-optimal surface.

3. The structure of $f_k(\mathbf{p})$ and $g_j(\mathbf{p})$. The following structures are distinguished in general:

    - (a): Linear functions: $f_k(\mathbf{p}) = \mathbf{c}^{\mathrm{T}}\mathbf{p}$.
    - (b): Quadratic functions: $f_K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^{\mathrm{T}}\mathbf{C}\mathbf{p} + \mathbf{d}^{\mathrm{T}}\mathbf{p}$.
    - (c): Sum of squares: $f_k(\mathbf{p}) = \sum_{i=1}^{m} (y_i - f(\mathbf{p}, t_i))^2$, y_i measured at time t_i.
    - (d): Maximum of functions: $f_k(\mathbf{p}) = \max_{j=1,\ldots,m_f} f_j(\mathbf{p})$.
    - (e): Non-linear differentiable functions.
    - (f): Non-linear non differentiable functions.
    - (g): Non-linear discontinuous functions.

According to the classification of the optimization problem, adequate algorithms can be chosen. This is shown in the following for the optimization problems considered in this thesis.

Optimization Algorithm for Workplace Optimization and Robot Synthesis

In the case of workplace optimization and robot synthesis, optimization parameters arise from e.g. the robot kinematics (sequence of joints, segment lengths), or the positioning of robots and other devices relative to the workspace. Optimization criteria and constraints may address e.g. the avoidance of joint limits and singularities, or the maximization of accuracy, dexterity and manipulability (see Sect. 4.3).

In general, it is interesting to determine the pareto-optimal surface of a multitude of optimization criteria. However, the determination of the pareto-optimal surface is very time-consuming and also not very intuitive if more than three criteria are of interest due to difficulties in visualizing the $(n_c - 1) -$dimensional pareto surface associated with n_c optimization criteria. Therefore, in this thesis the various optimization criteria are combined using a weighting vector $\mathbf{b}$ (case 1(b)):

$$f(\mathbf{p}) = \sum_{i=1}^{n_c} b_i f_i(\mathbf{p}) = \mathbf{b}^{\mathrm{T}}\mathbf{f}(\mathbf{p}), \quad b_i \in \mathbb{R}$$

Usually, the optimization parameters $\mathbf{p}$ are constrained through design or clearance conditions. Furthermore, certain requirements have to be met in all cases, resulting in further inequality constraints. Therefore, case 2(c) is relevant. Eventually, the structure of $f(\mathbf{p})$ and $g_j(\mathbf{p})$ has to be determined. If the whole workspace is reachable, it is usually possible to define non-linear differentiable functions for $f(\mathbf{p})$ and $g_j(\mathbf{p})$ (case 3(e)). Only in the first phase of the optimization procedure, when some positions inside the workspace are unreachable (causing discontinuities in $f(\mathbf{p})$ or $g_j(\mathbf{p})$), case 3(g) has to be considered.

Table 4.1 shows the chosen optimization procedure. Genetic Algorithms[4] (GA) exhibit good performance in case of non-linear discontinuous functions and can provide a variety of quasi optimal solutions, whereas Sequential Quadratic Programming (SQP) methods are generally considered as the most efficient algorithms for optimization of problems as described in case 3(e) [47].

Table 4.1.: Classification of the optimization problem for workplace optimization and robot synthesis

Phase	Problem Classification	Chosen Algorithm
Phase I: not all positions reachable	one-dimensional optimization (1(a)), existence of inequality constraints: $m_e < m$ and existence of $\mathbf{p}_l, \mathbf{p}_u$ (2(c)), non-linear discontinuous functions (3(g)).	Genetic Algorithms (GA)
Phase II: all positions reachable	one-dimensional optimization (1(a)), existence of inequality constraints: $m_e < m$ and existence of $\mathbf{p}_l, \mathbf{p}_u$ (2(c)), non-linear differentiable functions (3(e)).	Sequential Quadratic Programming (SQP)

Therefore, a two-step procedure is chosen. The procedure to assign a quality value $f(\mathbf{p})$ (to be minimized) is shown in Fig. 4.5 (with some sample optimization criteria). It integrates compulsory criteria in step one and relaxable criteria in step two. The function $f(\mathbf{p})$ yields:

$$f(\mathbf{p}) = \begin{cases} f^1(\mathbf{p}) & \text{if} \quad \exists\, f_k^2(\mathbf{p}) > f_{k,\min}^2\,, \\ f^2(\mathbf{p}) & \text{else}\,. \end{cases} \tag{4.3}$$

In case not all minimum demands $f_{k,\min}^2$ are fulfilled, the quality function $f^1(\mathbf{p})$ is chosen in (4.3) and optimized by GA. It is defined as follows:

$$f^1(\mathbf{p}) = a \sum_{k=1}^{n_c} \left(10^{(n_c-k)} f_k^1(\mathbf{p})\right) + \mathbf{b}^{\mathrm{T}} \left(f_{2,\min}^2, \ldots, f_{n_c,\min}^2\right)^{\mathrm{T}}, \quad \text{with} \quad a = \frac{1}{\sum_{h=0}^{n_c-1} 10^h} \tag{4.4}$$

a normalization factor and n_c the number of considered criteria. The performance index $f_k^1(\mathbf{p}) \in [0,\ldots 1]$ expresses the rate of configurations throughout the workspaces with $f_k^2(\mathbf{p}) > f_{k,\min}^2$, thus violating certain minimum demands $f_{k,\min}^2$ for criterion k (see Eq. 4.5 below for the definition of $f_k^2(\mathbf{p})$ and $f^2(\mathbf{p})$):

[4] A genetic algorithm is an optimization technique based on global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover.

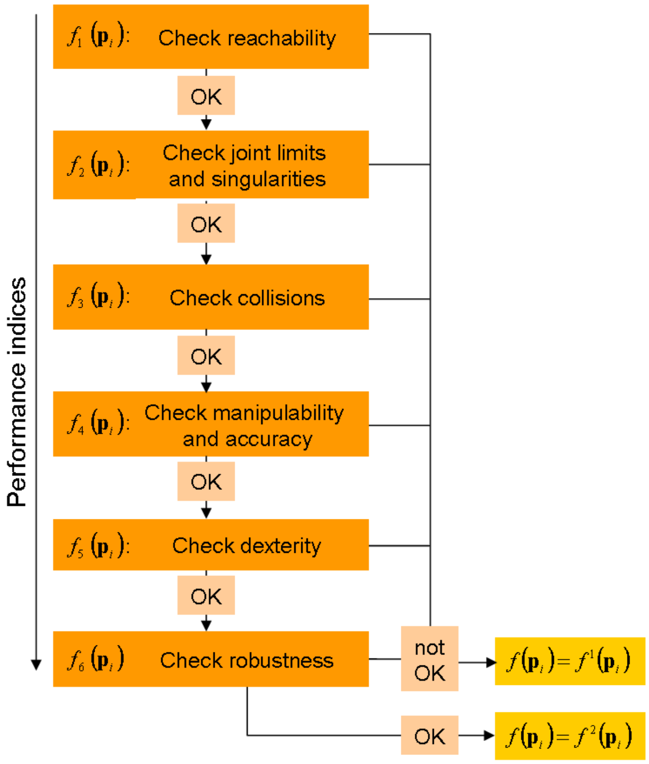

Performance indices
$f_1(\mathbf{p}_i)$: Check reachability
OK
$f_2(\mathbf{p}_i)$: Check joint limits and singularities
OK
$f_3(\mathbf{p}_i)$: Check collisions
OK
$f_4(\mathbf{p}_i)$: Check manipulability and accuracy
OK
$f_5(\mathbf{p}_i)$: Check dexterity
OK
$f_6(\mathbf{p}_i)$ Check robustness
not OK
$f(\mathbf{p}_i)=f^1(\mathbf{p}_i)$
OK
$f(\mathbf{p}_i)=f^2(\mathbf{p}_i)$

Figure 4.5.: Evaluation of a parameter vector $\mathbf{p}_i$.

$$f_k^1 = \frac{\text{number of configurations with } f_k^2(\mathbf{p}) > f_{k,\min}^2}{\text{number of all configurations}} .$$

In case a criterion v is violated, all performance indices $f_w^1(\mathbf{p})$ with $w > v$ are set to 1 without calculation. The term $\mathbf{b}^{\mathrm{T}}\left(f_{2,\min}^2, \ldots, f_{n_c,\min}^2\right)^{\mathrm{T}}$ is included to ensure a continuous transition between $f^1(\mathbf{p})$ and $f^2(\mathbf{p})$. The function $f^1(\mathbf{p})$ is defined as to guarantee that $f^1(\mathbf{p}_a) > f^1(\mathbf{p}_b)$ if the index of the first violated criterion for parameter vector $\mathbf{p}_a$ is lower than for $\mathbf{p}_b$. This way, the criteria k are ordered pursuant to their importance.

The quality function $f^2(\mathbf{p})$ (optimized using SQP), lower case in (4.3), is chosen, provided that all minimum demands $f_{k,\min}^2$ are fulfilled. It is derived as follows: The performance index $f_k^2(\mathbf{p})$ yields a quality value considering the worst occurring configuration among all quality values $f_k^2(\mathbf{p})$ throughout the workspaces with respect to criterion k. In the quality function $f^2(\mathbf{p})$, the performance indices $f_k^2(\mathbf{p})$ are combined:

$$f^2(\mathbf{p}) = \mathbf{b}^{\mathrm{T}}\left(\; f_2^2(\mathbf{p}) \quad, \ldots, \quad f_{n_c}^2(\mathbf{p}) \;\right)^{\mathrm{T}} . \tag{4.5}$$

Reachability (criterion 1) is guaranteed for a parameter vector $\mathbf{p}_i$, if $f_1^2(\mathbf{p}_i) < f_{1,\min}^2$ and is therefore omitted in the quality function $f^2(\mathbf{p}_i)$.

The quality function $f(\mathbf{p})$ has proven to be well suited for the formulation of the optimization problems within this thesis, since $f^1(\mathbf{p})$ enables a fast approach to good initial solutions (i.e. complying with all compulsory criteria) using Genetic Algorithms, whereas $f^2(\mathbf{p})$ allows for fine tuning of the (possibly concurring) relaxable optimization criteria using gradient-based methods. In the presented method stress is laid on compliance with compulsory requirements, and the weighting factor $\mathbf{b}$ merely serves to explore the convex part of the Pareto curve. Note that $f_k^1(\mathbf{p})$ implies also calculation of $f_k^2(\mathbf{p})$, thus computation of $f^2(\mathbf{p})$ is computationally cheap once $f^1(\mathbf{p})$ has been calculated. Due to the use of Genetic Algorithms, the proposed optimization scheme determines a variety of alternative solutions that comply with the compulsory criteria. This may allow the user to choose a solution that also complies with additional requirements (not known in advance), such as installation space for the components.

The considered optimization criteria are presented below. Dynamics are not included as optimization criteria since the occurring velocities of the robotic arms remain low.

4.3. Optimization Criteria and Constraints

Among the variety of optimization criteria and constraint formulations presented in literature so far, the following section presents those criteria that have proven in the considered examples to be well suited as well as specific criteria for the DLR MIRS setup. Usually, all following criteria can be included either as optimization criteria or as constraints into the optimization algorithm, depending on the user's needs.

4.3.1. Avoidance of Joint Limits

To take into account joint limits, the function $f_{\text{joint\_limits}}$ (to be minimized) is suggested:

$$\begin{aligned} f_{\text{joint\_limits}} &= \sum_i f_{\text{joint\_limit}_i}\,, \text{ with} \\ f_{\text{joint\_limit}_i} &= \begin{cases} \left(q_i + q_d - q_{\text{upper\_limit}_i}\right)^2, \text{ if } q_i > q_{\text{upper\_limit}_i} - q_d \\ \left(q_i - q_d - q_{\text{lower\_limit}_i}\right)^2, \text{ if } q_i < q_{\text{lower\_limit}_i} + q_d \\ 0 \quad \text{else} \end{cases} \end{aligned} \tag{4.6}$$

In Eq. 4.6, $q_{\text{upper\_limit}_i}$ and $q_{\text{lower\_limit}_i}$ denote the upper and lower joint limits of joint i respectively, whereas q_d denotes a small offset $q_d > 0$ to ensure a reasonable distance from the joint limits.

4.3.2. Avoidance of Singularities

In case a closed form determination of the singular configurations can be given, a criterion similar to the avoidance of joint limits can be easily formulated. Otherwise, the following function based on the condition number $\kappa(\mathbf{J})$ (using the Euclidean norm) of the Jacobian can be used:

$$f_{\text{singularities}} = \kappa(\mathbf{J}) = \frac{\sigma_{\max}(\mathbf{J})}{\sigma_{\min}(\mathbf{J})}, \tag{4.7}$$

with $\sigma_{\min}(\mathbf{J})$ and $\sigma_{\max}(\mathbf{J})$ the minimal resp. maximal singular value[5] of $\mathbf{J}$.

4.3.3. Joint Velocities

If an initial solution for the joint angles is given, one is usually interested in a robot configuration that minimizes the distance between the initial solution and the calculated one. In a real-time context, this is equivalent with a joint velocity minimization. The following quality function is suggested:

$$f_{\text{joint\_velocities}} = \sum_i \left(q_i - q_{\text{init}_i}\right)^2 .$$

4.3.4. Collision Avoidance

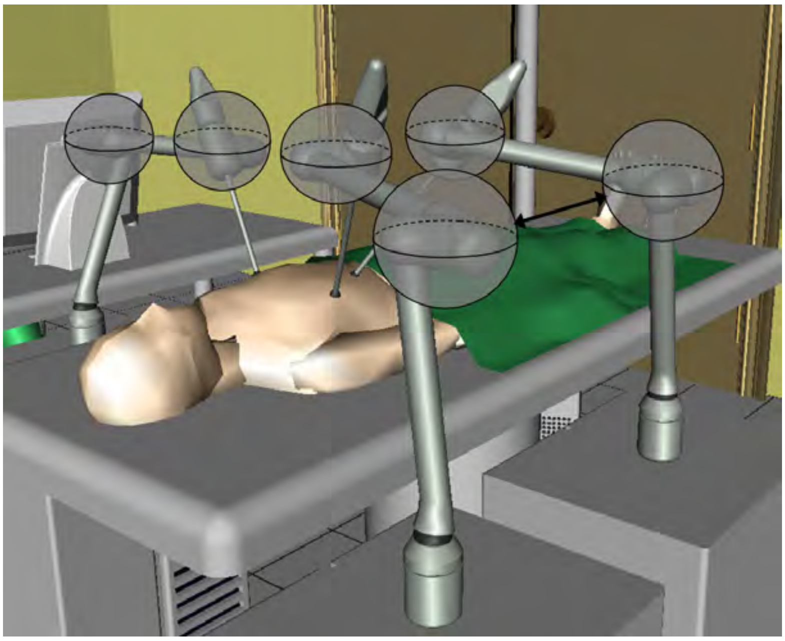


Figure 4.6.: In a preliminary collision check, the distances between control spheres attached to the robot elbows and wrists are considered.

[5] The singular values σ_i of the Jacobian matrix $\mathbf{J}$ are defined to equal the positive square roots of the eigenvalues λ_i of $\mathbf{J}^{\mathrm{T}}\mathbf{J}$.

In telemanipulation, collisions of the instruments should not be excluded by the system, since they are necessary and brought about by the operator to enable interaction between the tools (as e.g. in suturing tasks in medical robotics). However, collisions of the robotic arms should clearly be avoided. To save computation time, only a preliminary collision check is performed during the evaluation procedure by attaching virtual spheres S_l to the elbow (intersection of q_4 and q_5) and the wrist (intersection of q_6 and q_7) of the robot (see Fig. 4.6) and then cross-checking the distances between every sphere l of robot r with the spheres attached to all other robots in every of their configurations k throughout the workspace. With $n_\mathrm{p} = 100$ sampled points in the discretized workspace, $n_\mathrm{S} = 2$ control spheres per robot and $n_\mathrm{R} = 3$ robots, a total of

$$n_\mathrm{S}^2 n_\mathrm{p} \sum_{i=1}^{n_\mathrm{R}-1} i = 1200 \tag{4.8}$$

distances have to be calculated. The minimum occurring distance is used as measure for collision. For more thorough collision checks, according libraries (e.g. I-COLLIDE [17] or SOLID [106]) can be included.

4.3.5. Dexterity

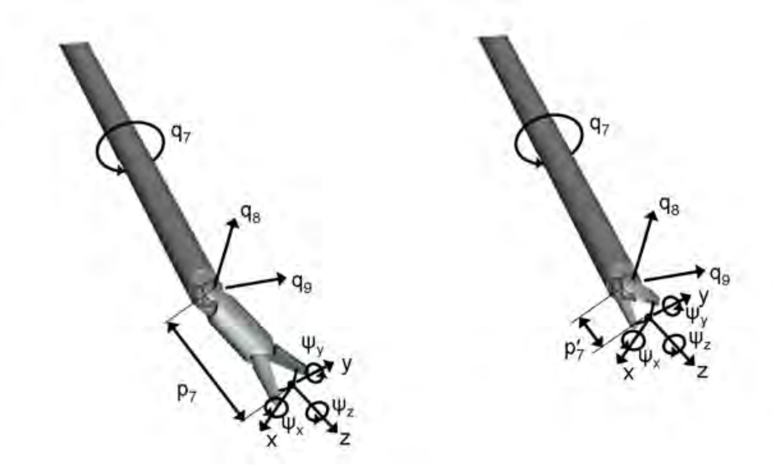


Figure 4.7.: The available purely rotational motion $\psi_{x,y,z}$ around the target tool tip frame is used as dexterity measure.

The tool dexterity for articulated instruments is defined in the scope of this thesis based on the available purely rotational motion $\psi_{x,y,z}$ around the target tool tip frame as shown in Fig. 4.7:

$$f_\mathrm{dexterity} = \sqrt{3} \cdot 2\pi - \sqrt{\Delta\psi_x^2 + \Delta\psi_y^2 + \Delta\psi_z^2},$$

with $\Delta\psi_i = \Delta\psi_{\mathrm{max}_i} - \Delta\psi_{\mathrm{min}_i}$ denoting the available purely rotational motion around axis i. If the tool tip frame is close to the wrist frame with three axes intersecting, this range of motion can be approximated with the joint angles (and the distance from the resp. joint limits) of the wrist joints, since these joints are primarily responsible for pure rotational motion at the tool tip (Fig. 4.7 right). The minimum distance of these joint positions from their joint limits therefore serves to formulate the criterion for dexterity. In case the wrist frame is considerably translated with respect to the tool tip frame (Fig. 4.7 left), different tool tip frame orientations would have to be tested in a brute force manner (see use case in Sect. (6.1.3.1)).

4.3.6. Manipulability and Accuracy Measures

Manipulability and positioning accuracy are of great interest due to the following reasons:

- Singular configurations which could cause system failures must be avoided. If they cannot be determined in closed form beforehand, they can be detected online through supervision of the manipulability (see Sect. 4.3.2).
- A certain minimal velocity of the instrument tip both in terms of translation and rotation must be achieved all over the considered workspace. This is important to assure that motions commanded by the operator can be performed by the robotic system.
- The demanded accuracy of the positioning of the instrument tip must be guaranteed. This is of great interest if very fine structures (e.g. blood vessels) are manipulated.

The classic formulation of manipulability based on [111] investigates the singular values σ_i of the Jacobian matrix $\mathbf{J}$:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}, \tag{4.9}$$

with $\dot{\mathbf{x}} = (v_x\, v_y\, v_z\, \omega_x\, \omega_y\, \omega_z)^{\mathrm{T}}$ the translational and rotational velocities and $\mathbf{J}$ the appropriate Jacobian matrix. The singular values define the shape of the manipulability ellipsoid and therefore give a local description of the available velocities of TCP motions with respect to different directions. Since one is often interested to achieve a uniform manipulability, the following measure can be used (similar to Eq. 4.7):

$$f_{\text{manipulability\_a}} = \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}$$

Alternatively, the overall manipulability might be of interest. An adequate measure could then be the volume of the manipulability ellipsoid (to be maximized):

$$f_{\text{manipulanility\_b}} = \prod_{i=1}^{6} \sigma_i$$

A critical issue concerning the shown measures is the normalization of the translational and rotational parts of $\dot{\mathbf{x}}$ in Eq. 4.9 because of unit discrepancies.

The next section shows that neither $f_{\text{manipulability\_a}}$ nor $f_{\text{manipulability\_b}}$ is usable in MIRS and presents new specially adapted measures.

4.3.7. Manipulability in Minimally Invasive Robotic Surgery

Previous studies concerning preoperative planning of MIRS procedures [1, 14, 25, 61, 95, 100] do not take into account manipulability and positioning accuracy sufficiently. Instead, the used approaches *tool dexterity* [1, 61] as well as *magic pyramid* [14] evaluate the orientation of the instruments, the endoscope, and the surface normal of the considered operating field to each other. Also, the distance of the joint angles of the robot from its maximum values is used as criterion [25]. Experimental trials [1] suggest the conclusion that the use of more sophisticated descriptions of manipulability and accuracy measures is reasonable.

The following drawing up presents measures for manipulability and accuracy that are specifically adapted to the conditions found in robotically assisted minimally invasive surgery. The presented manipulability and positioning accuracy measures are based on an "inverse Jacobian" approach since the constraints at the entry point into the human body forbid a classic formulation (see below). High significance of the measures is reached by including robot design parameters such as encoder resolution and maximum joint velocity.

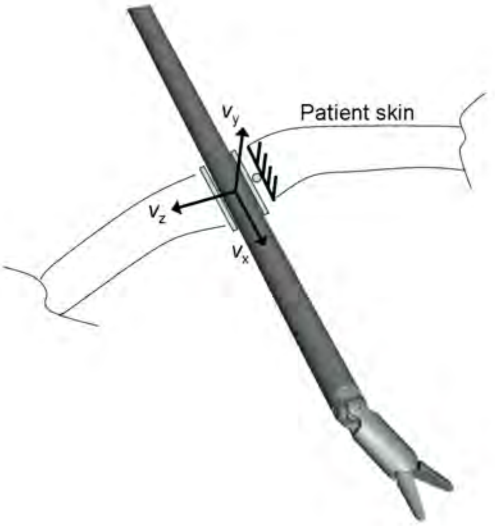


Figure 4.8.: At the entry point into the patient, the instrument motion is restricted so that the translational velocities v_y and v_z are equal to zero.

In this section, the kinematic structure of the DLR medical robot with an actuated instrument as shown in Fig. 3.3 and 3.4 is considered, therefore in Eq. 4.9 the respective dimensions are $\mathbf{J} \in \mathbb{R}^{6\times9}$ and $\dot{\mathbf{q}} \in \mathbb{R}^{9}$. In case of minimally invasive surgery, Eq. 4.9 cannot be used to address manipulability because it does not take into account the motion restriction due to the entry point into the patient and the possibility of nullspace motion. This is shown in the following equations. At the entry point, the following kinematic constraint must be satisfied [78] (see Fig. 4.8):

$$\mathbf{C}_1\dot{\mathbf{q}} = \mathbf{0}\,, \mathbf{C}_1 \in \mathbb{R}^{2\times9}. \tag{4.10}$$

Nullspace motion implies that a given tool tip position and orientation can be achieved with an infinite number of joint sets. To calculate the most appropriate joint set, another constraint has to be introduced which represents an optimization among the possible joint sets:

$$\mathbf{C}_2\dot{\mathbf{q}} = 0\,, \mathbf{C}_2 \in \mathbb{R}^{1\times9}, \tag{4.11}$$

resulting in

$$\mathbf{C}\dot{\mathbf{q}} = \mathbf{0} \quad \text{with} \quad \mathbf{C} = \begin{pmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{pmatrix} \in \mathbb{R}^{3\times9}. \tag{4.12}$$

If the vector of joint velocities $\dot{\mathbf{q}}$ is divided arbitrarily into a dependent part $\dot{\mathbf{q}}_\mathrm{d} \in \mathbb{R}^{3\times1}$ and an independent part $\dot{\mathbf{q}}_\mathrm{i} \in \mathbb{R}^{6\times1}$, it can be reordered as follows:

$$\dot{\mathbf{q}} = \begin{pmatrix} \dot{\mathbf{q}}_\mathrm{d} \\ \dot{\mathbf{q}}_\mathrm{i} \end{pmatrix}. \tag{4.13}$$

The Equations 4.9 and 4.12 can then be rewritten as [78]:

$$\dot{\mathbf{x}} = \mathbf{J}_\mathrm{d}\dot{\mathbf{q}}_\mathrm{d} + \mathbf{J}_\mathrm{i}\dot{\mathbf{q}}_\mathrm{i} \quad \text{with} \quad \mathbf{J} = (\mathbf{J}_\mathrm{d}\ \mathbf{J}_\mathrm{i}) \tag{4.14}$$

$$\text{and} \quad \dot{\mathbf{q}}_\mathrm{d} = \mathbf{B}\dot{\mathbf{q}}_\mathrm{i} \quad \text{with} \quad \mathbf{B} \in \mathbb{R}^{3\times6}\,. \tag{4.15}$$

After insertion of the kinematic constraint (Eq. 4.15), Eq. 4.14 has form

$$\dot{\mathbf{x}} = \mathbf{J}'\dot{\mathbf{q}} \quad \text{with} \quad \mathbf{J}' = \mathbf{J}_\mathrm{d}\mathbf{B} + \mathbf{J}_\mathrm{i} \in \mathbb{R}^{6\times6}, \tag{4.16}$$

where only 6 of the 9 joint velocities occur. Depending on which joint velocities are chosen as dependent, the Jacobian matrix $\mathbf{J}'$ has different elements and different singular values. The frequently used manipulability measures based on singular values therefore become arbitrary and are not useful any more.

Instead, a more suitable formulation is achieved if the inverse correlation is used. This can be done by solving Eq. 4.16 for $\dot{\mathbf{q}}_{\mathrm{i}}$ and combining the result with Eq. 4.15 [47]:

$$\dot{\mathbf{q}} = \mathbf{D}\mathbf{v} \quad \text{with} \quad \mathbf{D} = \begin{pmatrix} \mathbf{B}\mathbf{J}'^{-1} \\ \mathbf{J}'^{-1} \end{pmatrix} . \tag{4.17}$$

This equation includes all joint velocities. It relates a given instrument velocity $\dot{\mathbf{x}}$ to the joint velocities $\dot{\mathbf{q}}$ in a non-ambiguous way for every non-singular robot configuration by consideration of the kinematic constraint due to the entry point. Alternatively to the above described calculations, matrix **D** can also be obtained by numerical differentiation. This can be done by applying difference quotients to the inverse kinematics which expresses the joint angles as function of the instrument tip position and orientation and the position of the entry point.

To define a measure for manipulability, a connection between the maximum joint velocity $\dot{q}_{\max}$ and the desired minimal translational velocity $\dot{x}_{\min}$ and angular velocity $\dot{\alpha}_{\min}$ of the instrument tip has to be established. To do so, the matrix **D** in Eq. 4.17 is split into two components

$$\mathbf{D} = (\mathbf{E}\,\mathbf{F}) \quad \text{with} \quad \mathbf{E} = \begin{pmatrix} e_{11} & & e_{13} \\ \vdots & \ddots & \vdots \\ e_{81} & & e_{83} \end{pmatrix}, \mathbf{F} = \begin{pmatrix} f_{11} & & f_{13} \\ \vdots & \ddots & \vdots \\ f_{81} & & f_{83} \end{pmatrix}, \tag{4.18}$$

and Eq. 4.17 is rewritten:

$$\dot{\mathbf{q}} = \mathbf{E}\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} + \mathbf{F}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}, \tag{4.19}$$

with $v_{x.y.z}$ the translational velocity and $\omega_{x,y,z}$ the rotational velocity components. For each joint $i \in \{1, ..., 9\}$, the maximum velocity $\dot{q}_{i,\ \max}$ is computed by solving the following optimization problem:

$$\{\dot{q}\}_{i,\max} = \left\{ |\{\dot{q}\}_i(\dot{\mathbf{x}})| \xrightarrow{\text{opt}} \max \right\} = \{|e_{i1}v_x + e_{i2}v_y + e_{i3}v_z$$
$$+ f_{i1}\omega_x + f_{i2}\omega_y + f_{i3}\omega_z| \xrightarrow{\text{opt}} \max\Big\} \tag{4.20}$$

under the constraints

$$\sqrt{v_x^2 + v_y^2 + v_z^2} - \dot{x}_{\min} \leq 0 \quad \text{and} \quad \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2} - \dot{\alpha}_{\min} \leq 0 . \tag{4.21}$$

The problem is solved by using the Lagrange function and yields (see [47] for further details):

$$\dot{q}_{i,\ \max} = \sqrt{e_{i1}^2 + e_{i2}^2 + e_{i3}^2}\,\dot{x}_{\min} + \sqrt{f_{i1}^2 + f_{i2}^2 + f_{i3}^2}\,\dot{\alpha}_{\min} \ .$$

Thus the maximum joint velocity $\dot{q}_{\max} = \max(\dot{q}_{i,\,\max})$ is determined which can be used as a reciprocal manipulability measure (to be maximized):

$$f_{\mathrm{man}} = \frac{1}{\dot{q}_{\max}} \,. \tag{4.22}$$

For a given, desired minimal velocity at the tool tip, low values of $\dot{q}_{\max}$ denote that the joint velocities remain low. This indicates good manipulability.

4.3.8. Positioning Accuracy in Minimally Invasive Robotic Surgery

The following question is considered to define an (in)accuracy measure:

> How far can the instrument tip be moved at maximum with all the changes in the articular space remaining below the encoder resolution $\Delta q_{\min}$?

This describes the maximum translational and rotational movement $\Delta x_{\min}$ and $\Delta\alpha_{\min}$ of the instrument tip that is not detectable by the robot control system and thus not commandable. It provides a minimum estimation of the positioning accuracy that can be achieved.

For small changes $\Delta\mathbf{q}$ and $\mathbf{u}$, where $\mathbf{u} = \begin{pmatrix} \mathbf{u}_{\mathrm{trans}} \\ \mathbf{u}_{\mathrm{rot}} \end{pmatrix}$ describes small displacements of the instrument tip and small changes in the orientation of the instrument, the following relation approximately holds:

$$\Delta\mathbf{q} = \mathbf{D}\mathbf{u} \,. \tag{4.23}$$

Since the vector $\mathbf{u}$ contains both translational and rotational components, normalization is applied:

$$\Delta\mathbf{q} = \tilde{\mathbf{D}}\tilde{\mathbf{u}} \quad \text{with} \quad \tilde{\mathbf{u}} = \begin{pmatrix} \mathbf{u}_{\mathrm{trans}}/\Delta x_{\min} \\ \mathbf{u}_{\mathrm{rot}}/\Delta\alpha_{\min} \end{pmatrix},\ \tilde{\mathbf{D}} = (\Delta x_{\min}\mathbf{E}\ \Delta\alpha_{\min}\mathbf{F}) \,. \tag{4.24}$$

The matrix $\tilde{\mathbf{D}}$ maps small displacements $\tilde{\mathbf{u}}$ in Cartesian space onto small displacements $\Delta\mathbf{q}$ in joint space. Displacements in joint space caused by Cartesian displacements with $\|\tilde{\mathbf{u}}\|_2 \leq 1$ lie within an ellipsoid [111]. The length of the main axes of the ellipsoid are given by the singular values $\tilde{\sigma}_i$ of the matrix $\tilde{\mathbf{D}}$. Thus, calculating the smallest singular value $\tilde{\sigma}_{\min}$ of $\tilde{\mathbf{D}}$ yields [47]:

$$\|\Delta\mathbf{q}\|_2 \geq \tilde{\sigma}_{\min}\,\|\tilde{\mathbf{u}}\|_2 \,. \tag{4.25}$$

One is interested in the maximum change of one of the joint angles which is synonymous to the maximum norm $\|\Delta\mathbf{q}\|_\infty$ and not to the Euclidean norm as appearing in Eq. 4.25. Therefore, the following estimation is used:

$$\|\Delta\mathbf{q}\|_2 \leq \|\Delta\mathbf{q}\|_\infty \cdot \sqrt{\dim(\Delta\mathbf{q})}\,. \tag{4.26}$$

With Eq. 4.24 and movements $\|\mathbf{u}_{\text{trans}}\|_2 \geq \Delta x_{\text{min}}$ and $\|\mathbf{u}_{\text{rot}}\|_2 \geq \Delta\alpha_{\text{min}}$, Eq. 4.27 holds:

$$\|\tilde{\mathbf{u}}\|_2 = \sqrt{\|\mathbf{u}_{\text{trans}}\|_2^2/\Delta x_{\text{min}}^2 + \|\mathbf{u}_{\text{rot}}\|_2^2/\Delta\alpha_{\text{min}}^2} \geq \sqrt{2}\,. \tag{4.27}$$

Thus, with the estimations in Eq. 4.25, 4.26 and 4.27, the following equation is obtained:

$$\|\Delta\mathbf{q}\|_\infty \cdot \sqrt{\dim(\Delta\mathbf{q})} \geq \sqrt{2}\,\tilde{\sigma}_{\text{min}}. \tag{4.28}$$

Displacements in Cartesian space with $\|\mathbf{u}_{\text{trans}}\|_2 \geq \Delta x_{\text{min}}$ and $\|\mathbf{u}_{\text{rot}}\|_2 \geq \Delta\alpha_{\text{min}}$ therefore cause displacements in joint space of at least $\Delta q_{\text{min}} = \|\Delta\mathbf{q}\|_\infty$. Thus Δq_{min} is an estimation for the necessary encoder resolution to detect movements greater than Δx_{min} resp. $\Delta\alpha_{\text{min}}$, and the following measure f_{acc} for positioning accuracy can be established (from Eq. 4.28):

$$f_{\text{acc}} = \Delta q_{\text{min}} = \frac{\tilde{\sigma}_{\text{min}}}{\sqrt{\frac{1}{2}\dim(\Delta\mathbf{q})}}\,. \tag{4.29}$$

Higher values for f_{acc} denote better accuracy.

4.3.9. Comparison between Manipulability and Accuracy

Figure 4.9 shows the positioning accuracy measure f_{acc} and the manipulability measure f_{man} in MIRS applications for different instrument tip positions in the vicinity of a singularity. Since f_{man} decreases when approaching the border line to singularity, it can be used to avoid singular configurations.

The measure for positioning accuracy f_{acc} mainly shows a reverse behavior compared to f_{man}. This is understandable since high manipulability is defined to allow fast motion of the tool tip with low joint velocities, whereas high positioning accuracy means small displacements with large joint rotations. Therefore, during validation of a robot configuration, a trade off between high manipulability and high positioning accuracy has to be found.

The modified measures for manipulability and accuracy presented here are suitable to indicate problems such as unexpected singularities or poor manipulability that were encountered with previous approaches in the field of MIRS. The measures are very descriptive since they comprise the robotic design parameters *encoder resolution* and *maximum joint velocity*. In order to establish quantitative information whether a certain value for f_{man} or f_{pos} is sufficient to accomplish MIRS, the requirements concerning desired velocity and positioning accuracy of the tool tip have to be known. In this context, work remains to be done to analyze MIRS procedures more thoroughly since previous publications [13, 40, 86, 97] only cover analyses of translational aspects.

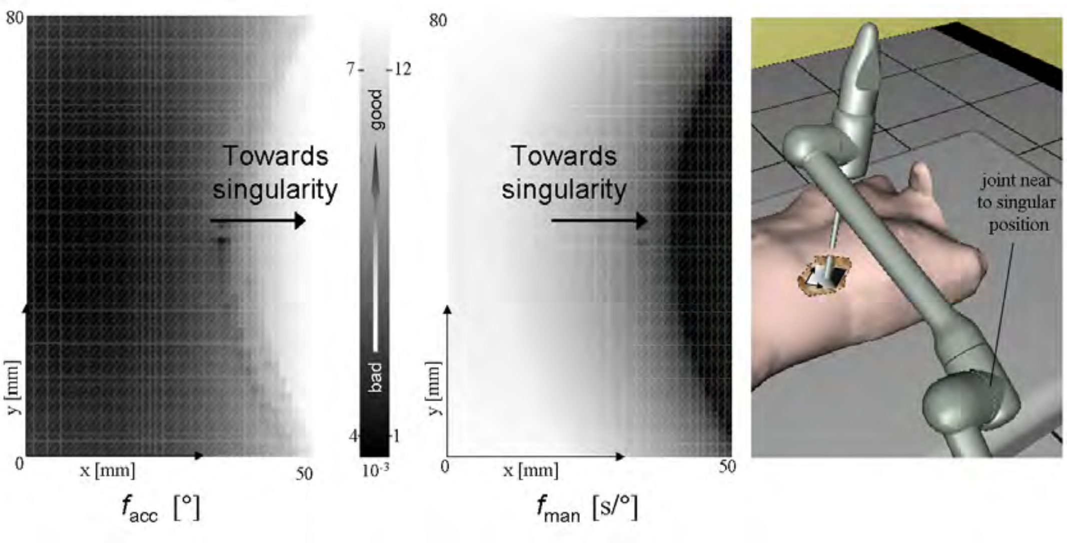


Figure 4.9.: The measures f_{man} and f_{acc} in the vicinity of a singular configuration.

4.3.10. Robustness

The transfer of the planned optimal setup of e.g. robot base positions into the real environment introduces positioning errors, mainly due to inaccurate registration or, in case of the surgical environment, also caused by motion of the patient and organs (due to heartbeat, breathing or insufflation) as well as measuring faults. These errors should be reduced as far as possible, but they can never be completely eliminated. It is thus important to ensure that the considered task can be carried out even if the real setup slightly deviates from the optimal setup. This is ensured by an optimization that also considers small deviations of the relevant parameters (see [53]): E.g. the robot base position and (for MIRS) the entry points into the patient are varied successively with a certain tolerance value and the optimization is then rerun.

4.4. Registration

Registration denotes the procedure of finding a transformation rule $\boldsymbol{R}$ for transfer of different representations (e.g. point clouds representing the object surface) of the same object into a common coordinate system.

In the scope of this thesis, one is mainly interested in transferring the results of the planning procedure into the real environment, performing a rigid registration, i.e. the transformation rule $\boldsymbol{R}$ can be described through a homogeneous transformation. Figure 4.10 shows the relevant coordinate systems and frames, with $\{P\}$ and $\{\tilde{P}\}$ the

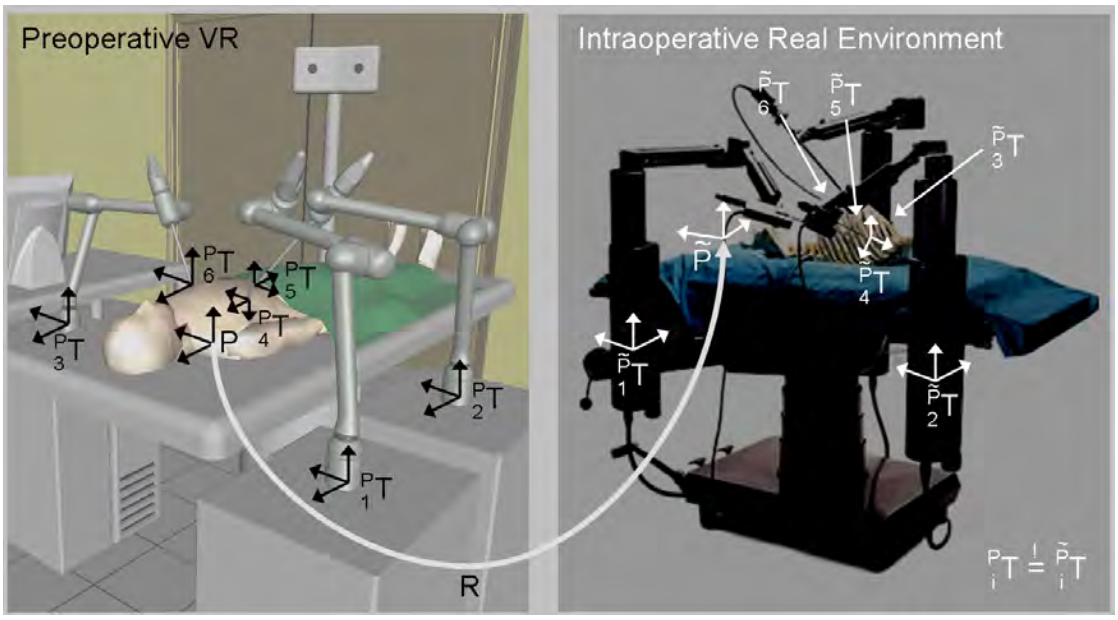


Figure 4.10.: Registration determines the transformation rule R to relate the preoperative coordinate system $\{P\}$ with the intraoperative coordinate system $\{\tilde{P}\}$ (right photo courtesy of [88]).

coordinate systems of the pre- and intraoperative environment[6] and ${}^{P}_{i}\boldsymbol{T}$ resp. ${}^{\tilde P}_{i}\boldsymbol{T}$ the relative frames of robot bases and entry points into the patient. Through registration, the transformation rule $\boldsymbol{R}$ is determined, and in the subsequent step (see next section) robots and ports are positioned so that

$$ {}^{\tilde P}_{i}\boldsymbol{T} \overset{!}{=} {}^{P}_{i}\boldsymbol{T} \quad \text{holds.} $$

In the medical context, standard approaches are often based either on artificial landmarks (e.g. a dental splint with markers or bone implanted markers) or anatomical landmarks. These landmarks are segmented in the preoperative image data and localized in the OR with e.g. optically tracked pointing devices. Corresponding intra- and preoperative landmark positions are then determined, and the (rigid) transformation matrix is calculated using least square fitting algorithms. Markerless methods using e.g. the patient skin or bone surface are in clinical use, too (e.g. z-touch from BrainLAB [4]). However, contact-free systems still lack dexterity and accuracy and are a current research topic [57, 66]. Section 6.2.5 presents a markerless and contact-free registration procedure using the DLR 3D-Modeller (3DMo).

[6] It is assumed that the workplace is arranged according to the (fixed) position of the working field (i.e. the patient in the shown example). Alternatively, another inertial coordinate system could be introduced, and the whole workplace could be arranged accordingly.

4.5. Positioning

To transfer preoperatively planned data into the real environment, registration is necessary as well as a method to then localize the planned data. This data may comprise e.g. entry point positions into the human body in case of minimally invasive interventions or cutting trajectories. State of the art methods for localization are e.g.:

- The robot itself is used as a pointing device (exploiting the forward kinematics) to position other devices and the workpiece with respect to each other (see e.g. [18]).
- An optically tracked pointer is used to find positions in the real environment, assisted either by a virtual reality (VR) representation of the scene where the pointer is visualized (see Fig. 4.11), or by simple commands ("move right/left/up/down...").
- The plan is projected with e.g. a video or laser projector (see e.g. [24, 44, 55]).

This section presents a first prototype of the *AutoPointer* [51], a new patent pending device using a handheld optically tracked laser scanner to localize preoperatively planned data in the real environment automatically. With the *AutoPointer*, it is possible e.g. to automatically show the right positions for the entry points into the patient in the OR by projecting them onto the patient. The underlying idea of the *AutoPointer* is shown in Fig. 4.12: The beam of a laser is deflected by a deflection unit. Laser and deflection unit are housed in a handheld device which is localized with a tracking system. If e.g. a point $\boldsymbol{P}$ on the patient surface has to be transferred from the preoperative plan into the OR, registration provides the transformation rule $\boldsymbol{R}$ to obtain the point coordinates $\boldsymbol{P}_{\mathrm{OR}}$ in the OR coordinate system: $\boldsymbol{P}_{\mathrm{OR}} = \boldsymbol{R}(\boldsymbol{P}_{\mathrm{preoperative}})$. Then, with the known (and constantly updated) position of the *AutoPointer* relative to $\boldsymbol{P}_{\mathrm{OR}}$, the deflection unit can deflect the laser beam so that it automatically points onto $\boldsymbol{P}_{\mathrm{OR}}$. The first prototype of the device is shown in Fig. 4.13. As deflection unit a Scanlab ScanCube 7 [89] is used. The ScanCube 7 is equipped with position-controlled high speed galvo-scanners to deflect a laser beam in two axes. A digital signal processing (DSP) board is used to control the scanner. The scanner is fast enough to project various data, including complex cutting trajectories and multiple symbols at the same time. The *AutoPointer* is optically tracked using an A.R.T. navigation system[7]. The system is portable and thus only takes up valuable space in the workplace while needed. It can be easily interfaced with any tracking system that is able to measure the 6-dimensional pose of an object. As opposed to video projectors, laser light is visible even in the presence of strong lighting as used in the OR. The first mock-up of the system shows an accuracy better than 1 mm in a distance of 0.5 m. Optimization of the tracking target and use of more accurate tracking systems will further increase the accuracy. The control loop currently is fast enough to compensate for motions with a frequency below 1 Hz. Various information such as arrows, crosses, words, or trajectories can be projected directly onto the patient as shown in Fig 4.14.

[7] See http://www.ar-tracking.de/. Accessed on March 13th, 2007.

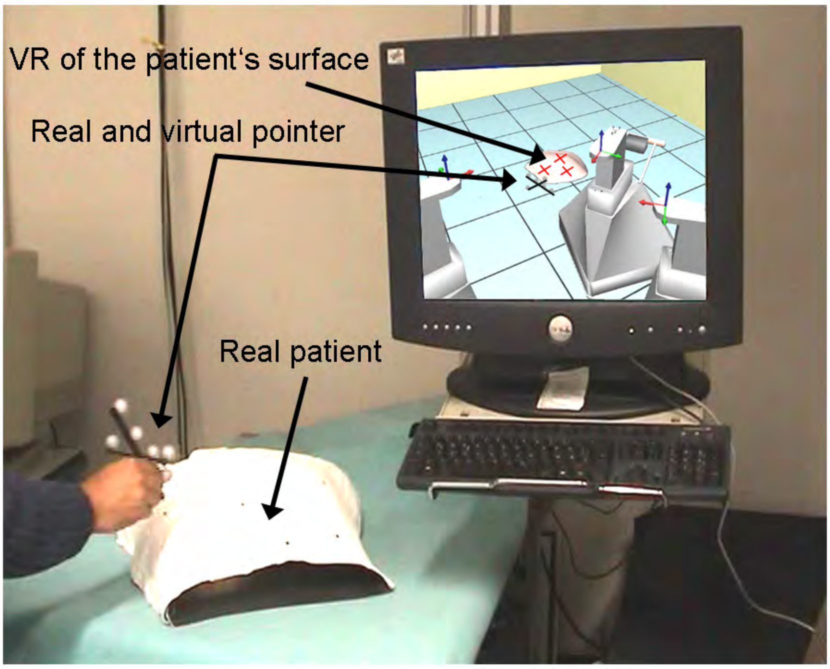


Figure 4.11.: Positioning with a tracked pointer and online VR feedback. The red crosses in the VR denote the optimal entry points into the human body that have to be transferred into the OR.

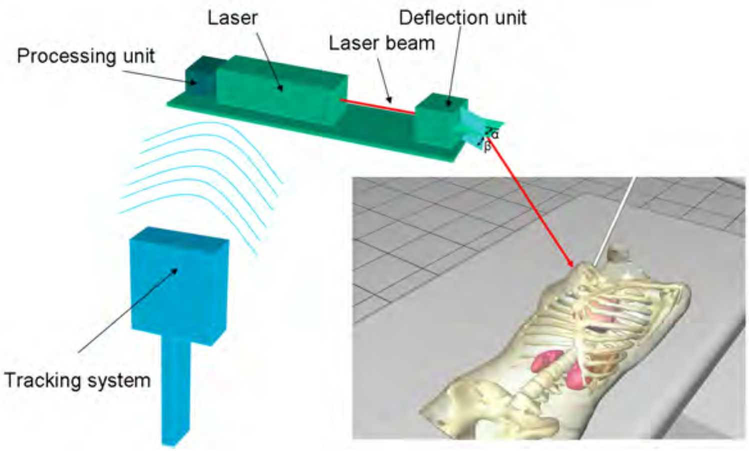


Figure 4.12.: Components of the *AutoPointer*.

4.6. Summary and Conclusion

This chapter presents the optimization and planning methods necessary for workplace optimization and robot synthesis.

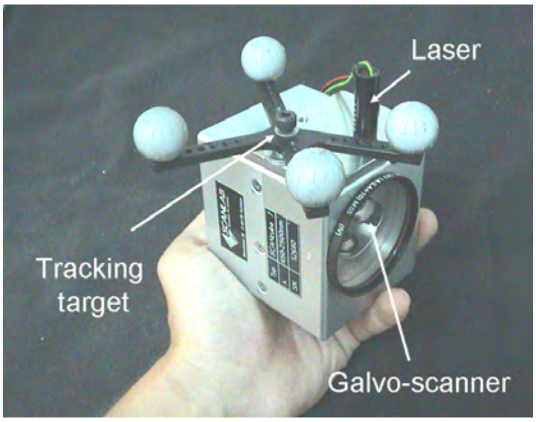


Figure 4.13.: First prototype of the *AutoPointer*.

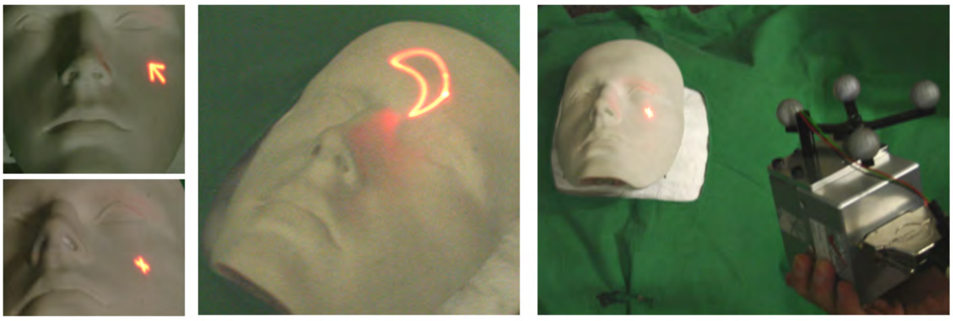


Figure 4.14.: Various symbols and arbitrary trajectories can be projected onto the patient.

Namely, the workflows are developed for preoperative and intraoperative workplace optimization as well as for robot synthesis. Optimization algorithms are then categorized, and a two-step algorithm based on Genetic Algorithms and a subsequent SQP method is chosen for the workplace optimization and robot synthesis problems addressed in this thesis. The two-step approach is advantageous since the Genetic Algorithms provide a variety of potential solutions without resorting to a single local minimum. Furthermore, Genetic Algorithms exhibit good performance in case of non-linear discontinuous functions as appearing in the first phase of the optimization due to reachability constraints. The subsequent gradient-based method provides fast convergence towards the local minima.

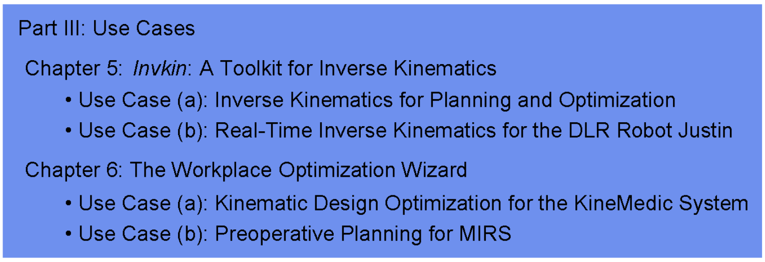
Various optimization criteria and constraints are then presented, including specialized criteria for accuracy and manipulability in MIRS applications.

To conclude the necessary steps in a workplace optimization or robot synthesis problem, registration is then introduced, and methods for transferring the results of the planning into the real workplace are shown. Namely, for the intraoperative positioning a new augmented reality device is presented, the *AutoPointer*. It automatically marks the relevant positions inside the workplace, thus facilitating the transfer of data between virtual and real world considerably. The implemented control loop allows for moderate motion of the *AutoPointer* while it keeps projecting the desired data automatically at the right positions with an accuracy of below 1 mm. Since the major source of inaccuracy is the optical tracking system, advances in the tracking system accuracy will directly improve the *AutoPointer* accuracy. In future setups it will furthermore be possible to improve the frequency of the control loop, thus allowing even for filtering the tremor. Currently, the frequency is limited by the used tracking system and the communication between scanner, control PC, and tracking system. Furthermore, image processing by the navigation system introduces a delay that has to be further reduced or compensated by an adequate motion prediction.

Part III.

Use Cases

In this part, the theories and methods presented in Part II are applied to the following use cases (see Fig. 4.15):



Part III: Use Cases

Chapter 5: *Invkin*: A Toolkit for Inverse Kinematics

- Use Case (a): Inverse Kinematics for Planning and Optimization
- Use Case (b): Real-Time Inverse Kinematics for the DLR Robot Justin

Chapter 6: The Workplace Optimization Wizard

- Use Case (a): Kinematic Design Optimization for the KineMedic System
- Use Case (b): Preoperative Planning for MIRS

Figure 4.15.: Structure of Part III.

Considerations in Chapter 3 about inverse position kinematics and singularity calculation have led to the implementation of an inverse kinematics library. Its structure is presented in the next chapter, and the functionality of the library is evaluated with two applications: Firstly, different algorithms are compared with each other in a design optimization task. Then, real-time capabilities are shown in an application coming from the field of service robotics, providing inverse kinematics for Cartesian control of the DLR robot Justin shown in Fig. 4.16.

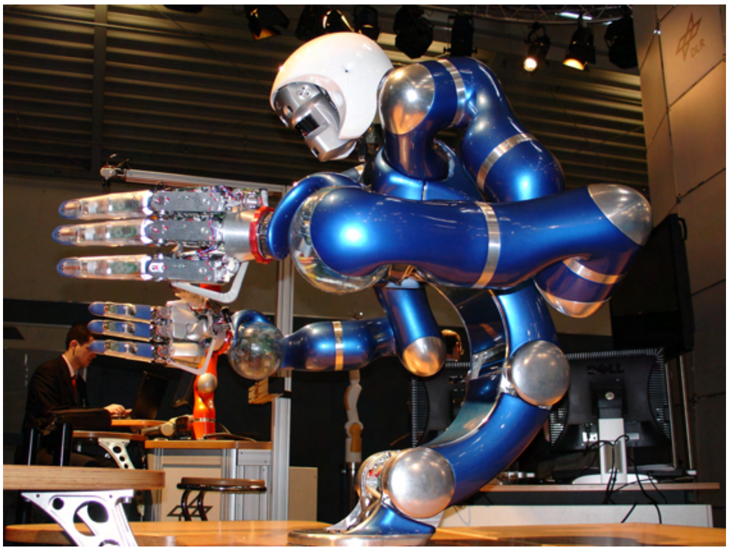


Figure 4.16.: The DLR robot Justin.

In Chapter 6, the methods for optimization and planning procedures developed in Chapter 4 are applied both to a robot synthesis problem and to a workplace optimization. Namely, a design optimization of the DLR medical robot KineMedic is carried out, an optimized design for an actuated carrier to position up to three KineMedic robots is suggested, and a workplace optimization for MIRS applications is presented.

5. *InvKin*: A Toolkit for Inverse Kinematics

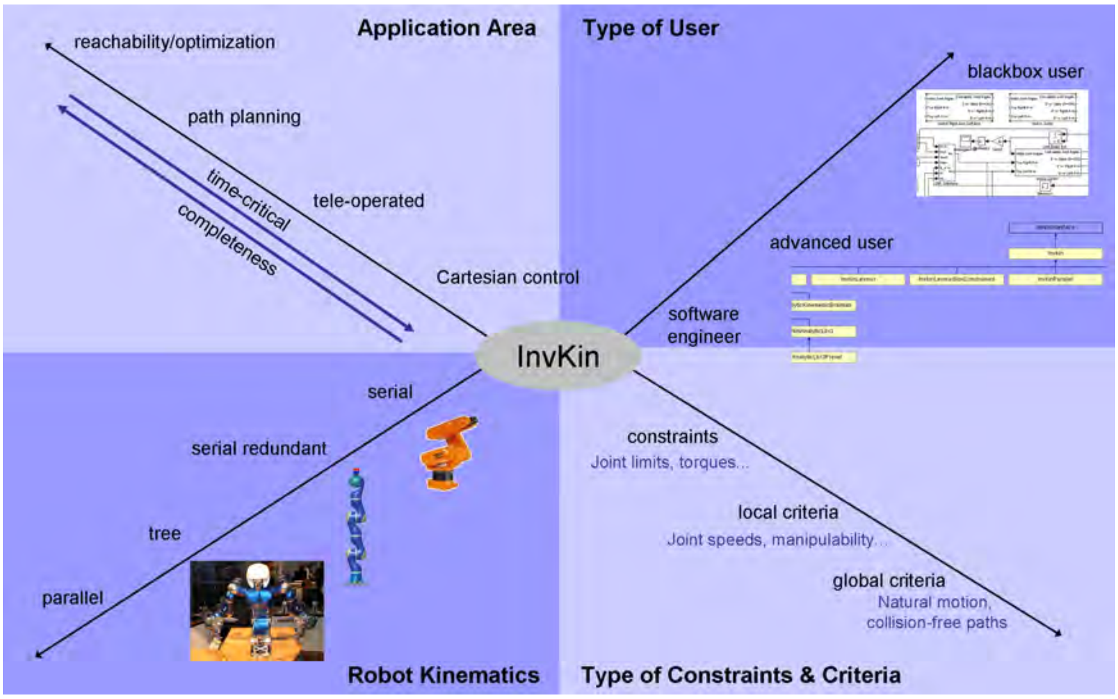


Figure 5.1.: Design criteria for an inverse kinematics library.

Universal tools for calculating the inverse kinematics of arbitrary manipulators are still not available. A reason for this might be the large variety such a tool would have to cover, as shown in Fig. 5.1. Different inverse kinematics algorithms have to be provided according to the following requirements:

1. Application area: In case of reachability and optimization studies, calculation time is not as important as in case of e.g. Cartesian control. However, since an initial solution in general can not be given, a global solution algorithm to the inverse kinematics has to be offered.

2. Type of user: Most of the users will probably use the inverse kinematics as a black box (e.g. a Simulink [68] block) to build their application around it. These users will in general not be interested in adding functionalities to the given library, they must, however, be provided with means to choose the adequate inverse kinematics

algorithm or weight optimization criteria. The advanced user can adapt the software to work with specific kinematic structures, create new optimization criteria for redundancy resolution and embed the software into his own work. Furthermore, the advanced user provides the "black boxes" for the black box user.

3. Robot kinematics: Serial, tree, parallel, redundant or non-redundant kinematic structures necessitate different inverse kinematics algorithms.

4. Constraints, optimization criteria etc.: For each robotic system with redundant DoF, the constraints and optimization criteria vary and have to be adapted into the algorithm.

In the scope of this thesis, the software library *InvKin* was implemented according to the requirements mentioned above. The library thus is a step towards a universal inverse kinematics library. Namely, all mentioned application areas are covered, as well as all types of users and a variety of ways to include constraints and optimization criteria into the determination of the optimal robot pose. Concerning robot kinematics, serial and tree kinematics with or without redundancy are included. Parallel structures are not yet available, however it is known that inverse kinematics is usually straightforward for any parallel robot [72]. More precisely in most cases for parallel robots

- there is a unique solution to the inverse kinematics (in some cases provided that physical constraints are taken into account like for the Delta robot [16]), and
- each joint variable may be computed independently, being given the desired pose of the robot.

The integration of inverse kinematics algorithms for parallel structures into the *InvKin* library is therefore supposed to be easily feasible in future work[1].

One challenge of a universal inverse kinematics library is to include a variety of calculation methods, based e.g. on the inverse Jacobian, pseudo-inverse Jacobian, augmented Jacobi, or on non-linear equation solvers, and to provide means to include custom closed form solutions: In the case of closed form solutions, individual algorithms have to be conceived for different families of kinematic chains since a generic algorithm does not exist. Therefore, easy inclusion of additional, robot-specific algorithms should be possible.

Another important issue comes into play once redundant robotic structures are considered: For a desired TCP position, an infinite number of robot poses is adequate (referred to as the robot nullspace). The kinematics of the human body is an example of high redundancy; as a result, the same task (e.g. opening a bottle of water) can be accomplished in many different ways, depending on being tired, afraid, excited, angry, etc. Criteria to map the human way of using their nullspace to robotics are current matter

[1] It is however known that forward kinematics for parallel structures may be tricky to solve.

of research. It is therefore a key feature of the *InvKin* library to provide the user with the possibility of including custom made optimization criteria for nullspace optimization into the inverse kinematics algorithm.

Figure 5.2 shows the basic structure of the *InvKin* library. Two main base classes are implemented: `Invkin` and `Optimization`. Derived from `Invkin`, various algorithms for the calculation of inverse kinematics are implemented, such as closed form solutions, algorithms based on non-linear equation solvers (Levenberg-Marquardt [65] (LM) and SQP [29, 91]), and methods based on the inverse Jacobian resp. pseudo-inverse Jacobian. Criteria to optimize the pose of the robot inside the nullspace are derived from the base class `optimization`. The advanced user can add custom made algorithms for inverse kinematics calculation (e.g. robot-specific closed form solutions) as well as additional optimization criteria by simply deriving them from the base classes.

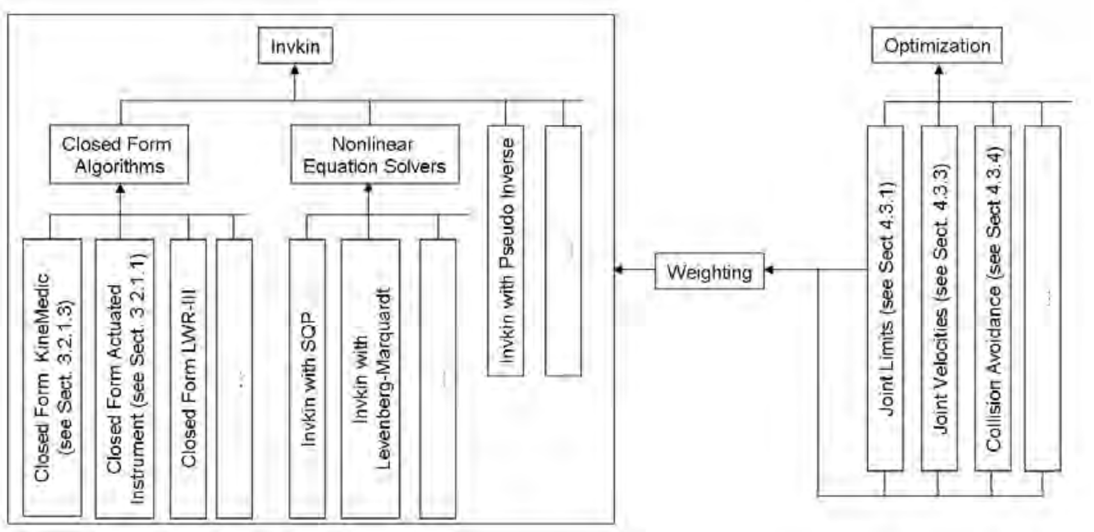

Invkin
Closed Form Algorithms
Nonlinear Equation Solvers
Invkin with Pseudo Inverse
Closed Form KineMedic (see Sect. 3.2.1.3)
Closed Form Actuated Instrument (see Sect. 3.2.1.1)
Closed Form LWR-III
Invkin with SQP
Invkin with Levenberg-Marquardt
Weighting
Optimization
Joint Limits (see Sect 4.3.1)
Joint Velocities (see Sect. 4.3.3)
Collision Avoidance (see Sect 4.3.4)

Figure 5.2.: Structure of the *InvKin* library.

Below, two use cases are presented: Use case (a) shows the performance of the library in planning tasks with the KineMedic robot. In use case (b), the library provides Cartesian control for the two handed experimental system Justin.

5.1. Use Case (a): Inverse Kinematics for Planning and Optimization

Real-time capabilities are usually not required in optimization and planning procedures for robotic workplaces, since the computations can be done offline. Calculation of the inverse kinematics should nonetheless be as fast as possible, since a large amount of robot poses has to be calculated during optimization and planning. Often one is interested in optimizing the setup of a workcell so that a robotic system can reach a certain workspace or trajectory in Cartesian space optimally with respect to given criteria and constraints. Inverse kinematics is then needed to calculate the joint angles $\mathbf{q}_i$ for a set of tool tip positions and orientations $\mathbf{x}_i$, representing the discretized workspace or trajectory. Solvability regardless of initial solutions is an important requirement of the inverse kinematics algorithm in this case.

The DLR medical robot is optimized for a variety of applications, including open and minimally invasive surgery. Various instruments will be attached to the robot tip such as laparoscopes, actuated instruments, or a scanner with a mirror joint arm that positions a laser beam to carry out osteotomies. To cover a wide range of kinematic structures, a universal interface to the inverse kinematics library is necessary (see next section). Different inverse kinematics algorithms are compared in terms of computational costs and robustness with respect to the initial solution in Sect. 5.1.2.

5.1.1. Implementation

The *InvKin* core is coded in C++ and made available as a dynamic library. It is tested under Linux with the GNU[2] compiler collection (gcc) (v3.3.4 and v4.1.2). Due to some of the linked libraries, a Windows version is not yet available.

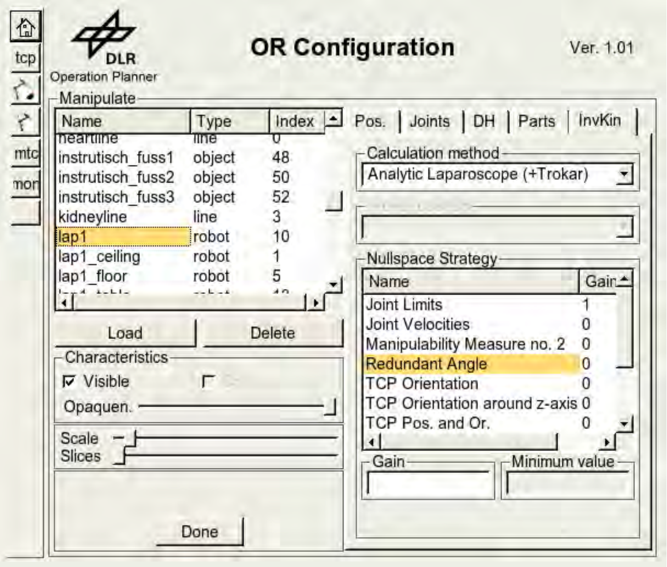
A common software structure to plan and optimize various tasks and robotic systems (as will be presented in Sect. 6) necessitates a method to automatically attribute suitable inverse kinematics algorithms to the resp. robotic structures. Both inverse kinematics algorithms and optimization criteria are therefore made available to the application through the design pattern of abstract factories [31]. According to the kinematic structure of the considered robot, all fitting inverse kinematics algorithms can be identified and displayed to the user (see Fig. 5.3), as well as all optimization criteria.

5.1.2. Evaluation

The following algorithms are available in the *InvKin* library for computation of the inverse kinematics of serial robotic structures with redundancy:

- Closed form algorithm 1: All solutions to the closed form inverse kinematics are optimized subsequently with respect to the redundant DoF, using an LM optimizer as provided in [65].
- Closed form algorithm 2: The best of all solutions of the closed form inverse kinematics is chosen and optimized with respect to the redundant DoF, using the LM algorithm.
- Closed form algorithm 3: Joint 3 is held constant and the closed form solution is used to calculate the remaining joint angles (i.e. the robot is considered as non-redundant).

[2] GNU is a computer operating system composed entirely of free software.



Name	Type	Index
heartline	line	0
instrutisch_fuss1	object	48
instrutisch_fuss2	object	50
instrutisch_fuss3	object	52
kidneyline	line	3
lap1	robot	10
lap1_ceiling	robot	1
lap1_floor	robot	5

☑ Visible ☐

Opaquen.

Calculation method: Analytic Laparoscope (+Trokar)

Nullspace Strategy

Name	Gain
Joint Limits	1
Joint Velocities	0
Manipulability Measure no. 2	0
Redundant Angle	0
TCP Orientation	0
TCP Orientation around z-axis	0
TCP Pos. and Or.	0

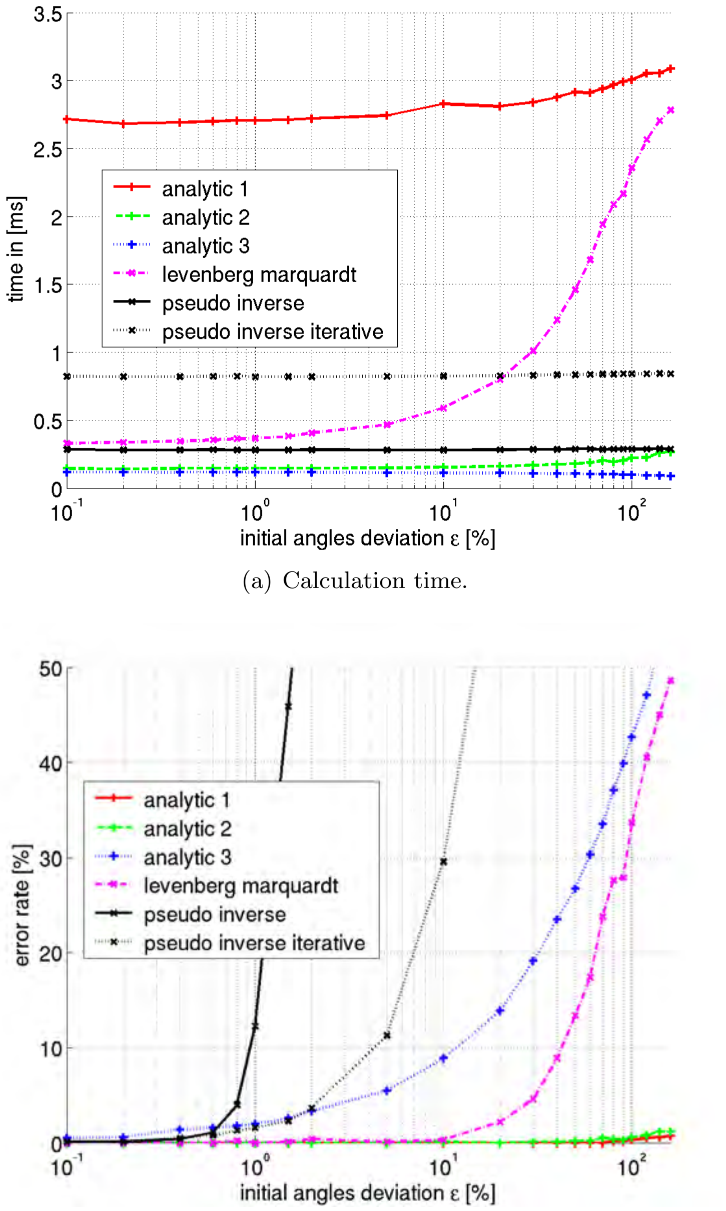
Figure 5.3.: For the robotic structure *lap1*, the user can choose the appropriate inverse kinematics algorithm (in this case: closed form solution of the inverse kinematics, including the trocar condition) and weight optimization criteria to determine the optimal pose in the nullspace.

- LM algorithm: All 7 joint angles are optimized using the LM algorithm without use of an closed form solution.
- Pseudo-inverse algorithm: in one step, the solution to the inverse kinematics is calculated using the pseudo-inverse Jacobian (and the forward kinematics to obtain an initial tool tip frame $\mathbf{x}_{\mathrm{init}_i}$): $\mathbf{q}_i = \mathbf{q}_{\mathrm{init}_i} + \mathbf{J}^+ (\mathbf{x}_i - \mathbf{x}_{\mathrm{init}_i})$
- Pseudo-inverse iterative algorithm: Five steps of the pseudo-inverse algorithm are performed subsequently.

A comparison of the calculation time and the error rate for the inverse kinematics solution of the 7 DoF KineMedic is shown in Fig. 5.4. Therefore, a set of 1000 reachable tool tip frames x_i is computed from randomly generated joint angle sets $\mathbf{q}_{\mathrm{valid}_i} \in [\mathbf{q}_{\min},\ \mathbf{q}_{\max}]$ using the forward kinematics. To evaluate the influence of the deviation of the initial solution from the sought solution, the deviation ε of the initial angles $\mathbf{q}_{\mathrm{init}_i}$ from the valid joint angles $\mathbf{q}_{\mathrm{valid}_i}$ is plotted onto the horizontal axes of Fig. 5.4, defined as follows:

$$\mathbf{q}_{\mathrm{init}_i} = \mathbf{q}_{\mathrm{valid}_i} + \mathsf{rand} \cdot \varepsilon \left(\mathbf{q}_{\max} - \mathbf{q}_{\min} \right). \tag{5.1}$$

The optimization criterion for the redundant DoF is the compliance with the joint limits. The mean computation time is measured using a standard computer (Intel Core$^{\mathrm{TM}}$ 2



(a) Calculation time.

(b) Error rate.

Figure 5.4.: Calculation time and error rate of different inverse kinematics procedures.

Duo Processor 1.86 GHz). The error rate expresses the number of wrong computations[3] with respect to all 1000 considered reachable tool tip frames.

[3] A computation is considered as wrong if the tool tip accuracy is worse than $\Delta x_{\min} = 0.1\,\text{mm}$ for translation or $\Delta\alpha_{\min} = 0.5^\circ$ for rotation.

It can be seen from Fig. 5.4a that the closed form algorithm 2 is fast compared to the other algorithms and that it provides very good results even if the initial joint angles are not known (see Fig. 5.4b)[4]. It is therefore preferable as inverse kinematics algorithm in workplace optimization tasks.

5.2. Use Case (b): Real-Time Inverse Kinematics for the DLR Robot Justin

This section presents the current implementation of real-time inverse kinematics for Cartesian control of the DLR robot Justin. Section 5.2.1 shows the kinematic structure of the robotic system. Details on the implementation of the inverse kinematics algorithm are given in Sect. 5.2.2, and an evaluation of some performance aspects is presented in Sect. 5.2.3.

5.2.1. Kinematic Structure

The DLR robot Justin (see Fig. 5.5) is a humanoid upper body system designed mainly for bimanual manipulation experiments in tasks coming e.g. from service robotics. From the kinematic perspective, the system consists of:

- A torso with 3 active and 1 passive DoF ($q_{1..4}$ in Fig. 5.5)),
- two LWR III arms with 7 DoF each ($q_{5..11}$ and $q_{12..18}$ in Fig. 5.5),
- two DLR hands with 12 DoF each, and
- a 3D-Modeller Head [99] mounted to a pan-tilt unit (two additional DoF).

All in all, the robotic system Justin has 43 active DoF.

5.2.2. Inverse Kinematics for Cartesian Control

The library *InvKin* is used to provide inverse kinematics for Cartesian control of the wrist frames ${}^{r}\mathbf{T}$ and ${}^{l}\mathbf{T}$ shown in Fig. 5.5 right. Therefore, in every time step, the following optimization problem is solved for the joint angles $\mathbf{q} \in \mathbb{R}^{18}$ (see Fig. 5.5 left):

$$\begin{aligned} f(\mathbf{q}) &\rightarrow \text{min}, & (5.2)\\ \text{subject to } {}^{r}\mathbf{T} &= g(q_{1..4}, q_{5..11}), & (5.3)\\ {}^{l}\mathbf{T} &= g(q_{1..4}, q_{12..18}). & (5.4) \end{aligned}$$

[4] Note, however, that large deviations ε lead to errors also in the case of the implemented closed form algorithms. This is due to the choice of the value for the nullspace parameter q_{null}. For the evaluation shown, the initial value for the nullspace parameter is chosen as $q_{\text{null}} = 0°$. In future implementations, the initial value for the nullspace parameter should be determined in an additional step so that the equations for the calculation of the inverse kinematics (see Sect. 3.2.1.2) remain solvable.

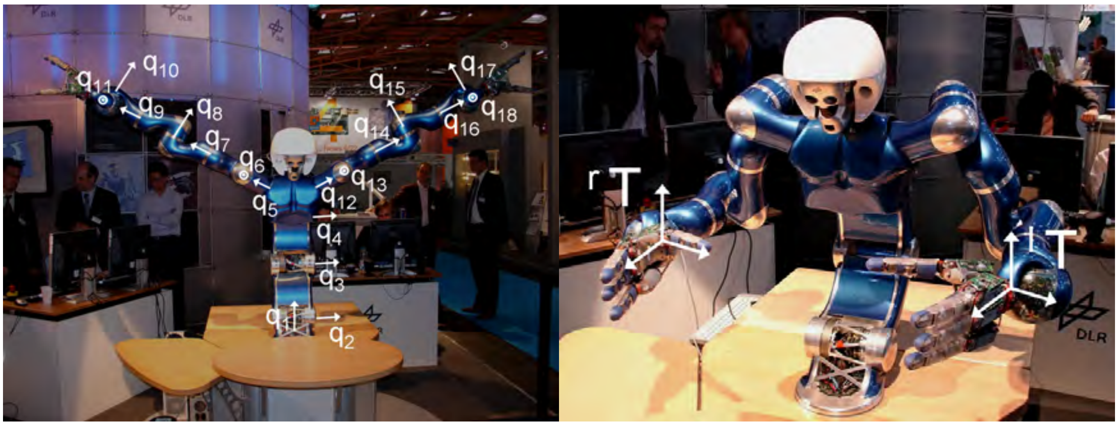


Figure 5.5.: The two-arm two-hand experimental system Justin. The joints $q_{1..18}$ relevant to the described inverse kinematics algorithm are shown left. The frames $^r\mathbf{T}$ and $^l\mathbf{T}$ (right) are subject to the Cartesian control.

Due to the branched kinematic structure, the joints of the torso $q_{1..4}$ appear both in Eq. 5.2 and 5.3. The passive joint q_4 is mechanically coupled through tendons with the joints q_2 and q_3 so that $q_4 = -q_2 - q_3$.

The library *InvKin* includes a class `InvkinBranched` that provides the possibility of connecting different kinematic branches and solves the inverse kinematics of the whole kinematic structure. For each branch, an adequate inverse kinematics algorithm can be chosen. Alternatively, the whole structure can be solved in one step.

The current implementation solves the whole 18 DoF kinematic structure using the known Levenberg-Marquardt algorithm [63, 67]. The following optimization criteria are considered:

- Reaching the desired frames of the left and right wrist is included in the optimization with high weighting factor and not as a constraint like shown in Eq. 5.2–5.4.
- Distance from joint limits and singularities as described in Sect. 4.3.1.
- Minimization of joint velocities as described in Sect. 4.3.3.

To speed up calculation, analytic solutions for the derivatives of all criteria are provided. Collision avoidance is not yet included. All available criteria can be chosen, weighted, and removed in real-time.

5.2.3. Evaluation

The described implementation of the inverse kinematics is used in various demonstrators for service robotics with the experimental system Justin (see e.g. Fig. 5.6). To enable black box use, S-functions[5] are provided as shown in Fig. 5.7: In the shown example, the Spacemouse[6] module continually integrates velocities commanded by the user (via two space mice for right and left wrist, respectively) and thus provides the current goal frames of the left and right arm to the inverse kinematics S-function. A variety of S-functions is available in any Simulink model by drag and drop from the *InvKin* Simulink library (see Fig. 5.8).

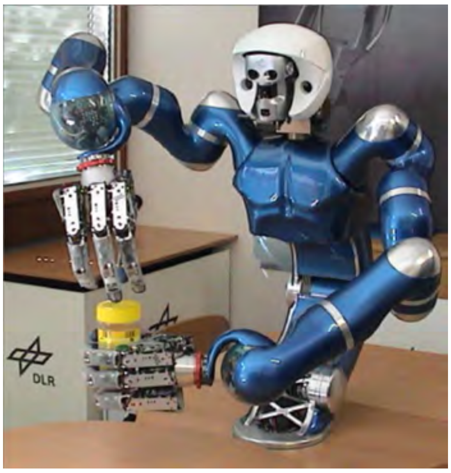


Figure 5.6.: The inverse kinematics is used to calculate the paths in tasks such as serving tea.

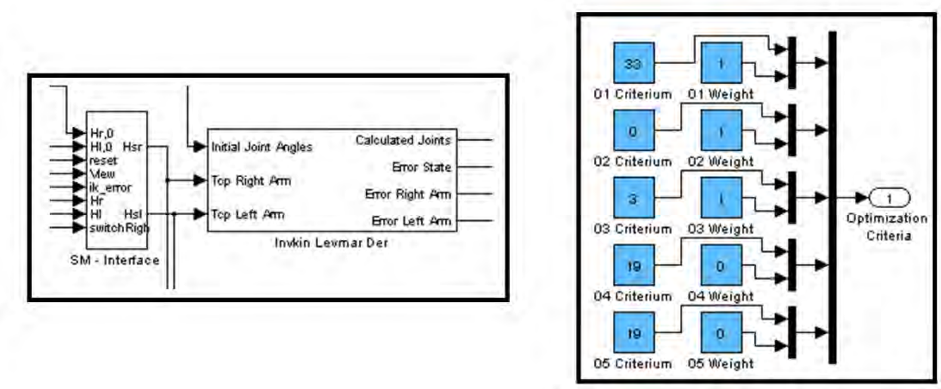


Figure 5.7.: Simulink model of the inverse kinematics for the robotic system Justin: The Simulink S-function `InvkinLevmarDer` (left) calculates the joint angles according to the given wrist frames and the initial joint angles. Optimization criteria, addressed by their ID, can be weighted and changed in every time step (right).

[5] An S-function is a computer language description of a Simulink block. S-functions can be written e.g. in C, C++, Ada, or FORTRAN and allow to add custom made blocks to Simulink models [68].

[6] See http://www.3dconnexion.de/. Accessed on April 22th, 2007.

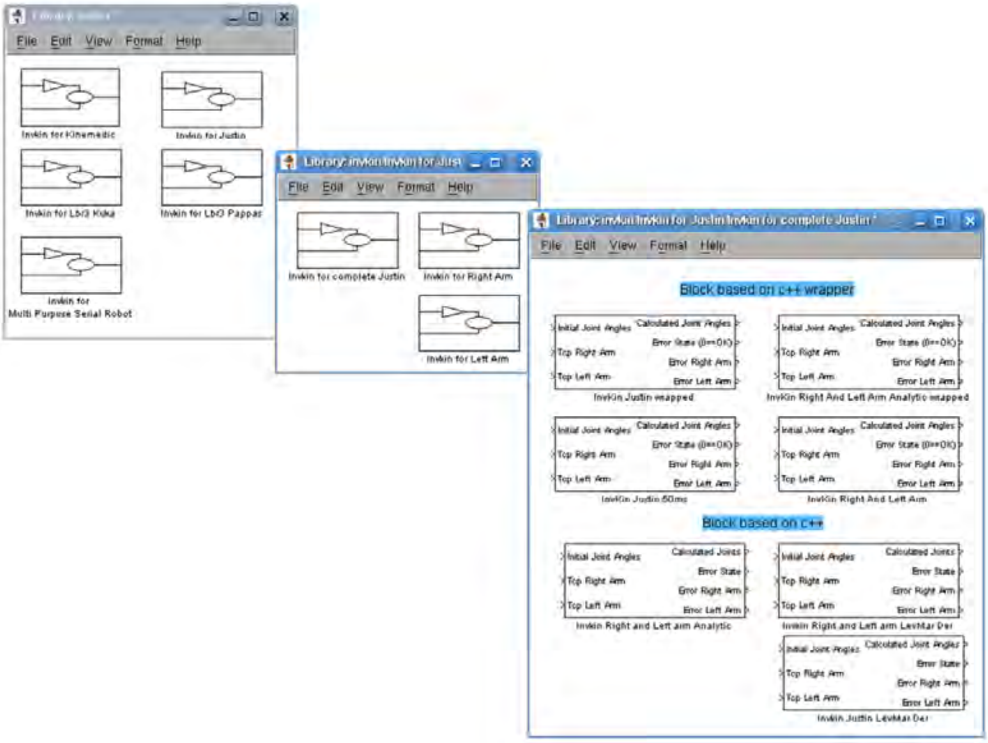


Figure 5.8.: The Simulink *InvKin* library.

In the Simulink model for the robotic system Justin, the calculation of the inverse kinematics runs at a rate of 200 Hz as opposed to the other parts of the Simulink model running at 1 kHz. Computation times for the inverse kinematics vary between 200 ms and 600 ms (worst case). See Fig. 5.9 for a snapshot of the system profiler on the QNX[7] computer during a Cartesian point to point motion.

[7] QNX is a commercial Unix-like real-time operating system.

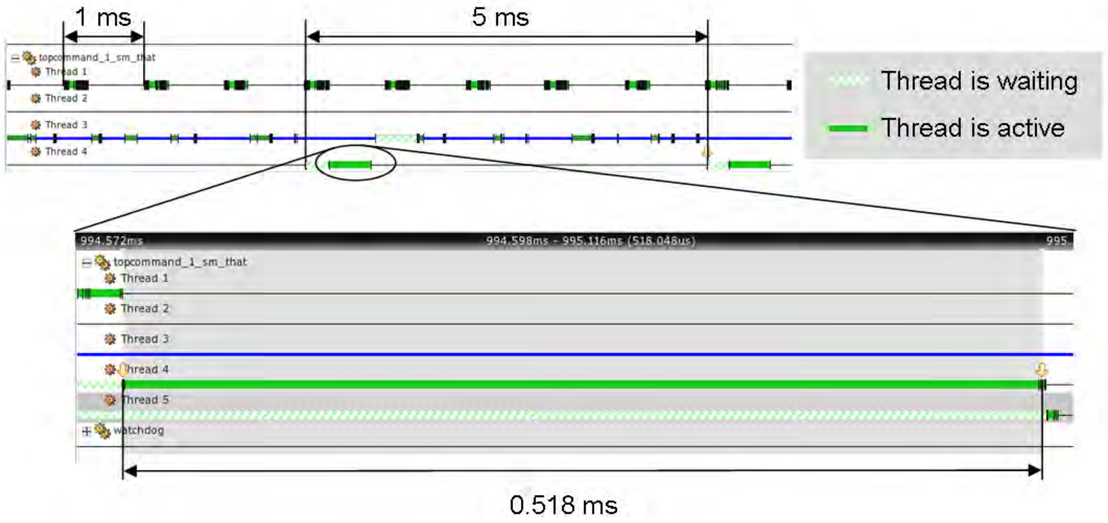


Figure 5.9.: Snapshot of the QNX system profiler: The control model has a sample rate of 1 kHz (Thread 1). The inverse kinematics algorithm (Thread 4) runs at 200 Hz due to computational limitations. In the shown cycle, inverse kinematics calculation takes 0.518 ms.

5.3. Summary and Conclusion

The *InvKin* toolbox provides basic functionality for the calculation of inverse position kinematics with high versatility in terms of (1) the application area, (2) the type of user, (c) the robot kinematics, and (4) the constraints resp. optimization criteria. This is shown by the two use cases presented above.

Future directions to extend the library could be to:

- Provide a Windows version of the library,
- include inverse velocity kinematics, and
- include both forward and inverse position kinematics algorithms for parallel structures.

The algorithms were tested mainly on the robotic systems available at DLR, namely the medical robot KineMedic, the LWR II and III, and the experimental system Justin. Tests on standard industrial robots remain to be done in the future.

In the current implementation of the inverse kinematics of Justin, it is not possible to include constraints in any other way than through high weighting in the quality function. Furthermore, available closed form solutions of branches (e.g. of the arms) are not exploited. It is therefore envisioned to implement a new version based on the SQP algorithm and including all closed form solutions. Furthermore, collision avoidance will be dealt with, too.

6. The Workplace Optimizer Wizard

The Workplace Optimizer Wizard is a tool to determine positions of robots and/or robot kinematics inside a given workplace in order to fulfill a task optimally according to a range of criteria. These include e.g. reachability, collision avoidance and accuracy. In this section, the functionality of the software is presented through two use cases, (a) the kinematic design optimization for the DLR medical robot KineMedic and for an actuated ceiling mounted carrier for up to three KineMedic robots (Sect. 6.1) and (b) the preoperative planning for robotically assisted minimally invasive interventions (Sect. 6.2). Sect. 6.3 concludes the chapter with a short summary.

6.1. Use Case (a): Kinematic Design Optimization for the KineMedic System

The OR is a remarkably ill-suited environment for robots. It is poorly structured, since the position of OR equipment is often changed during the intervention, the staff is moving and the patient itself is constantly in motion e.g. due to heartbeat and breathing. Naturally the safety requirements by far exceed those in industrial applications, where human interaction can be excluded in most cases. Furthermore, the surgical staff usually has no expertise in robotics.

However, the potential advantage of combining skills of man with those of robots (see e.g. [81] and Table 1.1 in Sect. 1.1) has led to a variety of propositions for robotic systems in the OR [101]. Especially the field of MIRS has proven to be a rewarding application area for robots, with about 550 daVinci$^{\text{TM}}$ systems of Intuitive Surgical installed by the end of the year 2006[1] and offering features such as e.g. motion scaling, tremor filtering, or multi robot control.

This section presents the optimization of the design of the DLR medical robot KineMedic to accomplish minimally invasive surgery. In a second step, a design optimization for an actuated carrier to position up to three KineMedic robots is carried out.

6.1.1. Introduction

The diversity of surgical interventions carried out or proposed to be carried out with robotic assistance increases constantly. In this context, the optimal design of a robot with respect to its envisioned surgical interventions is of special interest.

[1] See http://www.intuitivesurgical.com/. Accessed on June 14th, 2007.

For an advanced robotic system for thoracic surgery, the following questions are important:

1. Fields of application: What surgical interventions shall be carried out with the robot and how can these interventions be formulated mathematically?

2. Optimization criteria and constraints: Given different design alternatives, which of them is the best?

3. Optimization method: Which optimization method is suited best to solve the optimization problem?

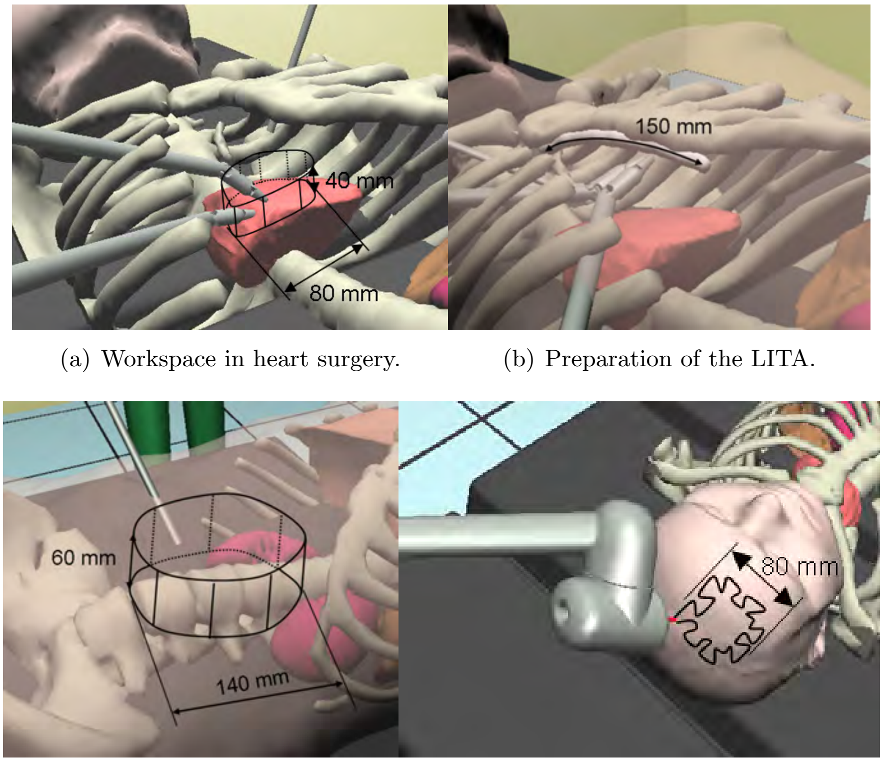
Surgical interventions are analyzed in terms of workspace and accuracy requirements in the next section. Details on the used optimization criteria and constraints, as well as on the optimization method are given in Sect. 6.1.3 for the design optimization of the KineMedic and in Sect. 6.1.4 for the design optimization of the actuated carrier.

6.1.2. Operation Scenarios

In the following, the considered operation scenarios are described. The scenarios are created using segmented computer tomography (CT) data of real patients and correspond with earlier examinations of typical workspaces in surgical robotics [56]. Every workspace is varied both in size and distance to the skin surface to take into account three (significantly different) patient geometries. Acquisition of a larger patient database is desirable as future work (bearing in mind that coverage of the whole diversity in human anatomy is naturally not possible). The following operation scenarios are considered:

Operation on the heart These operations comprise cardiac interventions such as totally endoscopic bypass graft, mitral/aortic valve repair/replacement and tricuspid valve repair. A workspace as shown in Fig. 6.1a is assumed. Two robots carry articulated instruments and a third robot carries a stereo endoscope. Inside the workspace, high dexterity and manipulability has to be guaranteed. Especially, requirements from the beating heart compensation as suggested in [77] have to be met.

Preparation of the left internal thoracic arteria (LITA) The medical robots are equipped with similar instruments as in the previous scenario, the workspace is shown in Fig. 6.1b. The preparation of the left internal thoracic arteria represents a special category of tasks since the workspace is very spacious in one direction and located above the entry points into the human body.



(a) Workspace in heart surgery.

(b) Preparation of the LITA.

(c) Abdominal exploration.

(d) Workspace in open surgery.

Figure 6.1.: Workspaces of the considered operation scenarios.

Abdominal interventions For abdominal interventions, cholecystectomy, appendectomy, hernia repair, and laparoscopy are considered. Especially abdominal exploration requires a very large workspace. In this scenario (Fig. 6.1c), only the robot carrying the endoscope is involved, being restrained by the trocar condition. The robot has to position the endoscope throughout a large workspace to enable investigation of the whole abdominal cavity. Since the endoscope is not actuated inside the patient, orientations are not taken into consideration.

Open surgery Additionally to [56], a workspace for open surgery, namely a cutting task on the human skull, is considered (see Fig. 6.1d). Applications in e.g. orthopedics can be performed by the system if the medical arm is equipped with a tailored cutting tool suitable for cutting bone. The robot is used to position a scanner that deflects a laser beam in such an extend that the laser performs a preoperatively planned cut into bony structure. A prototypic mock-up is shown in Fig. 6.2, right. Advantages of this method compared to standard osteotomy techniques like oscillating saws are (a) nearly arbitrary cutting trajectories (see Fig. 6.2, left) and (b) very thin cuts, both leading to faster recovery. Since the cutting is performed contact-free, a light-weight and flexible robot such as the KineMedic is suitable for the task.

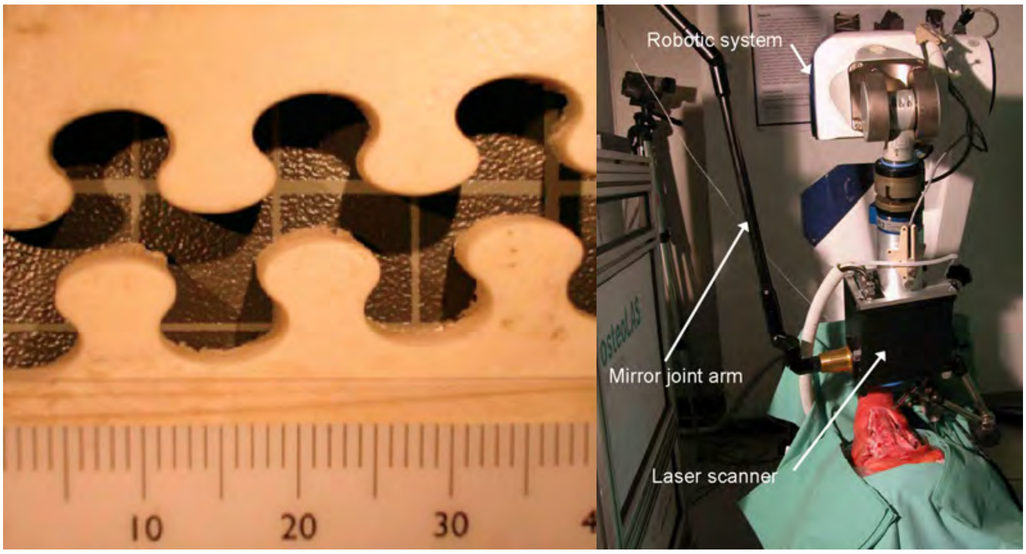


Figure 6.2.: First trials of laser osteotomy: A self-stabilizing laser incision in compact bone tissue (left, ex vivo) [42] and a robotic system to position the laser scanner (right) [110]. A mirror joint arm is used to deflect the laser beam towards the scanner.

Inside the workspaces, certain quality measures are evaluated, to ensure e.g. that the tool tips of the robots can move with at least 60 mm/s and 30°/s as needed for beating heart compensation. Concerning the accuracy that is necessary to perform the considered tasks, the anastomosis was identified to be most demanding. The stitches have a mean distance of $d = 0.5$ mm to each other. Thus, a necessary translational positioning accuracy of $\Delta x_{\mathrm{min}} = 0.1$ mm is assumed. This value corresponds to the positioning accuracy that a surgeon with visual feedback can achieve as measured in [27, 40, 86, 97]. No publications are known to the author that analyze rotational accuracies or appearing velocities in surgical interventions. In this thesis, a rotational accuracy of $\Delta\alpha_{\mathrm{min}} = 0.5^\circ$ is assumed (see Table 6.1 for these constraints and the parameters predetermined from design).

Every workspace $j \in \{1, ..., n_{\mathrm{A}}\}$, with $n_{\mathrm{A}} = 4$ the number of considered workspaces, is discretized into a set $\mathcal{A}_j$ of up to 100 tool tip frames $A_{j,p}$ to evaluate the quality measures throughout the workspace. The orientation of these frames is chosen in a way that the z-axis is pointing opposite to the surface normal of the nearest point on the respective organ, i.e. the instrument tip is perpendicular to the organ surface. In the following, the configuration $\mathbf{q}_{r,j,p}$ denotes the calculated joint angles of a robot r to reach a tool tip frame $A_{j,p}$.

For every design alternative i, an optimization parameter vector is defined:

$$\mathbf{p}_i^{\mathrm{T}} = \left(\mathbf{p}^{\mathrm{d}^{\mathrm{T}}}, \mathbf{p}_1^{\mathrm{o}^{\mathrm{T}}}, ..., \mathbf{p}_j^{\mathrm{o}^{\mathrm{T}}}, ..., \mathbf{p}_{n_{\mathrm{A}}}^{\mathrm{o}\ \mathrm{T}}\right), \tag{6.1}$$

Table 6.1.: Compulsory performance requirements and design parameters

Parameter	Value	cf. Sect.
Distance from joint limits/singularities	3°	4.3.1 and 4.3.2
Maximum joint velocity $\dot{\Theta}_{\mathrm{MAX}}$	$225^\circ\,\mathrm{s}^{-1}$	4.3.3
Translational velocity $\dot{x}_{\mathrm{min}}$	$60\,\frac{\mathrm{mm}}{\mathrm{s}}$	4.3.7
Rotational velocity $\dot{\alpha}_{\mathrm{min}}$	$30^\circ\,\mathrm{s}^{-1}$	4.3.7
Resolution of decoders $\Delta\Theta_{\mathrm{min}}$	0.001°	4.3.8
Translational accuracy Δx_{min}	0.1 mm	4.3.8
Rotational accuracy $\Delta\alpha_{\mathrm{min}}$	0.5°	4.3.8
Tool tip dexterity	$\pm 20^\circ$	4.3.5

and two types of optimization parameters are distinguished:

- Design parameters $\mathbf{p}^{\mathrm{d}}$ that are intrinsic to the design and cannot be modified once the mechanism is in use, such as segment lengths and the assembly of the actuators.
- Operation parameters $\mathbf{p}^{\mathrm{o}}_{j}$, that can be modified for every workspace j, according to a preoperative planning-step, like the position of the robot base or, in MIRS, of the entry points into the human body.

The parameters depicted in Fig. 6.3 and 6.7 can be either design or operation parameters, depending on the resp. design alternative. As an example, the instrument length p_5 could be either fixed (design parameter) or adjustable during preparation of the surgical procedure (operation parameter).

6.1.3. Design Optimization of the DLR medical robot KineMedic

Amongst others, the following studies have been performed to support the design process of the medical robot KineMedic (see Fig. 6.3):

- Optimization of the instrument length p_6 and determination whether the instrument length p_6 should be adjustable or fixed. Since the instrument is constrained in two directions at the entry point, its functionality can be compared with a lever that transfers motions inside the patient body, thereby scaling the robot motion outside of the patient with respect to motion inside the patient. The length of the instrument p_6 therefore has a strong influence on the robot performance and is analyzed in the following.
- Optimization of the last segment length p_7 of the actuated instrument. A force torque sensor is integrated into the last part of the instrument (see Fig. 6.3). Therefore, the segment length p_7 cannot be reduced to zero, as would be desirable to decouple translations and rotations at the tool tip. A feasible segment length with sufficient performance has to be determined.

- Minimization of the link lengths p_4 and p_5 to obtain a light-weight and slender design. Minimal link lengths p_4 and p_5 as depicted in Fig. 6.3 are desirable since a compact design is important in an overcrowded environment such as the operating room.

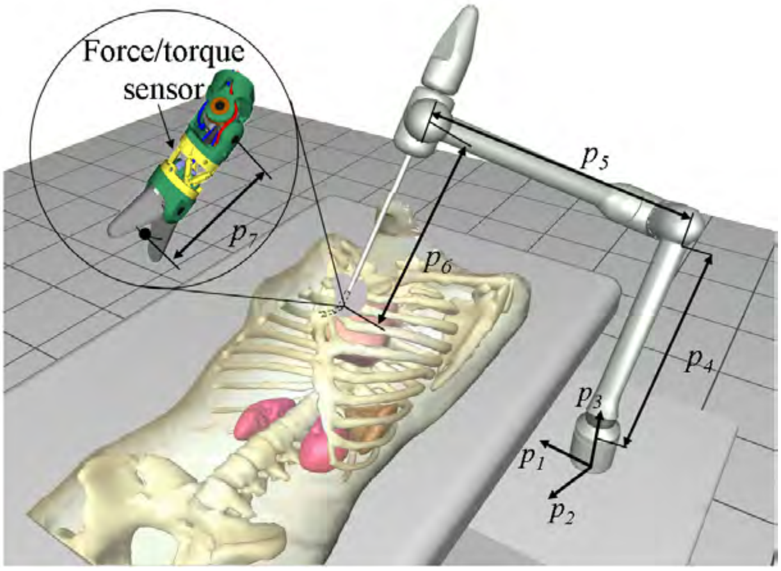


Figure 6.3.: Surgical robot and actuated instrument. The parameters $p_{1...3}$ denote translations of the base position, while the parameters $p_{4...7}$ denote segment lengths of the robot and instrument.

The optimization studies are carried out according to the method shown in Sect. 4.1.2: The considered surgical interventions are analyzed in terms of workspace and demanded positioning accuracy (see Sect. 6.1.2 above), the optimization criterion and the constraints are defined (Sect. 6.1.3.1), the optimization parameters are decided on and a suitable optimization method is chosen (Sect. 6.1.3.2) and applied.

6.1.3.1. Optimization Criteria and Constraints

Defining suitable optimization criteria and constraints is probably the most crucial factor in the optimization procedure. Among the many potential optimization criteria respectively constraints, the following are selected:

- Avoidance of joint limits and singularities (see Sect. 4.3.1 and 4.3.2)
- Manipulability (see Sect. 4.3.7)
- Accuracy (see Sect. 4.3.8)

- Dexterity (see Sect. 4.3.5 and below)
- Collision avoidance (see Sect. 4.3.4)
- Geometrical considerations (see below)
- Robustness of the robot setup (see below)
- Overall size of the robot (see below)

As described in Sect. 4.2, one way to handle multiple optimization criteria is to use weighting factors and thus to combine the different criteria into a scalar performance index. This however induces the problem of finding suitable weighting factors. Another possible way is to determine the pareto-optimal front [22]. This method works fine in case of two optimization criteria, however with three or more criteria it becomes very complex, both in terms of analysis of the results and computational costs.

For the design optimization of the KineMedic robot, another approach is chosen: The sole optimization criterion is the minimization of the overall link length $p_4 + p_5$ and thus the minimization of the overall size of the robot. This is of great interest in an unpredictable and overcrowded environment such as the operating room. Moreover, in emergency situations the robot has to be removed manually, therefore easy manageability (light-weight construction) is important. Manipulability, accuracy, and geometrical considerations are added to the optimization problem in form of constraints. Furthermore, robustness checks as well as dexterity constraints are included. Thus the optimal setup in the chosen modeling is the one with minimal overall size that still meets all mentioned constraints. Besides manipulability and accuracy constraints, there are two geometrical constraints that have to be met due to design reasons. The first one addresses the length of the instrument segment remaining outside of the body $p_{6,\text{ outside}}$ as depicted in Fig. 6.4:

$$p_{6,\text{ outside}} \geq 100 \text{ mm}\,. \tag{6.2}$$

The second geometrical constraint, also depicted in Fig. 6.4, restricts the distance of the robot base from the edge of the operating table:

$$l_{\text{table}} \geq 70 \text{ mm}\,. \tag{6.3}$$

Robustness With Respect to Inaccurate Registration and Positioning

Registration of the patient is – as a matter of principle – inaccurate to a certain degree. This is because of e.g. soft tissue deformation and measurement errors. Furthermore, also positioning of the robots during preparation of the intervention is inaccurate (depending e.g. on the used navigation system). Therefore, it has to be assured that the sought-after, optimized setup is robust with respect to small changes in the setup. If

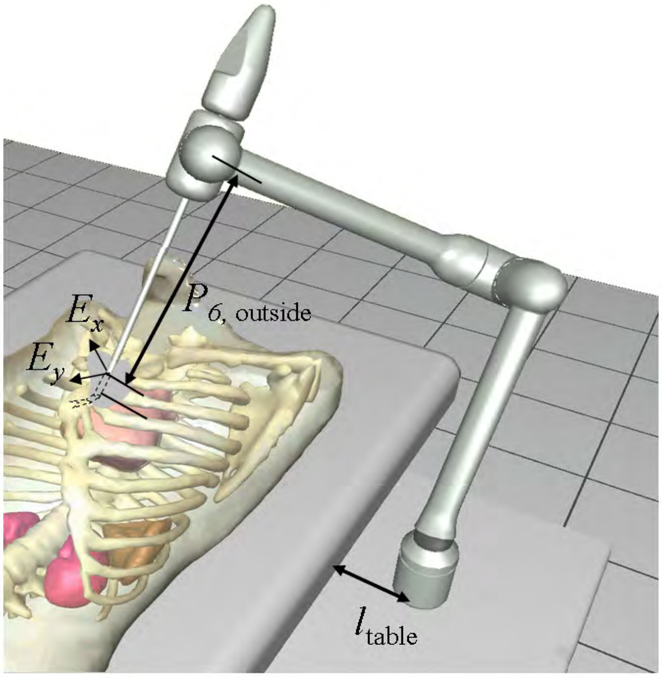


Figure 6.4.: Geometrical constraints $p_{6,\text{ outside}}$ and l_{table} and entry point coordinates $E_{x,y}$.

for example the entry point is shifted some centimeters, the robot still has to be able to accomplish the considered task. In order to provide robust solutions, variations of those parameters that had been found in a preliminary study to be critical are included in the optimization (see Fig. 6.4 for E_x):

- Operation on the heart: shift of the entry point $E_\text{x} = -30\,\text{mm}$ and $E_\text{x}= 30\,\text{mm}$ from nominal position.
- LIMA takedown workspace: shift of the entry point $E_\text{x} = -30\,\text{mm}$ from nominal position.
- Workspace of abdominal exploration: shift of the entry point $E_\text{x} = -20\,\text{mm}$ and $E_\text{x} = 20\,\text{mm}$ from nominal position.

This is done by considering the constraints not only for the nominal entry point position, but also for the above mentioned variations[2].

[2] The optimization procedure is thus carried out also for $E_\text{x} = -30\,\text{mm}$ and $E_\text{x}= 30\,\text{mm}$ in case of the operation on the heart, for $E_\text{x} = -30\,\text{mm}$ in case of the LIMA takedown workspace, and for $E_\text{x} = -20\,\text{mm}$ and $E_\text{x} = 20\,\text{mm}$ in case of the workspace of abdominal exploration. The worst results are used as quality measure for the considered design alternative. Note that it is not possible to take into account a complete discretization of possible variations of all parameters due to the limitation of available computing power. However, the considered variations provide, according to experience, good results for robustness.

Dexterity

Within each workspace (and its variations for robustness checks), a robot setup is evaluated at approximately 100 positions. Furthermore, in case of actuated instruments, 7 different orientations of the instrument tool are considered at each position to check for dexterity: Besides the straight posture, the instrument is angled at 30° and rotated in 60°-steps around the axis defined by the entry point and the goal position (see Fig. 6.5). Note that due to the non-negligible length of the last segment of the instrument, the dexterity measure presented in Sect. 4.3.5 has to be evaluated using an approach based on sampling.

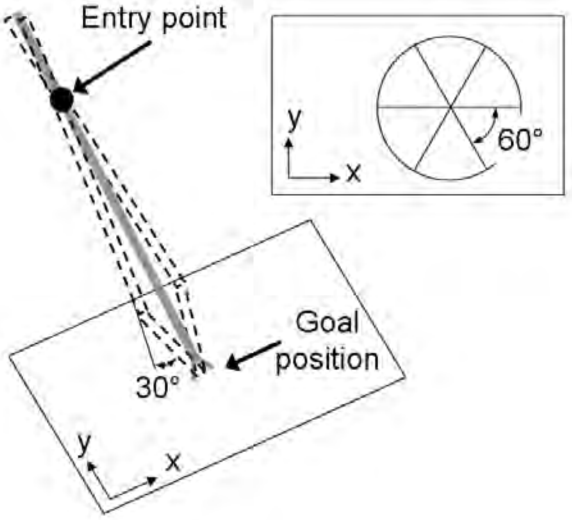


Figure 6.5.: Orientations of the instrument tool that are included into the optimization procedure.

6.1.3.2. Optimization Results

Optimization is carried out using Genetic Algorithms with a subsequent gradient-based method. Genetic Algorithms are very suitable since the constraints are discontinuous. Furthermore, Genetic Algorithms allow for parallel evaluation, and the use of clusters is therefore facilitated. Once a setup is found that meets the constraints, gradient-based methods are faster. The optimization is computationally quite expensive: On a cluster with 10 standard personal computers (Pentium 4, 2.4 GHz), the complete optimization cycle takes about one hour. Since however the analysis is not time-critical, high computational costs are no problem. Nevertheless, possibilities to reduce computational costs will be further analyzed in the future. This is especially of interest in more time-critical applications, as e.g. the workplace optimization described in Sect. 6.2.

Table 6.2 shows the optimization parameters considered in three different optimization studies: The first study considers the instrument length as design parameter (to be optimized), whereas the instrument length can be adjusted for each operation in the second study. The position of the robot base is adjustable in all optimization studies[3]. Results show that it would be advantageous to offer length-adjustable instruments. Since the complexity would increase however considerably due to the adjustment mechanism, a feasible solution would be to design instruments with different lengths that can be chosen according to the suggestions from preoperative planning. The lengths of the instrument should be in the range of

$$p_6 \in [210\,\text{mm}, ..., 330\,\text{mm}]\,.$$

Thus the following optimal link lengths for the robot are found:

$$p_4 = 309.2\,\text{mm} \quad \text{and} \quad p_5 = 384.8\ \text{mm}\,. \tag{6.4}$$

The third optimization study serves to determine the influence of the instrument length p_7 on the overall robot size $p_4 + p_5$. With the current design of the force torque sensor, the segment length cannot be decreased below $p_{7,\ \text{min}} = 20\,\text{mm}$. As shown in Table 6.3, smaller values of p_7 do not allow for a more compact robot design. However, with increased instrument length ($p_7 \geq 30\,\text{mm}$), the optimization constraints are violated. Therefore, a segment length of $p_7 = 20\,\text{mm}$ is suggested.

Table 6.2.: Considered optimization parameters and results

i	$p_h^\text{d}, h =$	$p_h^\text{o}, h =$	$p_4 + p_5$ [mm]
1	$\{4, ..., 6\}$	$\{1, ..., 3\}$	749, $p_6 = 330\,\text{mm}$
2	$\{4, 5\}$	$\{1, ..., 3, 6\}$	694, $p_6 \in [210\,\text{mm}, ..., 330\,\text{mm}]$
3	$\{4, ..., 6\}$	$\{1, ..., 3\}$	see Table 6.3

Table 6.3.: Influence of the instrument length p_7 on the overall robot size $p_4 + p_5$

p_7 [mm]	$p_4 + p_5$ [mm]
0	699
6.5	681
10	700
20	694
30	720
40	n/a
50	n/a

[3] Note that for the design optimization of the KineMedic robot it is supposed to be fixed to the floor with the robot structure upright. The robot can however be mounted in any angle as shown e.g. in Sect. 6.1.4 with the robot mounted to the ceiling.

Figure 6.6 shows an analysis of robustness of the optimized setup for the operation on the heart. In case of variations of the entry point (E_x and E_y in Fig. 6.6), a circle is drawn into the diagram to illustrate that variations of the entry point of 30 mm are tolerated by the optimized setup. The setup is also tolerant with respect to variations of the robot base (p_1 and p_3 in Fig. 6.6). Results for the other workspaces are conformable. Thus sufficient robustness to compensate for registration and positioning inaccuracies can be stated.

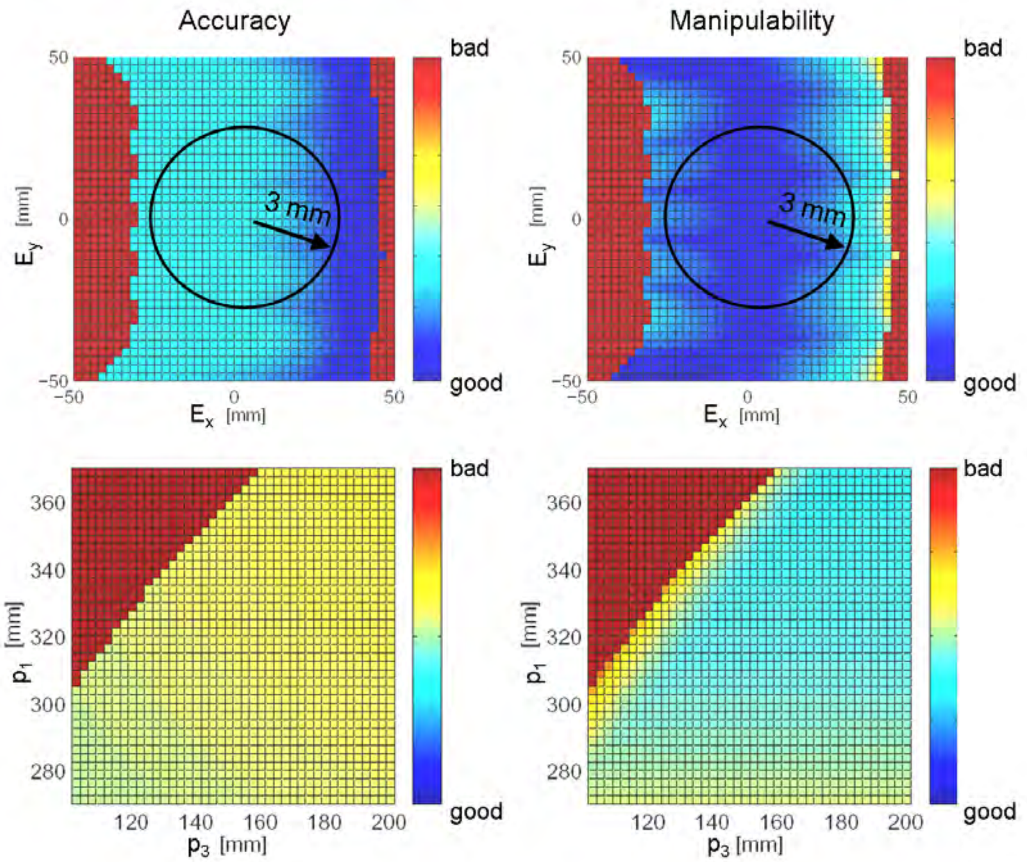


Figure 6.6.: Analysis of robustness in case of the operation on the heart for the found robot setup.

6.1.4. Kinematic Design Optimization of an Actuated Carrier for the DLR Multi-Arm Surgical System

In the following, a generic approach to optimize the design of an actuated carrier for the DLR multi-arm surgical system is presented according to [49]. The carrier is attached to the ceiling of the operating room and provides additional degrees of freedom to the surgical robots, with the purpose of automatic, optimal positioning of their bases as well as guaranteeing high stiffness. The minimum necessary degrees of freedom of the carrier as well as the optimal segment dimensions are obtained by use of an optimization with Genetic Algorithms.

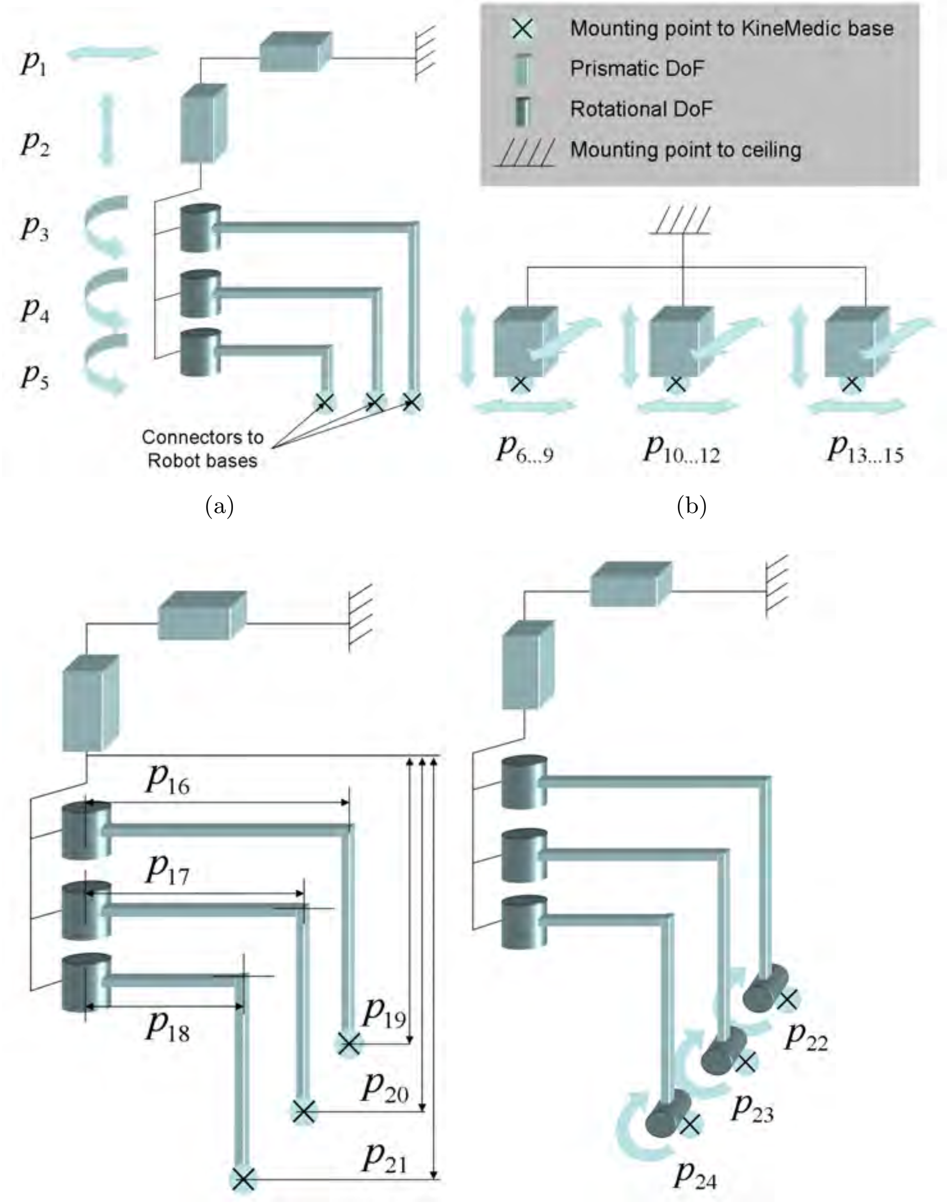
A good positioning of the robotic system relative to the patient is crucial for a successful intervention to e.g. ensure that the volume of the operating field coincides with the (confined) robot workspace. Current systems however hardly provide any assistance in this step, and the OR staff is mainly supposed to position the OR equipment using visual judgment and experience. Approaches to optimally position e.g. the daVinci$^{\text{TM}}$ system are reported in [18] to be rather cumbersome. The following sections therefore describe a systematic approach to determine a kinematically optimal design for an actuated carrier to automatically position up to three KineMedic robots in interventions covering both (telemanipulated) minimally invasive, robotic surgery and operations in open surgery.

The next Section 6.1.4.1 presents the chosen procedure to obtain an optimized design for the positioning device. Some favored design alternatives are shown in Sect. 6.1.4.2.

6.1.4.1. Problem Statement

The actuation and mounting setup should be able to deal with various operation scenarios as shown in Sect. 6.1.2. The presented design optimization incorporates both a structure synthesis to determine the general arrangement of the mechanical structure (type and number of DoF), as well as a dimensional synthesis to find optimal link lengths and joint locations[4]. To allow for free positioning of each robot base, a total of 18 DoF would be necessary. It would, however, be desirable to achieve a design with less actuations to reduce complexity, costs, weight, and size of the carrier. Therefore, the original design idea as depicted in Fig. 6.7a only disposes of 5 actuations. One goal of this study is to analyze whether 5 actuations are sufficient to ensure optimal positioning of the robot bases. Figure 6.7 shows all considered parameters. These parameters are combined with each other (see Sect. 6.1.4.2) to generate the considered design alternatives.

[4] A classification for optimal robot design problems is given in [73].



(a)

(b)

(c)

(d)

Figure 6.7.: Used parameters p_h in different design alternatives.

6.1.4.2. Optimization Results

The underlying idea to the current study is the design of an actuated carrier for up to three KineMedics, mounted to the ceiling. Naturally it is not possible to evaluate every possible kinematic chain that enables the positioning of three robots in the OR. However, certain design alternatives as depicted in Table 6.4 were optimized and compared with each other. The design alternatives are obtained by combining the parameters p_h as depicted in Fig. 6.7. The optimization function $f(\mathbf{p}_i)$ includes the criteria shown in Fig. 6.8 as described in Sect. 4.2.

Table 6.4.: Considered design alternatives and optimization results $f(\mathbf{p}_i)$

i	$p_h^{\mathrm{d}}, h =$	$p_h^{\mathrm{o}}, h =$	$f(\mathbf{p}_i)$
1	-	$\{1,...,5\}$	0.000262993
1^*	-	$\{1,...,5\}$	0.000261241
2	$\{22,...,24\}$	$\{1,...,5\}$	0.000262801
2^*	$\{22,...,24\}$	$\{1,...,5\}$	0.000261203
3	-	$\{6,...,15\}$	0.000261962
3^*	-	$\{6,...,15\}$	0.000261041
4	$\{16,..,24\}$	$\{1,...,5\}$	0.000262793
4^*	$\{16,..,24\}$	$\{1,...,5\}$	0.000260955
5^*	$\{3,4,5,16,..,24\}$	$\{1,2\}$	0.000371943
6^*	$\{5,16,..,24\}$	$\{1,...,4\}$	0.000298312
7^*	$\{19,..,24\}$	$\{1,...,5,16,..,18\}$	0.000260928
8^*	$\{16,..,21\}$	$\{1,...,5,22,..,24\}$	0.000260877
9	$\{16,..,21\}$	$\{1,...,5\}$	0.000262799

Table 6.5.: Suggested design parameters

Design parameter	Values
p_{16}^{d}	471 mm
p_{17}^{d}	346 mm
p_{18}^{d}	224 mm
p_{19}^{d}	7 mm
p_{20}^{d}	78 mm
p_{21}^{d}	148 mm

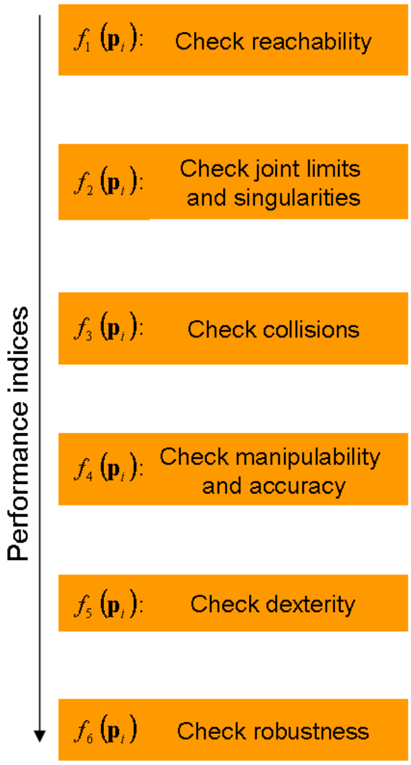

Performance indices
$f_1(\mathbf{p}_i)$: Check reachability
$f_2(\mathbf{p}_i)$: Check joint limits and singularities
$f_3(\mathbf{p}_i)$: Check collisions
$f_4(\mathbf{p}_i)$: Check manipulability and accuracy
$f_5(\mathbf{p}_i)$: Check dexterity
$f_6(\mathbf{p}_i)$ Check robustness

Figure 6.8.: Considered optimization criteria for the design optimization of the actuated carrier.

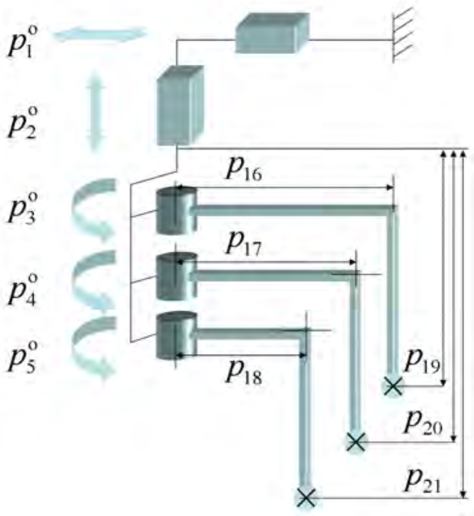


Figure 6.9.: Kinematic design of the suggested articulated carrier.

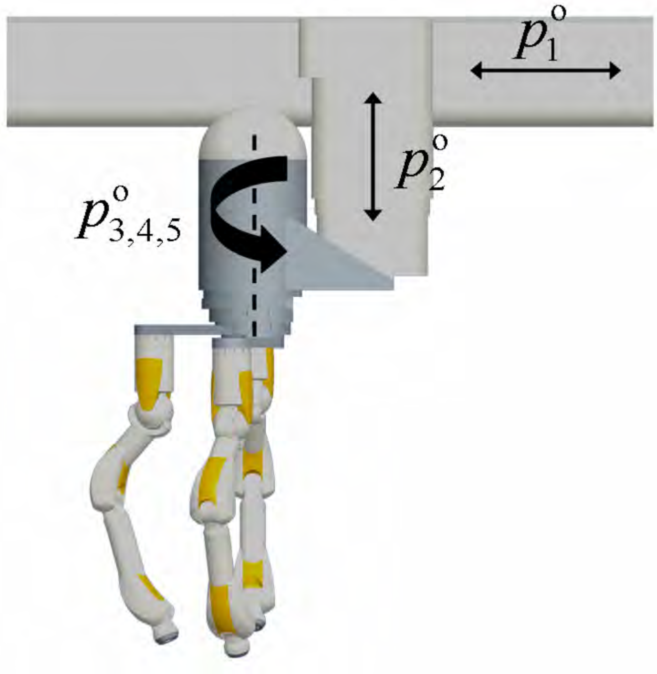


Figure 6.10.: Overview of the DLR medical operating scenario with the optimized actuated carrier.

The following results can be summarized:

- All design alternatives except for alternatives 5* and 6* yielded configurations that fulfill the requirements summarized in Table 6.1. Variation of the evaluation values is rather small, probably due to concurrence between certain optimization criteria such as e.g. manipulability and accuracy.

- The design of the KineMedic was optimized for operations in which it is mounted to the floor or the operating table (see Sect. 6.1), and resulted in an asymmetric joint range for joints 4 and 6. It was analyzed whether it would be advantageous to connect the instrument to the robot rotated 180° around q_6 (as has been done for all configurations in Table 6.4 denoted with a *) to reverse the joint limits of joint 6. However, the slight advantages in the quality measures do not justify the increased design complexity.

- Increasing the number of DoF (i.e., the operation parameters) does not significantly raise the considered quality criteria. However, the design alternatives 5* and 6* with a reduced number of DoF yield significantly worse results. Therefore, a design with 5 DoF as initially intended is approved.

- Compared to the original design (see Fig. 6.7a), it is suggested to alter the segment lengths of the robot bases as shown in Fig. 6.9, Fig. 6.10, and Table 6.5, according to the design alternative 9. First details about the design are given in [30].

Among a variety of design alternatives for an actuated, ceiling mounted carrier for up to three KineMedic robots, an actuated carrier with 5 DoF is suggested, providing a good compromise as regards sufficient dexterity and slim design. Stress is laid on the generation of a variety of design solutions, enabling the designer to choose the alternative that best fits to those criteria not definable a priori, such as the required installation space. Furthermore, the compliance with meaningful compulsory requirements (see Table 6.1) can be guaranteed. Note that the KineMedic robot can be attached not only to the ceiling, but also to the OR table, or it can be positioned on the floor (see Fig. 6.11).

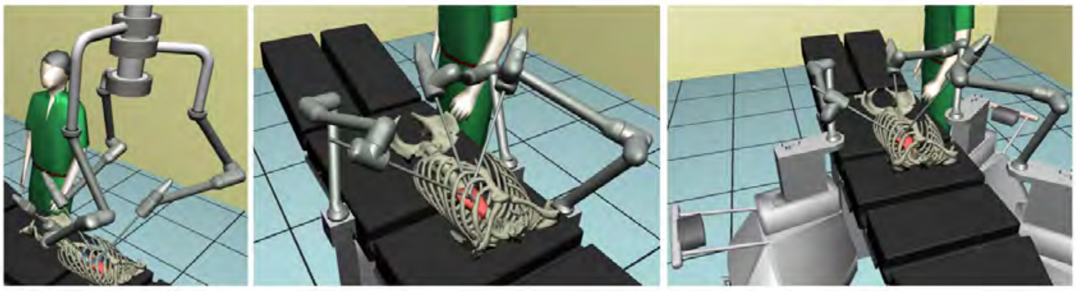


Figure 6.11.: The DLR medical robot KineMedic can be mounted to the ceiling (left), to the table (middle), or positioned on the floor (right).

6.2. Use Case (b): Preoperative Planning for Robotically Assisted Minimally Invasive Interventions

A successful robotically assisted minimally invasive intervention necessitates preoperative planning which is done by the surgeon to prepare the intervention and to decide on the best access to the surgical site. In the context of robotically assisted minimally invasive surgery, this requires to optimally place ports and robots so that important conditions and requirements like sufficient dexterity, optimal view and a collision free workspace are satisfied. The following describes the preoperative planning procedure for the DLR minimally invasive robotic surgery system which utilizes an optimization procedure. The topic is introduced in the next section. Section 6.2.2 describes the manual planning necessary to configure the optimization algorithm, while details about the planning algorithm itself are given in Sect. 6.2.3. Section 6.2.4 shows the implemented validation procedure. Registration and positioning in the OR are addressed in Sect. 6.2.5 and 6.2.6, respectively. As an alternative, a completely intraoperative planning procedure is discussed in Sect. 6.2.7, and Sect. 6.2.8 summarizes the results.

6.2.1. Introduction

A scheme of the DLR MIRS system (see [79]) is shown in Fig. 6.12. It consists of three light-weight, kinematically redundant, impedance controlled surgical robots (two of them equipped with sensorized and actuated instruments, the third one holding a laparoscope) and a master console. The master consists of an autostereoscopic video screen and two kinesthetic feedback devices.

Different approaches are possible to determine a feasible arrangement of the robotically assisted OR: Based on the surgeon's experience, it might be sufficient to use atlases or simple rules of thumb. A more sophisticated approach that involves e.g. optimization algorithms as presented here might however be necessary for more complex or completely new interventions. Another distinction can be made regarding the point of time in the planning procedure:

Preoperative planning is usually based on high resolution imaging such as CT or magnetic resonance tomography (MRT). Planning is not time-critical since it can be done offline. However, it is not yet possible to take into account the real situation in the OR (e.g. organ shifts due to insufflation), and registration is necessary to transfer the results of the planning into the OR.

Intraoperative planning on the other hand takes into account the actual situation in the OR. Furthermore, planning is done in the coordinate system of the OR, therefore registration is not necessary. However, imaging modalities accessible in the OR usually offer lower quality (e.g. CT scans are often not available intraoperatively), and planning algorithms have to be fast since time in the OR is very valuable.

Features of the proposed planning strategy are going to be compared with the preoperative planning environment by the Chir team of INRIA [2] which belongs to the most

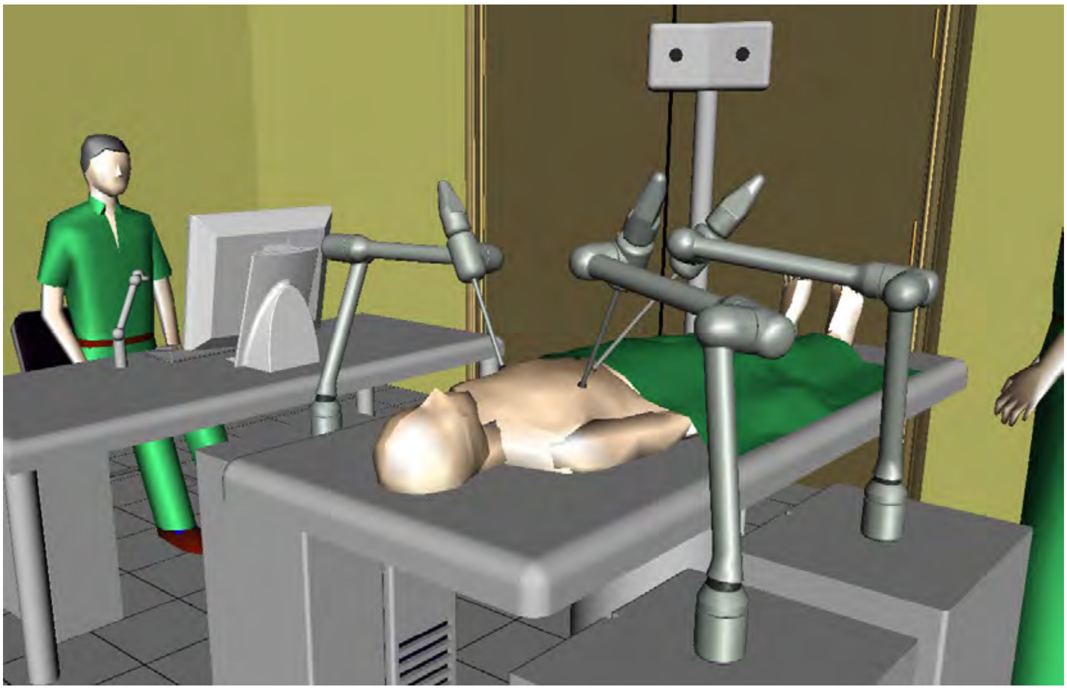


Figure 6.12.: The DLR MIRS system in development.

powerful medical preoperative planning tools developed so far. Other approaches can be found in [12, 14, 61, 95, 100].

The planning environment of the Chir team is highly tailored to the daVinci^{TM} robotic system by Intuitive Surgical[5] which consists of up to four robotic arms mounted on the same base as shown in Fig. 6.13. Each arm has a remote center which has to be positioned at the entry point (port) into the human body. Because all daVinci arms are mounted at a common base, the registration and port positioning can be conveniently done using the robotic arms as position sensor. In case of the DLR setup this is not possible and an external navigation system must be used while registration is done using the handheld 3D-Modeller (developed by DLR [99]) by matching the preoperative with the scanned data (see [48]). An advantage of using the handheld 3D-Modeller is that the application of fiducial markers[6] can be avoided. Furthermore, the handheld 3D-Modeller relies on an infrared tracking system which can also be used to position ports and robots. Alternatively, the 3D-Modeller could be fixed to the TCP of a KineMedic robot, and the registration could then be done automatically.

Another difference between the Chir and the DLR approach is that in case of Chir

[5] See http://www.intuitivesurgical.com/. Accessed on June 14th, 2007.

[6] A fiducial marker is an object used in the field of view of an imaging system which then appears in the image produced. By computation of the transformation rule between the coordinates of the fiducial marker in the preoperative imaging and the intraoperative measurement, registration can be achieved.

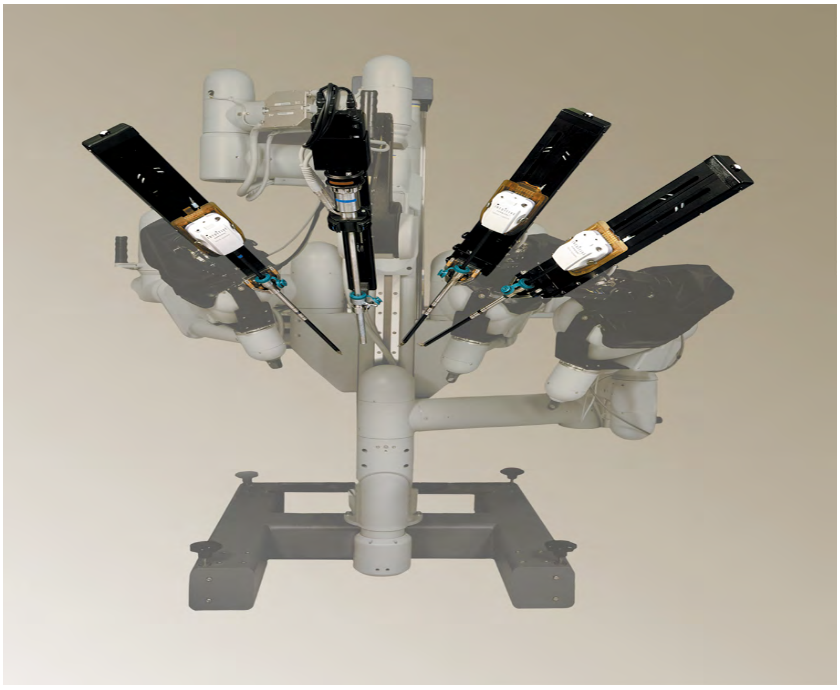
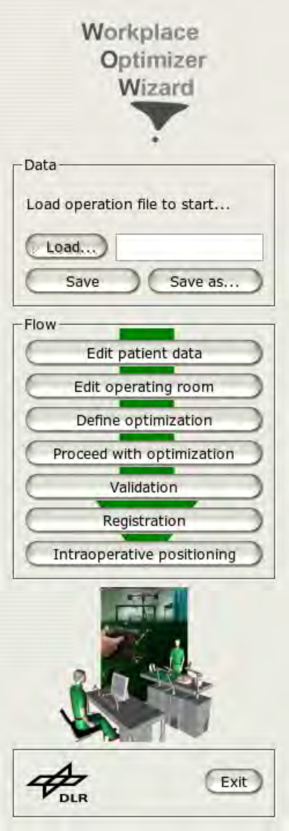


Figure 6.13.: The patient-side cart of the daVinci$^{\mathrm{TM}}$ robotic system by Intuitive Surgical according to [41].

the calculation of the optimal port and robot positions is done subsequently neglecting the dependence of the manipulator dexterity on the robot kinematics. Because this dependence seems to be in most cases rather weak the two-step approach delivers good results as reported in [2]. In contrast to this, the DLR planning tool utilizes an integrated approach where the problem of optimal port and robot placement is solved in one step exploiting more general measures of manipulability and accuracy [54].

The planning procedure implemented in this thesis is shown in Fig. 6.14: First patient specific tomographic data are recorded and segmented in order to obtain 3D models of the anatomical structures of interest (step *Edit patient data* in Fig. 6.14). The surgeon then defines the coarse OR setup, i.e. prepositions robots and other devices according to his experience (step *Edit operating room* in Fig. 6.14). After this the surgeon plans the intervention on the basis of the virtual patient specific model. The parts of the organs which must be reachable by the instruments and visible by the laparoscope are determined (step *Define optimization* in Fig. 6.14). This is followed by the determination of optimal positions for ports and robots which is conducted in the background automatically by the optimization algorithm (step *Proceed with optimization* in Fig. 6.14). The results of the optimization algorithm are displayed to the surgeon and validated by him. The intervention is simulated in a virtual environment (step *Validation* in Fig. 6.14), and the surgeon decides whether the proposed port and robot positions are feasible. In case any problems are encountered, the planning is possibly modified and the optimization is redone. Then, during the preparation of the actual intervention, the preoperatively determined data must be transferred into the OR. The first step in doing this is to register the virtual preoperative environment to the real anatomy of the patient (step *Registration* in Fig. 6.14). Eventually, the ports and robots must be positioned according to the preoperative plan (step *Intraoperative Positioning* in Fig. 6.14).



Load operation file to start...

Figure 6.14.: Steps of a robotically assisted intervention with preoperative planning as implemented in the DLR operation planner.

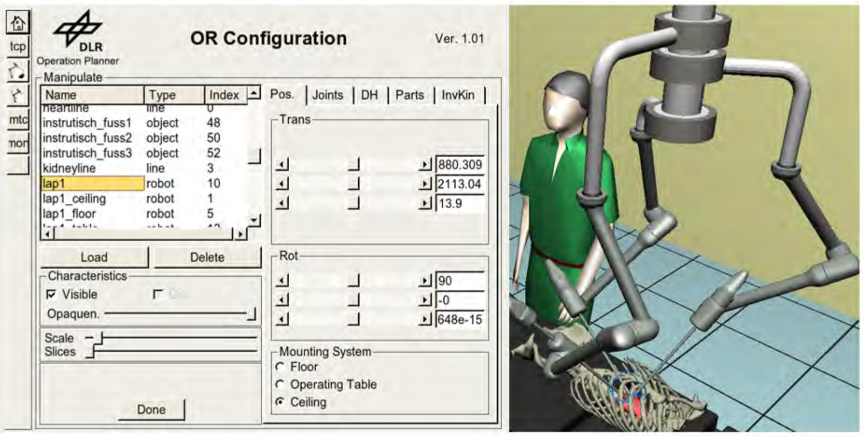
The next sections describe the used methods:

- The planning of the intervention by the surgeon (Sect. 6.2.2),
- the optimization procedure (Sect. 6.2.3),
- the process of validation and simulation (Sect 6.2.4),
- the registration procedure (Sect. 6.2.5), and
- the transfer of the planning results into the OR (Sect. 6.2.6).

An alternative procedure exclusively based on intraoperative data is described in Sect. 6.2.7 and Sect. 6.2.8 summarizes the results.

6.2.2. Intervention Planning

After segmenting the tomographic, patient specific data[7], the surgeon loads and adjusts a template OR setup according to the intervention to be performed (see Fig. 6.15). The template OR setup (coming e.g. from previous interventions) models the real OR and comprises e.g. the used robotic system, the OR table and various additional devices as needed in the real intervention.



Name	Type	Index
heartline	line	0
instrutisch_fuss1	object	48
instrutisch_fuss2	object	50
instrutisch_fuss3	object	52
kidneyline	line	3
lap1	robot	10
lap1_ceiling	robot	1
lap1_floor	robot	5

☑ Visible

Trans: 880.309, 2113.04, 13.9

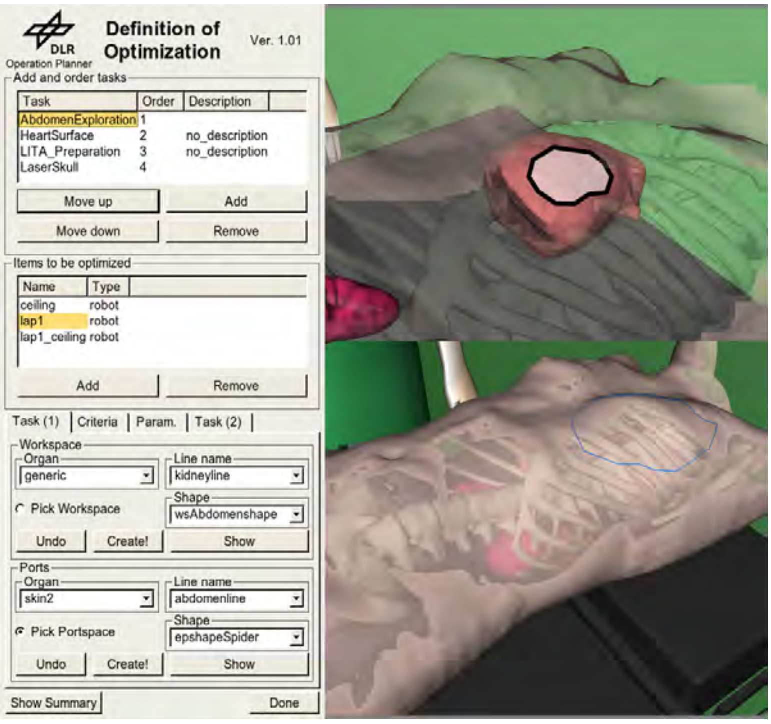
Rot: 90, -0, 648e-15

Mounting System
○ Floor
○ Operating Table
● Ceiling

Figure 6.15.: Planning of the intervention: Adaptation of the OR template.

Based on the 3D model of the patient the surgeon then determines the area of interest on the resp. organ as well as potential port locations on the skin of the patient. This is done by simply drawing on the organs as depicted in Fig. 6.16. An important issue to be taken into consideration is the intraoperative shift of the organs due to e.g. insufflation. To the author's knowledge, there are no algorithms available to predict the intraoperative position of the organs preoperatively with sufficient accuracy. However, the shift can be roughly estimated: According to [93], displacements are in the range of up to one centimeter for locations on the stabilized heart. To take into account this shift, the workspace which has to be accessible to the robots is therefore enlarged by 1 cm in all directions as compared to the workspace specified by the surgeon. The workspace is generated from the surgeon's specifications as follows: First the borderline of the area of interest is enlarged by 1 cm with respect to the reverse direction to the center of curvature. The workspace is then defined by translating the enlarged area of interest $h+1$ cm in direction of the averaged surface normal (with h the height of the workspace as given by the surgeon) and 1 cm in the opposite direction as depicted in Fig. 6.17(a).

[7] Image segmentation denotes the procedure of extracting features (in this context surface data of the patient organs) from images using e.g. image contrast information. Various software tools are available for (semi-)automatic segmentation of tomographic data, see e.g. the software Amira (http://www.amiravis.com, accessed on April 18th, 2007).



DLR Operation Planner — Definition of Optimization — Ver. 1.01

Add and order tasks

Task	Order	Description
AbdomenExploration	1	
HeartSurface	2	no_description
LITA_Preparation	3	no_description
LaserSkull	4	

Items to be optimized

Name	Type
ceiling	robot
lap1	robot
lap1_ceiling	robot

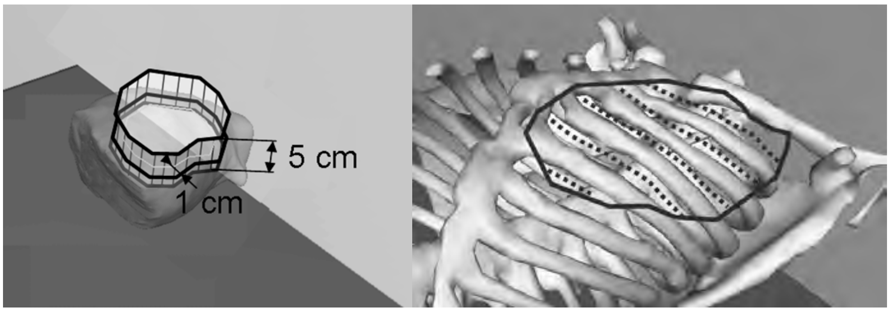
Task (1) | Criteria | Param. | Task (2)

Workspace: Organ: generic; Line name: kidneyline; Shape: wsAbdomenshape; ○ Pick Workspace

Ports: Organ: skin2; Line name: abdomenline; Shape: epshapeSpider; ● Pick Portspace

Figure 6.16.: Planning of the intervention: Determination of the area of interest on the organ to be operated (upper part, right) and of potential port locations (lower part, right).

From the area accessible for port locations, possible ones are calculated using the 3D model of the rib cage where the intercostal space is determined (Fig. 6.17(b)).



(a) (b)

Figure 6.17.: Workspace (a) and possible locations for the entry points in the intercostal space (b).

As the next step, the optimization procedure is described that leads to an optimal OR setup.

6.2.3. Determination of an Optimal OR Setup

In order to achieve an optimal OR setup, the positions of ports and robot bases must be determined so that important conditions and requirements like sufficient dexterity, optimal view and a collision free workspace are satisfied. This is assured using the optimization procedure described in Sect. 4.2. With respect to the DLR MIRS system, the bases of three robots have to be positioned as well as their resp. ports. Furthermore, variable instruments lengths may be chosen and eventually the position of the patient and that of other OR equipment can be taken into account. Reasonable optimization criteria such as avoidance of collisions and distance from joint limits are given in [2] for determination of an optimal OR setup. The approach presented here additionally formulates constraints on the dexterity of the robots itself. Furthermore, it is guaranteed that the robots provide sufficient manipulability and accuracy throughout the workspace (see Sect. 4.3). The optimization parameters as well as their upper and lower limits can be conveniently adjusted by the GUI of the Workplace Optimizer Wizard (see App. C). Likewise, optimization criteria and constraints can be chosen and weighted.

The discretized workspace must be completely accessible by any robot holding an instrument. To obtain a robust setup, this must also be true if small variations are applied to e.g. the port locations. By this way, unavoidable errors in registration can be compensated. As mentioned above, all optimal parameters are determined at the same time. Since the resulting search space is of high dimension (up to 17 optimal parameters are sought simultaneously), and the optimization problem itself is discontinuous, Genetic Algorithms are suitable for optimization (see Sect. 4.2 and [47] for details on the optimization algorithm). A subsequent gradient-based method speeds up the process. The optimization is particularly fast (less than 5 minutes on a standard Intel Core$^{\mathrm{TM}}$ 2 Duo Processor 1.86 GHz) if an adequate template can be provided. In this case, the optimization algorithm is initialized with parameter values close to the sought minimum and therefore achieves good solutions within a few iterations.

As a result of the optimization process, different feasible parameter sets are obtained, related to local minima of the optimization problem. In the next step, the surgeon validates the parameter set of the best solutions and chooses a set according to his experience.

6.2.4. Validation and Simulation

The computed optimal OR setup is presented to the surgeon in a virtual environment as depicted in Fig. 6.18, and it can be verified whether the whole workspace is accessible by moving the robots interactively or by automatically positioning the robots with respect to the discretized positions inside the workspace. The found optimal OR setup can also be compared to alternative setups referring to local optima. Furthermore, the view through the laparoscope can be simulated.

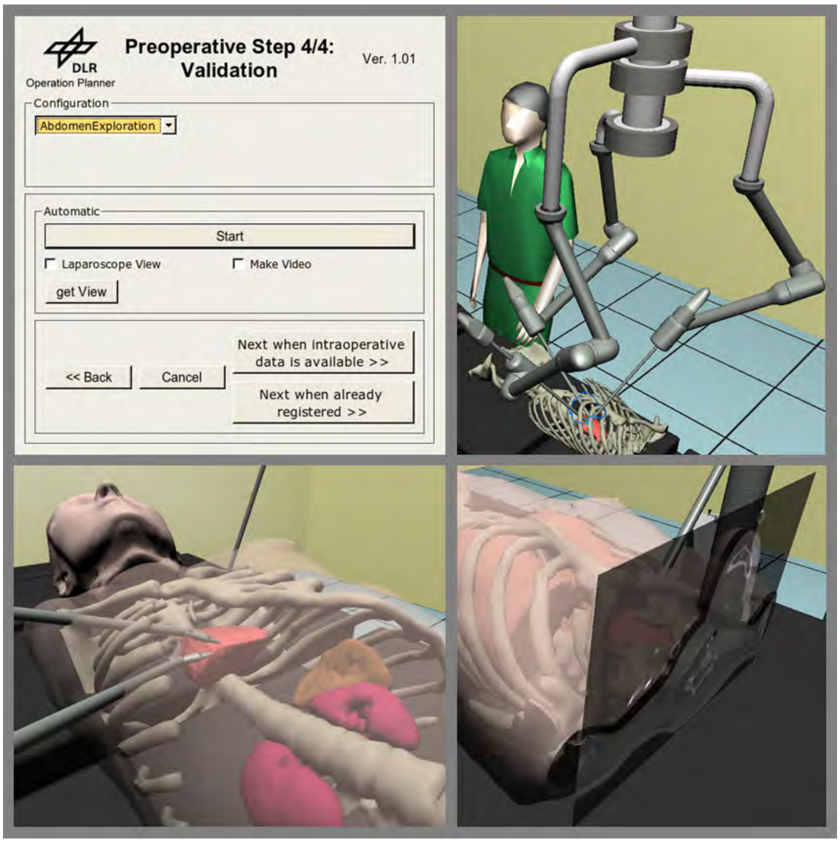


Figure 6.18.: Validation of the optimized OR setup.

6.2.5. Registration

Patient registration is necessary whenever planning data (e.g. port entry positions, or biopsy needle trajectories based on preoperative data) have to be transferred into the OR. In numerous medical interventions a successful registration is crucial for the quality of the medical procedure, including radio surgery and navigated surgery [37].

In the scope of this thesis, a markerless and contact-free registration procedure is developed, using the DLR 3D-Modeller as shown in Fig. 6.19. The 3D-Modeller [99] is a hand-guided device that allows for acquiring the patient surface intraoperatively. It provides three different range sensors: a stereo camera sensor (SCS), a laser-range scanner (LRS), and a light-stripe profiler (LSP). A medical application as presented in [57] was considered during the development period in order to make the 3DMo suitable for usage in the OR.

For registration, the patient surface (see Fig. 6.20 for an experimental scan of a plastic model of the thorax) is scanned, and the scan is then registered with the preoperative data using an Iterative Closest Point (ICP) algorithm [8], a frequently used algorithm for local registration of rigid structures. Naturally, soft tissue displacements of the thorax

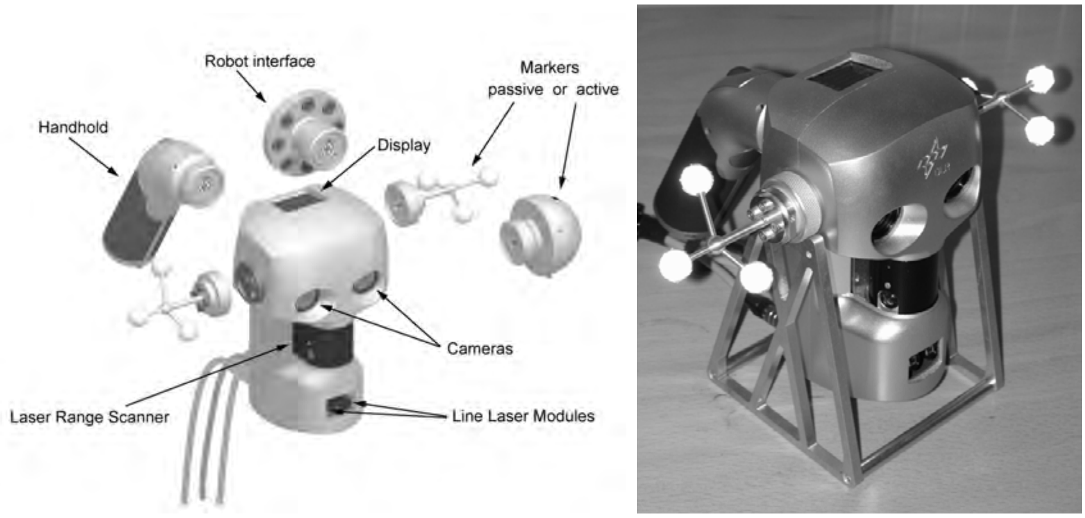


Figure 6.19.: The DLR 3D-Modeller.

are not considered in this method, and a more complex, non-rigid approach might be necessary which is work in progress.

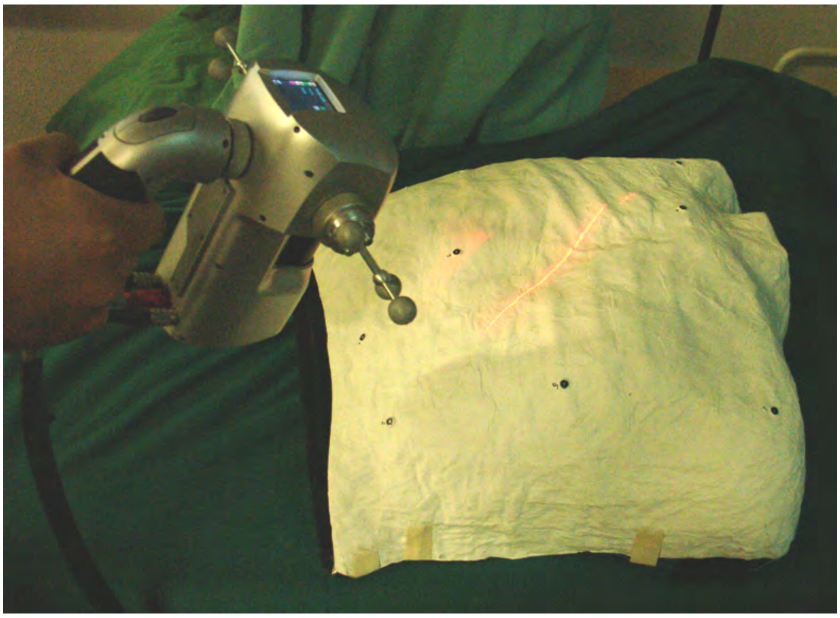


Figure 6.20.: Intraoperative scan of the patient for markerless registration with the handheld 3D-Modeller.

Each scan of the plastic model takes only a few seconds (ranging from 10 s – 20 s for LSP and 15 s – 30 s for LRS). The LRS method takes more time since the 3DMo has to be positioned closer to the surface, and the range of recorded data is thus smaller than in case of the LSP method. However, the LRS method is more robust against illumination changes and lighting conditions as compared to the LSP method.

Using the software Visu3D [9], fast surface reconstruction is possible and allows for very good online visualization during the scanning procedure. An ICP algorithm is integrated into the planning software, and registration takes between 30 s – 90 s, depending whether manual adjustment of the initial pose is necessary or not. Figure 6.21 shows a sample dataset before (left) and after registration (right). According to [48], the worst case deviation between the gold standard method based on fiducial markers and the used surface-based methods is lower than 3 mm and thus sufficient for planning of MIRS operations.

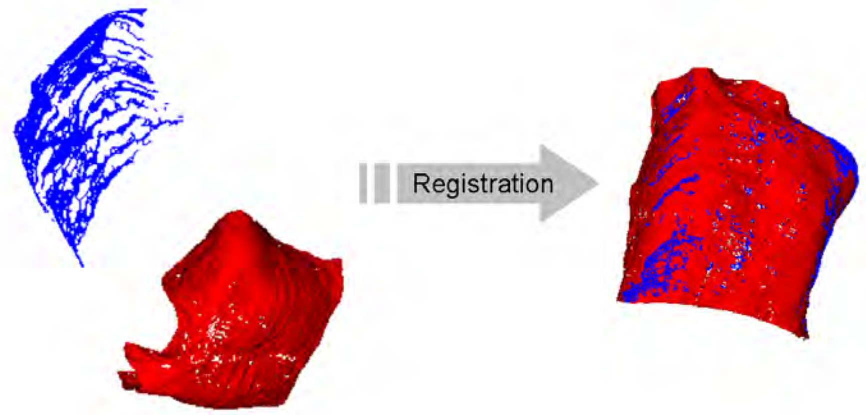


Figure 6.21.: Visualization of the registration procedure.

6.2.6. Positioning

Once registration is accomplished, the positioning of the ports and the robot bases can be carried out using pointing targets localized by the tracking system as shown in Fig. 6.22 (left) and markers attached to the robots. As alternative, a very convenient and automatic method is provided by the *AutoPointer* presented in Sect. 4.5. As shown in Fig. 6.22 (right), the *AutoPointer* can be used to automatically mark the port locations directly onto the patient using laser light. Additionally, if a motorized mounting device is used, the desired base positions of the robots can be obtained automatically as well.

6.2.7. Intraoperative Planning

Intraoperative planning is a promising alternative to preoperative planning, basically because it does not need registration and it takes into account the real situation in the

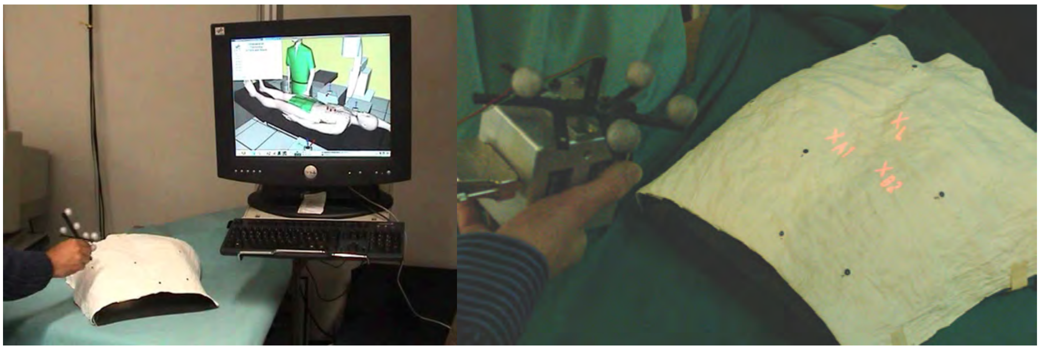
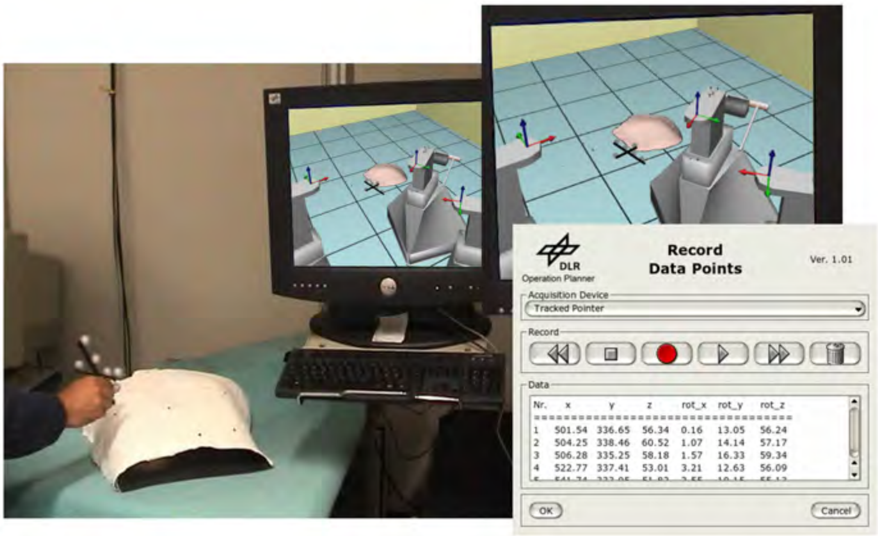


Figure 6.22.: Positioning of the ports on the patient using (left) an optically tracked pointer with feedback from the VR and (right) the *AutoPointer* to automatically mark the entry point positions onto the patient.



Record Data Points

Ver. 1.01

Acquisition Device: Tracked Pointer

Data

```
Nr.   x        y        z       rot_x   rot_y   rot_z
=================================================
1     501.54   336.65   56.34   0.16    13.05   56.24
2     504.25   338.46   60.52   1.07    14.14   57.17
3     506.28   335.25   58.18   1.57    16.33   59.34
4     522.77   337.41   53.01   3.21    12.63   56.09
```

Figure 6.23.: Validation of the optimized OR setup.

OR. The following will analyze the possibility of using the methods presented above in a completely intraoperative procedure.

Basically, the optimization algorithm requires the following mathematic descriptions of the situation in the operating room [55]:

a) The surface that contains possible entry points into the patient,

b) the desired workspace, and

c) the volumes that contain the robot bases.

An approximation of a) can be obtained by use of an optically tracked pointer or the 3DMo to scan the patient surface. To approximate b), the surgeon could first choose an

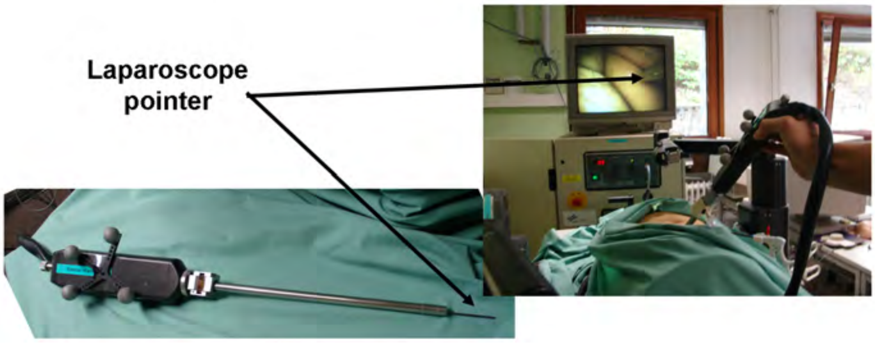


Figure 6.24.: An optically tracked endoscope with a calibrated probe to record sample points inside the patient.

entry point for the endoscope[8] and then explore the desired workspace with an optically tracked endoscope equipped with a calibrated probe to record sample points as shown in Fig. 6.24. Also the use of ultrasound or tracked mobile X-ray fluoroscopic equipment could be feasible. Upon completion, a volume is calculated that contains all the sample points. The volumes in c) are determined by the surgeon's choice of the mounting option. The optimization algorithm then defines good robot base positions and entry points. Since this optimization is performed intraoperatively, it is important to obtain the results within a few minutes. Fast algorithms e.g. for computation of inverse kinematics as developed in Sect. 3.2.1 are therefore very important.

Preoperative planning definitely is crucial for a successful intervention. However, for the placement of the trocars and the robots itself, intraoperatively easily obtainable data could be sufficient and probably even more appropriate for intervention planning as compared to state of the art preoperative planning. This is mainly due to differences between the preoperative data and the situation in the OR. Since the intraoperative procedure proceeds without registration, it furthermore promises to be faster than conventional methods, both pre- and intraoperatively. It might therefore gain more attention in future research.

6.2.8. Results

Preoperative planning of robotically assisted minimally invasive interventions including optimization algorithms is necessary for high quality interventions. The planning process has to be transparent to the surgeon, and the optimization criteria must be significant for and adaptable to the resp. intervention. Especially the intraoperative steps of the planning procedure are highly time-critical.

Starting from segmented patient data, the following time requirements for the planning

[8] It is supposed that the choice of the port for the endoscope is less critical than the choice of the other ports for the instruments.

procedure are estimated[9]: The interactive definition of the optimization takes between 10 – 15 min. Automatic optimization is performed within 2 – 20 min. Time requirements of the latter step strongly depend on the size of the search space (the search space is defined by the number of parameters and their limits) and the initial solution. The subsequent validation takes about 5 min. Registration and positioning require about 5 – 10 min each.

The proposed planning procedure utilizes an optimization approach. The optimization procedure includes criteria to guarantee the dexterity of the robotic system and provides robustness with respect to inaccuracies in transferring the results of the planning into the OR. The validation of the optimized OR setting by the surgeon is done through simulation in a virtual environment. The results are transferred into the OR by markerless registration with the DLR 3D-Modeller and by positioning the robots and ports using either an optical tracking device, or the *AutoPointer* presented in Sect. 4.5, or an actuated carrier for automatically positioning the robot base positions.

Alternatively, the whole planning procedure could take place intraoperatively. This way, planning is based on the real situation in the OR and registration is not necessary. However, preoperative imaging devices usually provide better quality than imaging devices available during the intervention, and the optimization has to be very fast not to delay the operation.

6.3. Summary and Conclusion

This chapter shows the implementation of planning and optimization methods for robotically assisted workplaces.

Section 6.1 describes the optimization procedure to assist during the design of the medical robot KineMedic. The following studies are presented:

- Optimization of the link lengths of the KineMedic robot
- Determination of the structure and size of the actuated instrument for the KineMedic robot
- Optimization of the design for an actuated carrier to position up to three KineMedic robots

In this context, the robustness of the found robot setup with respect to registration and positioning inaccuracies is an important feature. While for the design optimization of the KineMedic robot the overall size of the robot was the only optimization criterion, for the design optimization of the actuated carrier a multi criteria optimization using a

[9] Up to now, no clinical tests were carried out. Therefore, the mentioned time requirements have to be further verified.

weighting factor has been chosen. Both approaches lead to an optimized design and observe the specified constraints. Work remains to be done to get more precise information concerning required velocities and rotational accuracies of surgical instruments during MIRS interventions. Publications in this context [13, 40, 86, 97] only cover analyses of translational accuracy. The presented methods can easily be adapted to solve further robot design optimization problems.

In Sect. 6.2, the planning procedure for the DLR MIRS setup is described. After a manual planning by the surgeon in order to configure the optimization algorithm, determination of good setups with respect to the chosen criteria is done automatically. In the subsequent validation procedure, the surgeon verifies the computed setup. Registration is performed using a markerless and contact-free procedure. A key advantage of this method as opposed to artificial marker-based registration is the non-invasiveness of the procedure. However, the achieved accuracy is often reported to be lower, and known systems often lack dexterity. The method presented in this thesis shows a worst-case accuracy of better than 3 mm (according to [48]). To further increase accuracy and reduce error-proneness, the multi sensory concept of the 3DMo allows for fusion of additional pose estimation algorithms such as e.g. visual SLAM [10], subject to current work. Positioning can be carried out advantageously using the *AutoPointer*. The first mock-up of the device shows an accuracy of better than 1 mm at a distance of 0.5 m.

The whole OR planning procedure is very fast, taking only $20-35$ min preoperatively and $10-20$ min intraoperatively. Preoperative time requirements will be further reduced as soon as various templates are available which then only have to be slightly modified for the planned operation. This way, both the manual planning by the surgeon and the automatic optimization algorithm supposably will take less time. The templates are acquired by storing previous operations in e.g. a database. As an alternative to preoperative planning, the feasibility of a completely intraoperative planning procedure is analyzed. Using e.g. an optically tracked laparoscope with a calibrated probe, it might be possible to perform the planning of optimal robot positions and entry points into the patient completely intraoperatively, and thus take into account the real situation in the OR.

Part IV.

Conclusion and Outlook

Conclusion

In the last 50 years, robotics has unceasingly attracted interest: Robotic systems nowadays assist the human in various tasks coming from industrial, security, service, rehabilitation, and surgical applications. In these tasks, the robots work either autonomously, teleoperated, or in close cooperation with human or other robotic systems. While robotic workplaces were rather static and straightforward in the beginning, current workplaces are very complex, and in many cases the robotic systems must provide techniques to adapt to a cluttered and constantly changing environment. This is especially true for the field of robotic surgery, since motions of the patient and the organs are highly unpredictable up to now. In applications that exhibit close interaction with humans, safe operation is the most crucial issue. Thorough planning and optimization of both the robot itself and the setup of the robotic workplace are important instruments to cope with the increased complexity of today's robotic applications. It can lead to safer operation and may avoid various sources of malfunction of the robotic system.

This thesis provides methods and tools for the optimization of robot design (robot synthesis) and of robotic workplaces based on lifelike simulation and optimization algorithms. Special attention is paid to closed form solutions for inverse kinematics and singularity computation for robots with kinematic redundancy. Due to closed form solutions it is possible to choose the operating field of the robot in a way that no singular configurations or robot joint limits occur during task execution. This is important to guarantee safe operation.

The task-oriented optimization procedure developed in this thesis relies on both gradient-based and evolutionary optimization algorithms, aiming at combining the advantages of both algorithms: The evolutionary algorithm provides a variety of near-optimal solutions, whereas the gradient-based algorithm allows for fast determination of the local minima. Various criteria can be weighted and included in the optimization problem, and new criteria for minimally invasive surgery are developed, such as manipulability and accuracy measures, as well as collision avoidance.

Results of the analysis of robot kinematics lead to the implementation of a library for inverse kinematics, offering a variety of algorithms for different application areas, users, and robot kinematics. The presented methods for robot synthesis are used to design the newly developed medical robot KineMedic at DLR and yield optimal segment lengths for the robot itself as well as for an articulated surgical instrument. Furthermore, the design of a ceiling mounted automatic positioning device for up to three KineMedic robots is optimized in terms of necessary actuators and segment lengths. Eventually, planning and optimization of a robotic workplace is exemplified by the preoperative planning of a robotically assisted minimally invasive surgery. Additionally, methods to transfer the results of the planning (based on the virtual environment) into the real world are developed: For registration, a markerless and contact-free procedure using the DLR 3D-Modeller is presented. The positioning of all devices is assisted by the *AutoPointer*, a device that automatically marks positions in space.

Outlook

This thesis provides the basic steps towards a universal library for inverse kinematics computation. The algorithms are currently used in a variety of applications at the Institute of Robotics and Mechatronics such as task planning, path planning, service robotics, and teleoperation. Due to the user feedback, the library will constantly advance. Important next developments comprise the inclusion of inverse velocity kinematics, and both forward and inverse position kinematics algorithms for parallel structures. In addition, a new algorithm for branched kinematic structures is envisioned, based on the SQP algorithm and including all available closed form solutions. Moreover, efficient algorithms for collision avoidance should be included in the library.

The methods and applications presented in the scope of workplace optimization and robot synthesis need to be further evaluated in new application scenarios. In particular, work remains to be done to promote intraoperative planning. The advantages of intraoperative planning are obvious since preoperative prediction of the real environment with high accuracy is often difficult. Especially in surgical robotics, the simulation of soft tissue behavior is in most cases still not sufficiently realistic. This is mainly due to (1) the various interaction effects between the organs, (2) the complex physical models of the organs, and (3) the fact that patient specific material parameters of the organs are difficult to obtain. In case of human robot cooperation, the motions of the human operator are only very roughly predictable and add further uncertainty. Preoperative planning therefore has to be supported by intraoperative methods to adjust the plan to the real situation and to perform replanning whenever necessary.

During task execution, sensor data can be used to update e.g. the model of the environment. Large differences between model and reality thereby either originate from sensor errors or from large soft tissue displacements and need further verification by additional sensor data. Strategies for an efficient update of the scene based on entropy considerations are currently explored [98]. Especially for image processing, reliable algorithms to process the sensor data are barely available up to now. Methods to fuse different sensor data, using e.g. a Kalman filter in order to obtain data with increased reliability and accuracy are therefore a vital requirement. On the other hand, reliable sensor data acquisition is only possible if the sensors are placed so that the environment to be gaged is in the sensor range throughout the task. It would be therefore advantageous to include an optimization of the sensor placement into the planning procedure.

Preoperative planning can be used as basis for the intraoperative task execution, while the actual plan should be constantly updated according to the sensor data. The algorithms for replanning have to be particularly fast to cope with near-real-time requirements, which abets to the use of fast closed form algorithms as presented in this thesis.

Eventually, sensor-based control on the one side and planning algorithms on the other hand should seamlessly interact with each other, thus closing the gap between real-time control and preoperative planning.

Part V.

Appendices

A. Nomenclature

A frame F is denoted as $\{F\}$. Two frames $\{j\},\{k\}$ are related to each other by the following homogeneous transformation:

$$
{}^{i}_{k}T = \begin{pmatrix} {}^{i}_{k}R & {}^{i}_{k}p \\ 0 & 1 \end{pmatrix},
$$

with ${}^{i}_{k}R \in R^{3\times 3}$ a rotation matrix and ${}^{j}_{k}\mathbf{p} = \begin{pmatrix} {}^{j}_{k}p_x & {}^{j}_{k}p_y & {}^{j}_{k}p_z \end{pmatrix}^T$ the origin of the frame $\{k\}$ written in frame $\{j\}$. The inverse transformation matrix will be written as $\left({}^{j}_{k}\mathbf{T}\right)^{-1} = {}^{k}_{j}\mathbf{T}$.

To describe the robot kinematics, the notation of DH parameters is used in the variant of Craig [19], and the transformation matrix between two subsequent link frames $\mathbf{T}_{i-1}$ and $\mathbf{T}_i$ is obtained through the joint angle q_i and the DH parameters a_{i-1}, α_{i-1}, d_i, and θ_i:

$$
{}^{i-1}_{i}\mathbf{T} = \begin{pmatrix} c_{\theta_i+q_i} & -s_{\theta_i+q_i} & 0 & a_{i-1} \\ c_{\alpha_{i-1}} s_{\theta_i+q_i} & c_{\alpha_{i-1}} c_{\theta_i+q_i} & -s_{\alpha_{i-1}} & -d_i\, s_{\alpha_{i-1}} \\ s_{\alpha_{i-1}} s_{\theta_i+q_i} & s_{\alpha_{i-1}} c_{:\theta_i+q_i} & c_{\alpha_{i-1}} & d_i\, c_{\alpha_{i-1}} \\ 0 & 0 & 0 & 1 \end{pmatrix},
$$

with $s_\bullet = sin(\bullet)$ and $c_\bullet = cos(\bullet)$. The transformation matrix between the inertial frame $\{I\}$ and an arbitrary link frame $\{i\}$ of a serial kinematic chain with n joints can be obtained as follows:

$$
{}^{I}_{i}\mathbf{T} = {}^{I}_{B}\mathbf{T} \prod_{k=1}^{i} {}^{k-1}_{k}\mathbf{T} = {}^{I}_{n}\mathbf{T} \left(\prod_{k=i+1}^{n} {}^{k-1}_{k}\mathbf{T} \right)^{-1}, \tag{A.1}
$$

with $\{B\}$ the robot base frame. Vectors and matrices are written in bold (as opposed to scalars). Though especially the use of quaternions [75] has proven to be beneficial in many cases, it is not required in the scope of this thesis. Further symbols are denoted in Table A.1. Abbreviations are given in Table A.2.

Table A.1.: Symbols

Symbol	Definition
$\dot{f}$, $\ddot{f}$	First and second derivative $\frac{df}{dt}$, $\frac{d^2f}{dt^2}$ w.r.t. time
ω	Angular velocity vector
$\mathbf{q}$	Joint angle vector
$\mathbf{x}$	Position and orientation of e.g. the tool tip frame (TCP), written as a 6-DoF vector
$\mathbf{J}$	Jacobian matrix, with $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$
$\mathbf{J}^{+}$	Pseudo-inverse of the Jacobian, with $\mathbf{J}^{+} = \mathbf{J}^{\mathbf{T}}\left(\mathbf{J}\mathbf{J}^{\mathbf{T}}\right)^{-1}$
$\mathbf{J}^{i}$	The i-th submatrix (minor) obtained by suppressing column i of the matrix $\mathbf{J}^{i} \in \mathbb{R}^{m\times n}$, $\;n = m+1$
a, α, d, and θ	DH parameters
$h(\ldots)$	Forward kinematics function
$f(\ldots)$	Optimization function
$g(\ldots)$	Constraints function
$\mathbf{p}$	Optimization parameter vector
σ	Singular value
λ	Eigenvalue

Table A.2.: Abbreviations

3DMo	3D-Modeller
API	Application Programming Interface
CCD	Charge-Coupled Device
COS	Coordinate System
CT	Computer Tomography
DH	Denavit-Hartenberg
DICOM	Digital Imaging and Communications in Medicine
DLR	German Aerospace Center
DoF	Degree(s) of Freedom
DSP	Digital Signal Processing
EMC	Electromagnetic Compatibility
GA	Genetic Algorithm
gcc	GNU Compiler Collection
GNU	GNU's Not UNIX
ICP	Iterative Closest Point
LITA	Left Internal Thoracic Arteria
LM	Levenberg-Marquardt
LRS	Laser-Range Scanner
LSP	Light-Stripe Profiler
LWR	Light-Weight Robot
MIS	Minimally Invasive Surgery
MIRS	Minimally Invasive Robotic Surgery
MRT	Magnetic Resonance Tomography
OR	Operating Room
PSD	Position Sensitive Device
SCS	Stereo Camera Sensor
SQP	Sequential Quadratic Programming
TCP	Tool Center Point
VR	Virtual Reality
VTK	Visualization ToolKit
WOW	Workplace Optimizer Wizard

B. Minors for Calculation of Singularities of the DLR Medical Arm

The relevant determinants from Sect. 3.3.1.1 yield:

$$|\mathbf{J}_{22}| = -s_6, \quad \left|\mathbf{J}_{11}^1\right| = -a_3 d_5 c_3 s_4 \left(d_5 c_4 + a_3\right),$$
$$\left|\mathbf{J}_{11}^2\right| = a_3 d_5 c_2 s_3 s_4 \left(d_5 c_4 + a_3\right), \quad \left|\mathbf{J}_{11}^3\right| = a_3 d_5 s_4 \left(s_2 c_3 \left(a_3 + d_5 c_4\right) + d_5 c_2 s_4\right),$$
$$\left|\mathbf{J}_{11}^4\right| = 0, \quad \left|\begin{pmatrix} \mathbf{J}_{11} & \mathbf{0} \\ \mathbf{J}_{21} & \mathbf{J}_{22}^1 \end{pmatrix}\right| = -a_3 d_5 c_2 s_4 \left(d_5 s_6 + a_3 s_4 c_5 c_6 + a_3 c_4 s_6\right),$$
$$\left|\begin{pmatrix} \mathbf{J}_{11} & \mathbf{0} \\ \mathbf{J}_{21} & \mathbf{J}_{22}^2 \end{pmatrix}\right| = a_3^2 d_5 c_2 s_4^2 s_5 s_6, \quad \left|\begin{pmatrix} \mathbf{J}_{11} & \mathbf{0} \\ \mathbf{J}_{21} & \mathbf{J}_{22}^3 \end{pmatrix}\right| = a_3^2 d_5 c_2 s_4 c_5;$$

The determinants equal zero for the following joint angles:

$$\left|\mathbf{J}_{11}^1\right| = 0: \quad q_3 = \frac{\pi}{2} + \pi k \quad \vee \quad q_4 = \pi k \quad \vee \quad q_4 = \pm \operatorname{arccos}\left(-\frac{a_3}{d_5}\right) + 2\pi k;$$
$$\left|\mathbf{J}_{11}^2\right| = 0: \quad q_2 = \frac{\pi}{2} + \pi k \vee q_3 = \pi k \vee q_4 = \pi k \vee q_4 = \pm \operatorname{arccos}\left(-\frac{a_3}{d_5}\right) + 2\pi k;$$
$$\left|\mathbf{J}_{11}^3\right| = 0: \quad q_4 = \pi k \quad \vee \quad q_2 = \pm \arctan\left(\frac{s_4 d_5}{c_3 \left(d_5 c_4 + a_3\right)}\right) + \pi k,$$
$$\left|\mathbf{J}_{11}^4\right| = 0 \forall q_i;$$
$$\left.\left|\begin{pmatrix} \mathbf{J}_{11} & \mathbf{0} \\ \mathbf{J}_{21} & \mathbf{J}_{22}^1 \end{pmatrix}\right|\right|_{s_6=0} = 0: q_2 = \frac{\pi}{2} + \pi k \quad \vee \quad q_4 = \pi k;$$
$$\left.\left|\begin{pmatrix} \mathbf{J}_{11} & \mathbf{0} \\ \mathbf{J}_{21} & \mathbf{J}_{22}^2 \end{pmatrix}\right|\right|_{s_6=0} = 0, \forall q_i;$$
$$\left.\left|\begin{pmatrix} \mathbf{J}_{11} & \mathbf{0} \\ \mathbf{J}_{21} & \mathbf{J}_{22}^3 \end{pmatrix}\right|\right|_{s_6=0} = 0: \ q_2 = \frac{\pi}{2} + \pi k \quad \vee \quad q_4 = \pi k \quad \vee \quad \frac{\pi}{2} + \pi k;$$

C. Selected Documentation of the Workplace Optimizer Wizard

This Appendix describes the basic usage of the software *Workplace Optimizer Wizard*, namely the *Workplace Optimizer Wizard GUI*, the *Geoserver Viewer*, and the *Registration Tool.* The software runs on the Linux computers of the DLR Institute of Robotics and Mechatronics.

C.1. The Workplace Optimizer Wizard GUI

The *Workplace Optimizer Wizard GUI* provides the user interface to plan robotically assisted interventions. In addition to the *Workplace Optimizer Core*, it uses the functionality of the *Geoserver Viewer* (Sect. C.2) for visualization, the *Registration Tool* (Sect. C.3) for registering pre- and intraoperative data, and the commercially available software Amira[1] in case CT data has to be segmented.

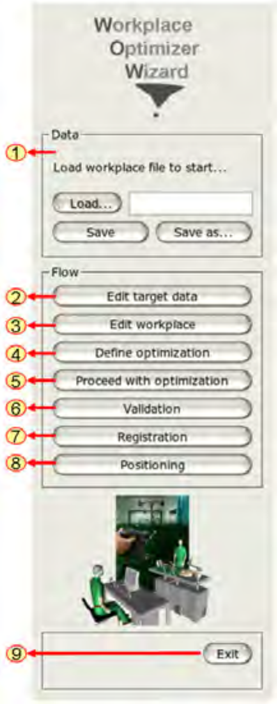
C.1.1. Flow (Fig. C.1)

The main window of the software (Fig. C.1) controls the flow of the optimization procedure. Furthermore, it provides buttons to save and load data as well as to exit the program.

The following functions are available, according to Fig. C.1:

1. Load and save data.
2. Edit target data: In case of surgical interventions, patient data is chosen, segmented and stored in a format suitable for the Workspace Optimizer Wizard. For this step, the commercially available software Amira is used. Segmented surfaces are stored as “Wavefront” files and then loaded into the *Workplace Optimizer Wizard.*
3. Edit Workplace: The available equipment is chosen in this step, along with their coarse position in the OR.

[1] See http://www.amiravis.com. Accessed on April 18th, 2007.



Load workplace file to start...

Figure C.1.: Flow of the Workplace Optimizer Wizard.

4. Define Optimization: Each operation consists of specific tasks that have to be performed in a certain sequence. Bypass operations could e.g. be structured into (1) Preparation of the arteria mammaria interna, (2) Preparation of the heart, and (3) Bypass anastomosis. For each task, candidate entry points, robotic instruments, and workspaces are defined by the user.

5. Proceed with optimization: According to the chosen optimization algorithms (e.g. Genetic Algorithms), a variety of favorable OR setups is determined automatically.

6. Validation: Each proposed OR setup is shown to the user, and each task can be shown in simulation. The user can choose the preferred setup or otherwise go back to redefine the optimization problem.

7. Registration: As soon as intraoperative data of the patient in the OR is available (e.g. obtained through scan with the DLR 3DMo), registration is performed using the software described in Sect. C.3.

8. Positioning: The results of the planning, e.g. optimal trajectories, robot base positions or entry points into the human body are transferred into the OR. Therefore, the results are visualized and a tracked pointer or the *AutoPointer* can be used to localize the positions in the OR.

9. Exit: Opens a dialog to save or discard and exit.

C.1.2. Edit Workplace (Fig. C.2a-e)

This window offers functions for the creation of new workplace setups and the adaptation of already existing template setups. Furthermore, new robotic structures can be built. All devices that are present in the workplace can be included in the virtual world and prepositioned according to the user's preferences. Wavefront objects can directly be loaded into the viewer. Additionally, robots with serial kinematic chains can be loaded. To better visualize the workspace of the used robots, these can be moved both in Cartesian space and in joint space. To move the robots in Cartesian space, an appropriate inverse kinematics algorithm has to be chosen. Among others, the following functions are supported:

- Load/delete, show/hide and move objects
- Load/delete, show/hide, define and move (both in Cartesian and joint space) robots
- Choose adequate inverse kinematics algorithms for each robot, including optimization criteria for redundancy resolution
- Browse through the patient specific DICOM[2]-slices (if available)
- Switch between different view angles, e.g. view through the endoscope

All available functions are described in the following.

Common Functions and Positioning Functions (Fig. C.2a):

1. Home position view: view the scene from a standard view position that can be defined in the data file.
2. TCP position view: if there is an object called "camera", view the scene through it. Opening angle and optical center of the camera can be defined in the data file.
3. "Snap" robot to workspace: if a workspace is already defined, inverse kinematics is used to position the TCP of the active robot in the center of the workspace (without moving the robot base). If different tasks with different workspaces are defined, the first task in the list is chosen.
4. Show redundant motion: if the robot is a 7-DoF robot, the Nullspace motion of this robot is shown according to the current position.
5. If valid inverse kinematics are chosen for a robot, this button triggers whether the robot base or the robot TCP is moved in Cartesian space using the sliders 15 and 16.

[2] DICOM: Digital Imaging and Communications in Medicine

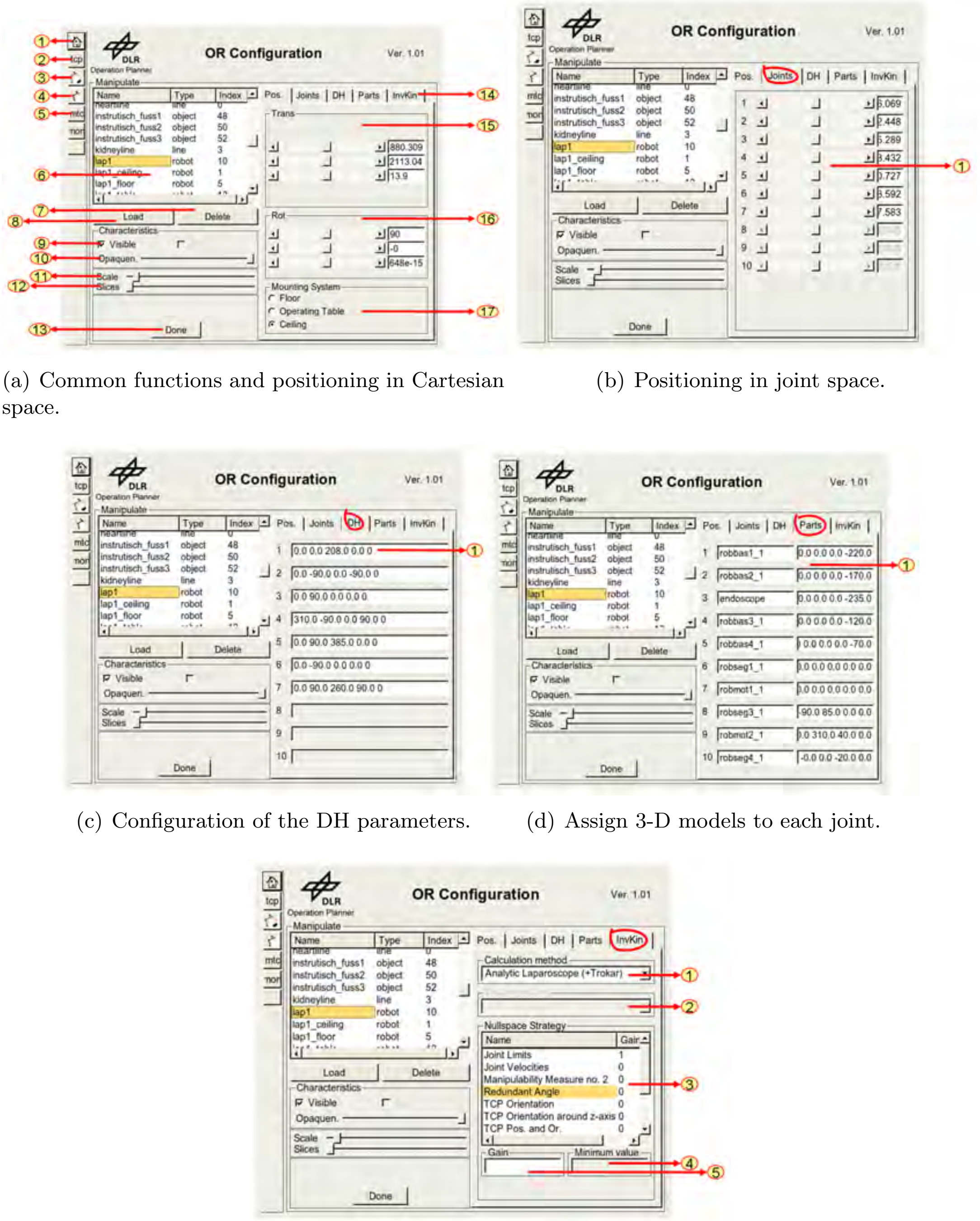
6. A list shows all existing entities in the current scene. This comprises objects, robots and lines. An entity can be highlighted by click and then edited using the remaining functions of the window. Each robot is composed of different objects that can be accessed by double-click on the resp. robot.
7. The active entity can be deleted.
8. Both robots and objects can individually be loaded.
9. The visibility state of each entity can be toggled.
10. For each object and robot, the transparency can be adjusted.
11. The range of each slider can be adjusted.
12. If slice data is available (e.g. from DICOM images), one can browse through these slices. The slices are shown directly in the scene corresponding to the 3D model of the patient.
13. Go back to the main window once prepositioning is done.
14. The right part of the window offers different functions grouped in separate sections. This comprises:
    - Positioning of the TCP and the robot base in Cartesian space (Fig. C.2a, sliders 15 and 16),
    - control of the robot pose in joint space (Fig. C.2b),
    - definition of the DH parameters of the robot (Fig. C.2c),
    - assignment of objects to each robot segment (Fig. C.2d), and
    - determination of the inverse kinematics algorithm and redundancy resolution (Fig. C.2d).
15. Sliders to control the translational position in Cartesian space.
16. Sliders to control the rotational position in Cartesian space.

Joint Space Positioning (Fig. C.2b):

1. The robot joint space can be controlled using sliders and input fields for each joint.

DH Parameters of the Robot (Fig. C.2c):

1. Each joint of the (serial) robot is defined through its DH parameters (convention of Craig [19]). Furthermore, the initial joint angle of the robot as well as the type of joint (rotation or translation) can be set.



Manipulate

Name	Type	Index
heartline	line	0
instrutisch_fuss1	object	48
instrutisch_fuss2	object	50
instrutisch_fuss3	object	52
kidneyline	line	3
lap1	robot	10
lap1_ceiling	robot	1
lap1_floor	robot	5

Load | Delete

Characteristics: ☑ Visible □ ; Opaquen.; Scale; Slices

Done

Pos. | Joints | DH | Parts | InvKin

Trans: 880.309; 2113.04; 13.9

Rot: 90; -0; 648e-15

Mounting System: ○ Floor ○ Operating Table ● Ceiling

(a) Common functions and positioning in Cartesian space.

Joints: 1 6.069; 2 2.448; 3 5.289; 4 8.432; 5 0.727; 6 8.592; 7 7.583; 8; 9; 10

(b) Positioning in joint space.

DH:
1 0.0 0.0 208.0 0.0 0
2 0.0 -90.0 0.0 -90.0 0
3 0.0 90.0 0.0 0.0 0.0 0
4 310.0 -90.0 0.0 90.0 0
5 0.0 90.0 385.0 0.0 0
6 0.0 -90.0 0.0 0.0 0.0
7 0.0 90.0 260.0 90.0 0
8
9
10

(c) Configuration of the DH parameters.

Parts		
1	robbas1_1	0.0 0.0 0.0 -220.0
2	robbas2_1	0.0 0.0 0.0 -170.0
3	endoscope	0.0 0.0 0.0 -235.0
4	robbas3_1	0.0 0.0 0.0 -120.0
5	robbas4_1	0.0 0.0 0.0 -70.0
6	robseg1_1	0.0 0.0 0.0 0.0 0.0
7	robmot1_1	0.0 0.0 0.0 0.0 0.0
8	robseg3_1	-90.0 85.0 0.0 0.0
9	robmot2_1	0.0 310.0 40.0 0.0
10	robseg4_1	-0.0 0.0 -20.0 0.0

(d) Assign 3-D models to each joint.

Calculation method: Analytic Laparoscope (+Trokar)

Nullspace Strategy

Name	Gain
Joint Limits	1
Joint Velocities	0
Manipulability Measure no. 2	0
Redundant Angle	0
TCP Orientation	0
TCP Orientation around z-axis	0
TCP Pos. and Or.	0

Gain; Minimum value

(e) Assign inverse kinematics and criteria for redundancy resolution.

Figure C.2.: Edit workplace.

Parts of the Robot (Fig. C.2d):

1. To each joint, an object can be connected through an adjustable transformation matrix.

Inverse Kinematics and Redundancy Resolution (Fig. C.2e):

1. Adequate inverse kinematics algorithms can be set.
2. In case trocar kinematics have to be included, the location i of the trocar (the trocar thus is positioned between joint i and $i+1$) can be chosen.
3. Optimization criteria for redundancy resolution and inverse kinematics calculation can be adjusted.
4. The gain for each optimization criterion can be chosen.
5. A minimum value can be given as constraint. In case no solution is found that achieves a better performance than the minimum value, the inverse kinematics return an error.

C.1.3. Definition of Optimization (Fig. C.3a-d)

This window offers functionality to define the optimization problem. Therefore,

- accessible parameters for optimization are chosen and bordered,
- the workspace and entry point locations for each robot in each task is defined by the user, and
- the optimization criteria are chosen and weighted.

General Window Description and Task Definition (Fig. C.3a):

1. The upper list shows currently defined tasks and the order of execution. A task can be highlighted and then edited.
2. The order of tasks can be modified, tasks can be added and removed.
3. The second list shows the objects and robots whose parameters are subject to optimization in the active task resp. whose motion is considered.
4. Objects and robots can be added to or deleted from the list.
5. Editing of task definition, optimization parameter definition, optimization criteria definition and task refinement can be chosen.

6. For each robot in each task, a workspace can be chosen.
7. To define a workspace, in a first step a line has to be created. This can be done by drawing on a target object chosen here.
8. Select this to start drawing on the selected object.
9. Through the *undo* button, the last drawn point and normal is removed. The *create* button creates a closed line from the points drawn by the user. The *show* button shows the final workspace, depending on the drawn line and the chosen shape.
10. Similar functions are available to define the accessible surface for port locations.
11. The currently defined optimization problem is shown in a summary.

Definition of Optimization Parameters (Fig. C.3b):

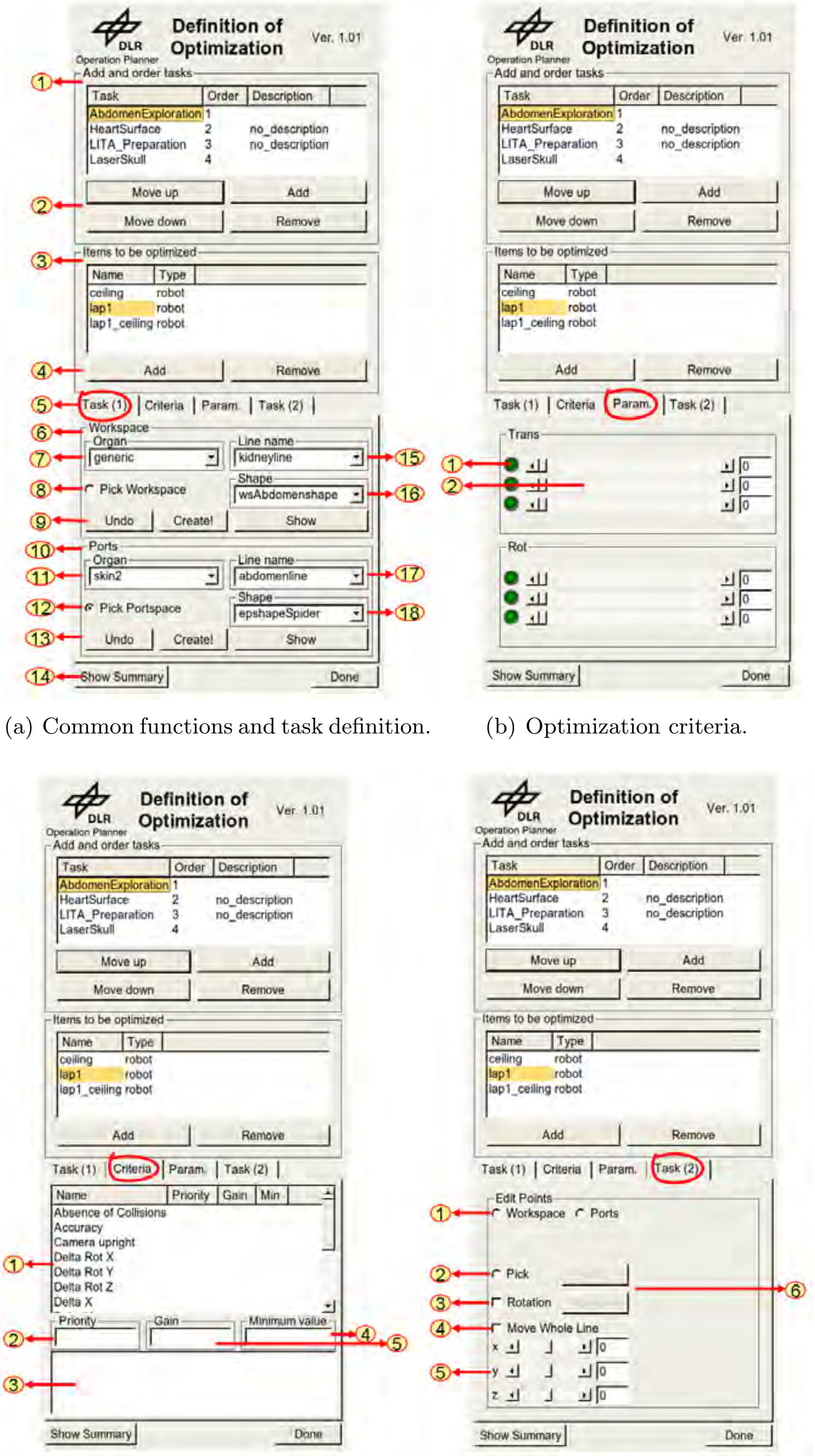
1. For each item to be optimized, the accessible parameters are shown. The green light indicates that the respective parameter is subject to optimization. The window shows those parameters accessible for objects. Robots additionally provide the option to optimize their DH parameters.
2. The initial value as well as the accessible range for optimization can be adjusted.

Definition of Optimization Criteria (Fig. C.3c):

1. Optimization criteria can be defined here.
2. Grouping of optimization criteria into priority classes has not yet been implemented.
3. A short description of the currently active optimization criterion is shown.
4. A minimum value can be given.
5. Each optimization criterion can be weighted.

Task Refinement (Fig. C.3d):

1. Fine tuning of either the line currently used for workspace or port surface definition.
2. Choose pick to mark a point and normal in the line.
3. Switch between rotation or translation of the currently active point and normal.
4. Choose to either move the whole line or a single point and normal.
5. Sliders and inputs to adjust the position and orientation.



Definition of Optimization Ver. 1.01

Add and order tasks

Task	Order	Description
AbdomenExploration	1	
HeartSurface	2	no_description
LITA_Preparation	3	no_description
LaserSkull	4	

Items to be optimized

Name	Type
ceiling	robot
lap1	robot
lap1_ceiling	robot

Workspace: Organ: generic; Line name: kidneyline; Shape: wsAbdomenshape

Ports: Organ: skin2; Line name: abdomenline; Shape: epshapeSpider

(a) Common functions and task definition.

Trans: 0, 0, 0

Rot: 0, 0, 0

(b) Optimization criteria.

Name	Priority	Gain	Min
Absence of Collisions			
Accuracy			
Camera upright			
Delta Rot X			
Delta Rot Y			
Delta Rot Z			
Delta X			

(c) Parameters to be optimized.

Edit Points: ○ Workspace ○ Ports

○ Pick

□ Rotation

□ Move Whole Line

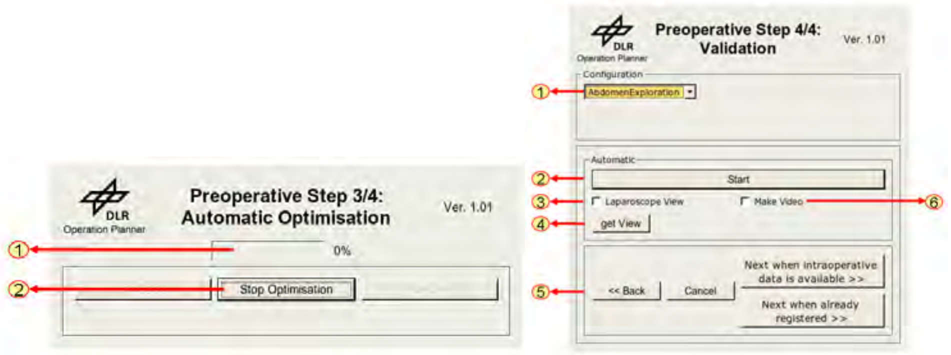
x 0

y 0

z 0

(d) Refinement of task definition.

Figure C.3.: Definition of the optimization problem.



(a)

(b)

Figure C.4.: Optimization (a) and validation (b).

Automatic Optimization (Fig. C.4a)

1. The progress is shown during optimization.
2. The optimization can be stopped at any time. The best found solutions so far are then used in the subsequent steps.

Validation (Fig. C.4b)

1. Choose the task to be validated among those previously defined.
2. Start automatic validation: The TCP positions and orientations used during optimization are shown automatically.
3. Choose between Camera view and observer view.
4. Print current view position as a homogeneous transformation matrix to the screen.
5. The OK-Button leads back to the main menu.

C.1.4. Workplace Optimizer Wizard File Format

During data load, the user chooses

- a template consisting of the coarse structure of the operation, including e.g. the necessary intervention steps, the used instruments and robots and a coarse positioning of all equipment in the virtual operating room, and

- patient specific data, consisting of either DICOM images or already segmented 3-D surfaces of the patient structures.

Currently, the template data structure is organized in a plain ASCII file. Saving the status of the planning is possible at all times and thus allows the user to interrupt and close the software without loss of data.

C.2. The Geoserver Viewer

For visualization, a viewer based on OpenGL is used that was developed at DLR by Holger Weiss. The viewer can be started as a standalone program (the *geoserver*) with the following command:

geoserver <options>

The following options are available:

- `<file=X>`
  X denotes the file that describes the scene. If no file is given, an empty scene is loaded.
- `<silent>`
  The *geoserver* is started without graphical output. This can be useful since the *geoserver* can also be used for collision detection (based on the library I-COLLIDE [17]). In this case graphical output may not be needed or wished.
- `<stereo>`
  Provide a stereo view of the scene.
- `<mono>`
  Provide a mono view of the scene (default).
- `<local>`
  Set up a local connection to other software connecting to the *geoserver* via the C-application programming interface (API) (all software runs on the same PC).
- `<inet port=X>`
  Provides a connection for software running on remote PCs, specifying a communication port.
- `<sockfile=X>`
  Set the name and location of the *socketfile* for communication.
- `<posx=X posy=X width=X height=X>`
  Determines position and size of the *geoserver* graphical output window.

- `<help>`
  Shows all available options.

The *geoserver* can be controlled by key commands:

- F1: Toggle between wireframe/filled mode
- F2: Switch to full screen
- F9: Increase eye distance for stereo mode
- F10: Decrease eye distance for stereo mode
- F11: Increase eye angle for stereo mode
- F12: Decrease eye angle for stereo mode
- ESC: Exit *geoserver* and close window

Furthermore, an API is provided in C:

Connect and disconnect to the geoserver

- `int connectToLocalServer(char *sockfile);`
  Connect to a *geoserver* running locally
- `int connectToInetServer(char *sockfile, char *port);`
  Connect to a remote *geoserver*
- `int closeConnection(void);`
  Close connection to currently connected *geoserver*
- `int selectConnection(int conn);`
  Select *geoserver* connection in case several *geoservers* are running
- `void sendQuit(void);`
  Close *geoserver*
- `void sendShutDown(void);`
  Close all connections

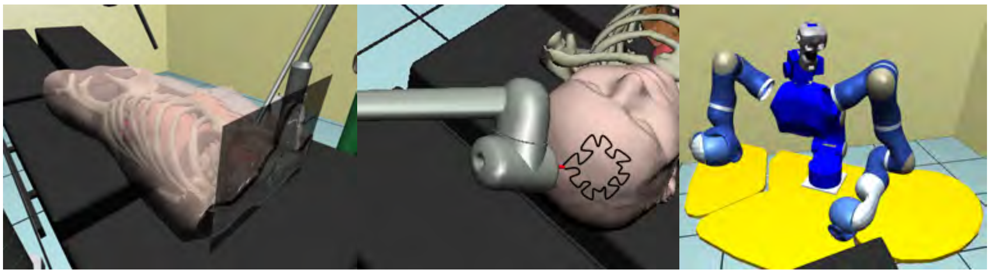


Figure C.5.: Screenshots of the *geoserver* graphical output.

Choose viewer attributes

- `void sendSetLightPos(double px, double py, double pz);`
  Set the position of the light source in the scene
- `void sendSetClipPlanes(double near, double far);`
  Set the clipping planes
- `int sendGetBase(double mat[4][4]);`
  Get the base coordinate system
- `void sendSetBase(double mat[4][4]);`
  Set the base coordinate system
- `void sendSetViewAngle(double angle, double fovy);`
  Change view angle

Load and delete entities

Three entities are known to the *geoserver*: *Objects* contain the surface data (loaded from wavefront files), *robots* define connections between objects, and *lines* define an editable sequence of points (with normals) in space. Lines are shown in the *geoserver* as connections between subsequent points (see Fig. C.5 for lines, objects and robots).

- `void sendLoadObject(ObjectData *data);`
  Load an object into the viewer (ObjectData contains e.g. the path and name of the resp. wavefront file)
- `void sendDeleteObject(char *name);`
  Delete an object
- `void sendLoadRobot(RobotData *data);`
  Load a robot, consisting of e.g. the definition of links between objects

- `void sendDeleteRobot(char *name);`
  Delete a robot (removes the links between objects used in that robot)
- `void sendLoadLine(LineData *data);`
  Load a new line
- `void sendDeleteLine(char *name);`
  Delete a line

Change/provide entity attributes

- `void sendGetNames(char **objects, char **robots, char **lines);`
  Provides lists of the names of all objects, robots and lines currently loaded into the *geoserver*
- `int sendGetObject(char *name, double mat[4][4]);`
  Gives back the current position of the object asked for with *name*
- `void sendSetObject(char *name, double mat[4][4]);`
  Sets the position of the object *name*
- `int sendGetOpaqueness(char *name, double *opaque);`
  Gives back the opaqueness of the object asked for with *name*
- `void sendSetOpaqueness(char *name, double opaque);`
  Sets the opaqueness of the object *name*
- `int sendGetMaterial(char *name, char *mat, float *ns,`
  `float *kd, float *ks, float *ka);`
  Gives back the material properties of the object asked for with *name*
- `void sendSetMaterial(char *name, char *mat, float ns,`
  `float *kd, float *ks, float *ka);`
  Sets the material of the object *name*
- `int sendGetRobot(char *name, int *num, double *phi);`
  Gives back the joint angles of the robot asked for with *name*
- `void sendSetRobot(char *name, char num, double *phi);`
  Sets the joint angles of the robot *name*
- `void sendShowObject(char *name);`
  Makes the object *name* visible
- `void sendHideObject(char *name);`
  Makes the object *name* invisible

Change mouse behavior

- `int sendGetMouseSens(double *sens);`
  Get mouse sensitivity

- `void sendSetMouseSens(double *sens);`
  Set mouse sensitivity

- `void sendEnablePointPick(char *object, char *line);`
  Enables *PointPick:* With mouse clicks the positions on the surface of the *object* can be selected. Point coordinates are then added to the *line*

- `void sendDisablePointPick(void);`
  Disables PointPick

- `void sendEnableObjSelect(void);`
  Enables *ObjectSelect*: The object visible at the mouse position is selected

- `void sendDisableObjSelect(void);`
  Disable ObjectSelect

- `int sendGetSelObject(int *selected, char *name, char *mat, double *point, double *normal);`
  Gives back the currently selected object name and position. Furthermore, the position of the mouse pointer on the object and the normal direction of this point are returned

Line manipulation

- `void sendAddPoint(char *name, double *point, double *normal);`
  Adds a point (with normal) to the line *name*

- `void sendRemovePoint(char *name);`
  Removes the last added point from the line *name*

- `void sendCloseLine(char *name);`
  Connects the first with the last point in line *name*

- `void sendSaveLine(char *name, char *fname);`
  Saves the line data in a file *fname.*

- `void sendShowLine(char *name);`
  Makes the line *name* visible

- `void sendHideLine(char *name);`
  Makes the line *name* invisible

Collision detection

- `int sendCheckColl(char *name1, char *name2, int *ncoll);`
  Checks if the objects *name1* and *name2* collide

Video output

- `void sendDumpWindow(void);`
  Saves a snapshot of the *geoserver* window into the system temp folder. Using this routine continuously, a video can be created from the snapshots.

C.3. The Registration Tool

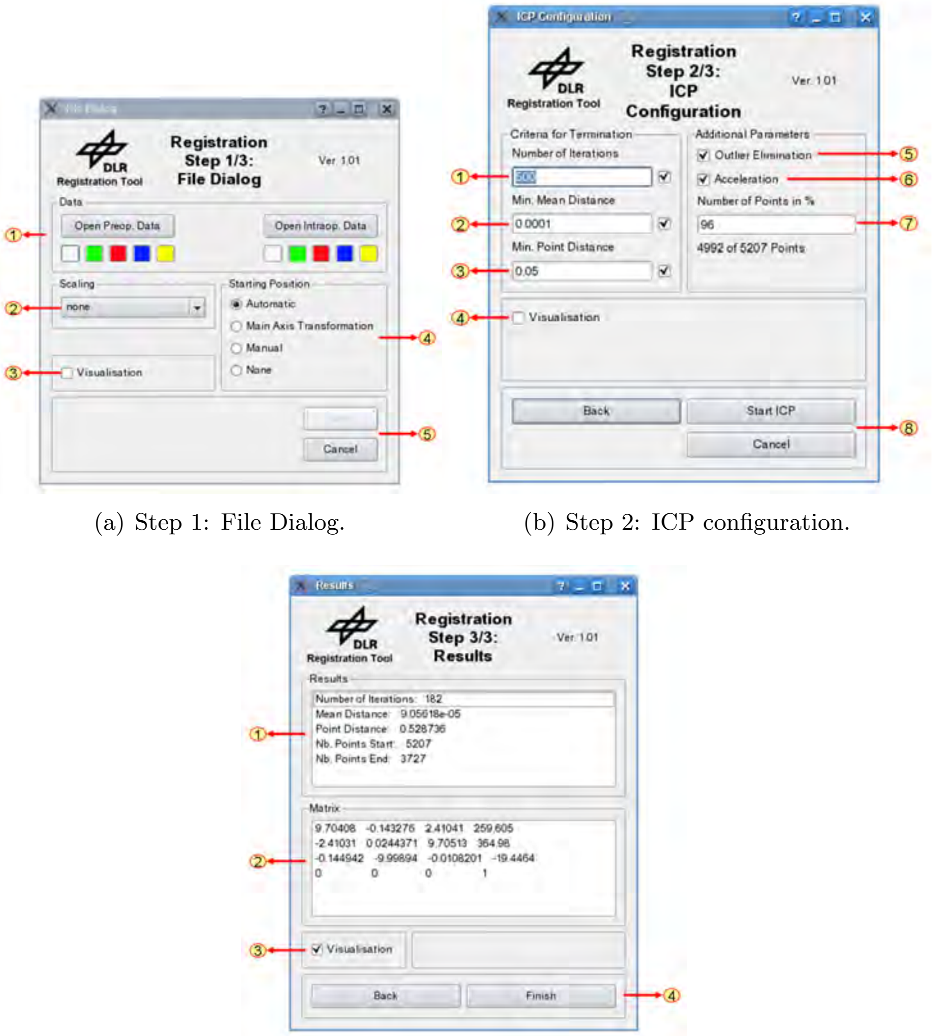
A registration software (the *Registration Tool* implemented at DLR by Michael Mayr [69]) is integrated in the *Workplace Optimizer Wizard*. The *Registration Tool* is started as a standalone program with the command:

matching <arg1> <arg2> <arg3> <arg4> <arg5> <arg6>.

The following arguments can be added:

- <arg1>: Preoperative surface data
- <arg2>: Intraoperative surface data
- <arg3>: Path and file name of a file to save the results
- <arg4>: Scaling factor for the preoperative data
- <arg5>: Automatic mode, the *Registration Tool* either runs (1) until the window *Results* (Fig. C.6(c)) is reached, or (2) completely automatically and closes upon completion.
- <arg6>: Offset values $\begin{pmatrix} x & y & z & x_{\mathrm{rot}} & y_{\mathrm{rot}} & z_{\mathrm{rot}} \end{pmatrix}^{\mathrm{T}}$ for manual initial adjustment of translations $\begin{pmatrix} x & y & z \end{pmatrix}^{\mathrm{T}}$ and rotations $\begin{pmatrix} x_{\mathrm{rot}} & y_{\mathrm{rot}} & z_{\mathrm{rot}} \end{pmatrix}^{\mathrm{T}}$ of the preoperative data can be given.

The *Registration Tool* is structured in three subsequent windows as shown in Fig. C.6. The following functions are available, according to the numbering in Fig. C.6(a)-(c).



Scaling: none

□ Visualisation

Starting Position:
● Automatic
○ Main Axis Transformation
○ Manual
○ None

(a) Step 1: File Dialog.

Criteria for Termination

Number of Iterations: 500 ☑

Min. Mean Distance: 0.0001 ☑

Min. Point Distance: 0.05 ☑

Additional Parameters

☑ Outlier Elimination

☑ Acceleration

Number of Points in %: 96

4992 of 5207 Points

□ Visualisation

(b) Step 2: ICP configuration.

Results

```
Number of Iterations:  182
Mean Distance:  9.05618e-05
Point Distance:  0.528736
Nb. Points Start:  5207
Nb. Points End:  3727
```

Matrix

```
9.70408    -0.143276   2.41041     259.605
-2.41031   0.0244371   9.70513     364.98
-0.144942  -9.99894    -0.0108201  -19.4464
0          0           0           1
```

☑ Visualisation

(c) Step 3: Results.

Figure C.6.: The ICP *Registration Tool*.

File Dialog (Fig. C.6a)

In this dialog, the datasets are loaded, scaled, and an initial adjustment procedure is chosen:

1. The surface data for both pre- and intraoperative situation is loaded. The surface data has to be in the wavefront, VRML or vtk format.
2. In case the data are in different scale, a scaling factor can be used.
3. The data can be visualized in a common COS. This way, it can be verified that e.g. the scaling is right.
4. The ICP algorithm is known to require coarse initial adjustment [8], i.e. it often fails if the rotations and translations necessary to register the two datasets are too large[3]. Therefore different methods for initial adjustment are implemented for the *Registration Tool*:

   a) Automatic mode: The preoperative data are translated such that the centroids of pre- and intraoperative data coincide. Then, a series of 24 ICP registrations with very few iterations is performed, each run with a different rotation of the preoperative data. The best fitting result is used as initial adjustment.

   b) Main axis transformation: The preoperative data are translated such that the centroids of pre- and intraoperative data coincide. Furthermore, points defining the surface are interpreted as point masses with unity mass, and thus the principal moments of inertia are computed. The preoperative data are rotated such that the axes of the principal moments of inertia of pre- and intraoperative data are corresponding to achieve an initial adjustment.

   c) Manual: Assisted by visualization, the preoperative data can be translated and rotated manually.

   d) None: No initial adjustment of the datasets is done.
5. The *Registration Tool* can be closed, or it can be proceeded with the next step.

ICP Configuration (Fig. C.6b)

The following stop criteria for the ICP algorithm are available and can be activated individually:

1. Number of iterations: Stop after a certain number of iterations.

[3] It is not possible to quantify a general value for the radius of convergence since it depends on many parameters such as the shape of the surface represented by the datasets, the number of points, or the overlap of the two datasets.

2. Min. mean distance: Stop if the minimal mean distance improves less than the given value. The minimal mean distance specifies the improvement between two iteration steps regarding the mean distance of all surface points.
3. Min. point distance: Stop if the mean distance between points in pre- and intraoperative data is lower than the given value.

The algorithm stops if either one of the activated criteria is fulfilled. Additionally, the following options can be chosen in the dialog:

4. Visualization can be switched on or off.
5. Outlier elimination: In case some points in one dataset do not have a corresponding point in the other dataset, these points are eliminated.
6. Acceleration: As described in [8], a method to speed up the standard ICP algorithm implemented in the VTK toolkit [92] is included into the *Registration Tool*.
7. Number of points: The number of points used for registration and thus the computation time can be reduced.
8. The *Registration Tool* can be closed, it can be returned to the previous dialog, or it can be proceeded with the next step.

Results (Fig. C.6c)

The results of the ICP registration are displayed as follows:

1. General Information:
   a) Number of iterations: Number of iterations performed to reach the result.
   b) Mean distance: Improvement in the last iteration regarding the mean distance of corresponding points in pre- and intraoperative data.
   c) Point distance: Mean distance between corresponding points in pre- and intraoperative data after registration.
   d) Number of points at Start/End: In case outlier elimination is activated, the number of considered points before and after registration is displayed.
2. Transformation matrix: The preoperative dataset can be registered with the intraoperative dataset by applying the shown transformation.
3. Visualization: The registered datasets are visualized.
4. The *Registration Tool* can be closed, or it can be returned to the previous window.

Bibliography

[1] L. Adhami. *An Architecture for Computer Integrated Mini-Invasive Robotic Surgery: Focus on Optimal Planning.* Ph.D. Thesis, Ecole des Mines de Paris, Paris, 2002.

[2] L. Adhami and È. Coste-Manière. Optimal planning for minimally invasive surgical robots. *IEEE Transactions on Robotics and Automation: Special Issue on Medical Robotics*, 19(5):854–863, 2003.

[3] A. Albu-Schäffer, S. Haddadin, Ch. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger. The DLR Lightweight Robot – Lightweight Design and "Soft Robotics" Control Concepts for Robots in Human Environments. *Industrial Robot Journal*, 34(5), 2007.

[4] S. Ayari, D. Abedipour, D. Bossard, and P. Fröhlich. CT-Assisted Surgery in Choanal Atresia. *Acta Oto-Laryngologica*, 124(4):502–504, 2004.

[5] B. Baginski. *Motion Planning for Manipulators with Many Degrees of Freedom – The BB-Method.* PhD thesis, Munich, Germany, 1998.

[6] J.-C. Baillie. Grouding Symbols in Perception with Two Interacting Autonomous Robots. In *Proceedings of the Fourth International Workshop on Epigenetic Robotics*, Genoa, Italy, 2004.

[7] J.-C. Baillie and M. Nottale. Dynamic Evolution of Language Games between Two Autonomous Robots. In *Proceedings of the IEEE International Conference on Development and Learning*, pages 68–73, Osaka, Japan, 2005.

[8] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.

[9] T. Bodenmüller and G. Hirzinger. Online Surface Reconstruction from Unorganized 3D-Points for the DLR Hand-Guided Scanner System. In *2nd Symposium on 3D Data Processing, Visualization and Transmission*, pages 285 – 292, Thessaloniki, Greece, 2004.

[10] D. Burschka and G. D. Hager. V-GPS(SLAM): Vision-Based Inertial System for Mobile Robots. In *Proc. of ICRA*, pages 409–415, Barcelona, Spain, 2004.

[11] G. B. Cadière and J. Leroy. Principes Généraux de la Chirurgie Laparoscopique. *Encycl Méd Chir (Techniques chirurgicales - Appareil digestif)*, 40, 1999.

[12] J. W. Cannon, J. A. Stoll, S. D. Selha, P. E. Dupont, R. D. Howe, and D. F. Torchiana. Port Placement Planning in Robot-Assisted Coronary Artery Bypass. *IEEE Transactions on Robotics and Automation: Special Issue on Medical Robotics*, 19(5):912 – 917, 2003.

[13] C. G. L. Cao, C. L. MacKenzie, and S. Payandeh. Task and Motion Analyses in Endoscopic Surgery. In *1996 ASME IMECE Conference Proceedings: 5th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Atlanta, Georgia, USA, 1996.

[14] A. M. Chiu, D. Dey, M. Drangova, D. Boyd, and T. M Peters. 3-D Image Guidance for Minimally Invasive Robotic Coronary Artery Bypass. *The Heart Surgery Forum*, 3(3):224–231, 2000.

[15] R. Cichon, U. Kappert, J. Schneider, I. Schramm, V. Gulielmos, S. Tugtekin, and S. Schüler. Robotically Enhanced "Dresden Technique" with Bilateral Mammary Artery Grafting. *Thorac Cardiovasc Surg*, 48(4):189–192, 2000.

[16] R. Clavel. DELTA, a Fast Robot with Parallel Geometry. In *18th Int. Symp. on Industrial Robot*, pages 91 – 100, Lausanne, Switzerland, 1988.

[17] J. D. Cohen, M. C. Lin, D. Manocha, and M. Ponamgi. I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scaled Environments. In *Proc. ACM Symposium on Interactive 3D Graphics*, pages 189–196, Monterey, Canada, 1995.

[18] È. Coste-Manière, L. Adhami, F. Mourgues, and O. Bantiche. Optimal Planning of Robotically Assisted Heart Surgery: Transfer Precision in the Operating Room. In *8th International Symposium on Experimental Robotics (ISER)*, pages 424–434, Sant'Angelo d'Ischia, Italy, 2002.

[19] J. J. Craig. *Introduction to Robotics*. Addison-Wesley, Reading, MA, USA, 1989.

[20] A. Dalgalarrondo, D. Dufourd, and D. Filliat. Controlling the Autonomy of a Reconnaissance Robot. In *SPIE Defense and Security 2004 Symposium. Unmanned Ground Vehicle Technology VI Conference*, pages 314–325, Orlando, Florida, USA, 2004.

[21] M. Dastani, F. de Boer, F. Dignum, W. van der Hoek, M. Kroese, and J. J. Meyer. Programming the Deliberation Cycle of Cognitive Robots. In *Proceedings of The Third International Cognitive Robotics Workshop*, Edmonton, Canada, 2002.

[22] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[23] J. J. DeRose, R. C. Ashton, S. Belsley, D. G. Swistel, M. Vloka, F. Ehlert, R. Shaw, J. Sackner-Bernstein, Z. Hillel, and J. S. Steinberg. Robotically Assisted Left Ventricular Epicardial Lead Implantation for Biventricular Pacing. *J American College Cardiology*, 41(8):1414–1419, 2003.

[24] G. Eggers, G. Sudra, L. Kahrs, S. Ghanai, J. Raczkowsky, R. Dillmann, H. Wörn, J. Mühling, and R. Marmulla. A Comparision of Intraoperative Augmented Reality Systems. *Biomedizinische Technik*, 50(1):907–908, 2005.

[25] D. Engel, W. Korb, J. Raczkowsky, S. Hassfeld, and H. Wörn. Location Decision for a Robot Milling Complex Trajectories in Craniofacial Surgery. In *Proceedings of the 17th International Congress and Exhibition CARS 2003*, pages 760–765, London, UK, 2003.

[26] V. Falk. Robotic Technology in Cardiac Surgery. *Applied Cardiopulmonary Pathophysology*, 10(1):31–38, 2001.

[27] V. Falk. Manual Control and Tracking – A Human Factor Analysis Relevant for Beating Heart Surgery. *Annals of Thoracic Surgery*, 74(2):624–628, 2002.

[28] V. Falk, A. Diegeler, T. Walther, N. Löscher, B. Vogel, C. Ulmann, T. Rauch, and F. W. Mohr. Endoscopic Coronary Artery Bypass Grafting on the Beating Heart Using a Computer Enhanced Telemanipulation System. *Heart Surg Forum*, 2(6):199–205, 1999.

[29] R. Fletcher. *Practical Methods of Optimization; (2nd ed.).* Wiley-Interscience, New York, NY, USA, 1987.

[30] S. Frumento, R. Michelini, R. Konietschke, U. Hagn, T. Ortmaier, and G. Hirzinger. A Co-Robotic Positioning Device for Carrying Surgical End-Effectors. In *Proceedings of ESDA 2006*, Torino, Italy, 2006.

[31] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns, Elements of Reusable Object-Oriented Software.* Addison-Wesley Professional Computing Series. Addison-Wesley, Boston, San Francisco, New York, 1995.

[32] F. R. Gantmacher. *The Theory of Matrices.* American Mathematical Society, New York, USA, 1959.

[33] M. Ghodoussi, S. E. Butner, and Y. Wang. Robotic Surgery – The Transatlantic Case. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 1882 – 1888, Washington, DC, USA, 2002.

[34] G. S. Guthart and K. Salisbury. The Intuitive Telesurgery System: Overview and Application. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 618 – 621, San Francisco, CA, USA, 2000.

[35] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger. Safety Evaluation of Physical Human-Robot Interaction via Crash-Testing. In *Proceedings of Robotics: Science and Systems Conference (RSS2007)*, Atlanta, USA, 2007.

[36] F. Hao and J.-P. Merlet. Multi-Criteria Optimal Design of Parallel Manipulators Based on Interval Analysis. *Mech. Mach. Theory*, 40(2):157–171, 2005.

[37] S. Hassfeld and J. Mühling. Computer Assisted Oral and Maxillofacial Surgery a Review and an Assessment of Technology. *International Journal for Oral Maxillofacial Surgery*, 30(1):2–13, 2001.

[38] G. Hirzinger, A. Albu-Schäffer, M. Hähnle, I. Schaefer, and N. Sporer. On a New Generation of Torque Controlled Light-weight Robots. In *IEEE International Conference of Robotics and Automation*, pages 3356–3363, Seoul, Korea, 2001.

[39] J.-S. Hong, T. Dohi, M. Hashizume, K. Konishi, and N. Hata. A Motion Adaptable Needle Placement Instrument Based on Tumor Specific Ultrasonic Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002: 5th International Conference, Proceedings*, pages 122–129, Tokyo, Japan, 2002.

[40] L. Hotraphinyo and C. Riviere. Three-Dimensional Accuracy Assessment of Eye Surgeons. In *Proc. 23rd Annual Intl. Conf. IEEE Engineering in Medicine and Biology Society*, pages 3458–3461, Istanbul, Turkey, 2001.

[41] Intuitive Surgical Inc. The Intuitive Telesurgery System DaVinci. [web page] http://www.intuitivesurgical.com. [Accessed on April 18th, 2007.].

[42] M. Ivanenko, R. Sader, S. Afilal, M. Werner, M. Hartstock, C. von Hänisch, S. Milz, W. Erhardt, H.-F. Zeilhofer, and P. Hering. In-vivo Animal Trials with a scanning CO2 Laser Osteotome. *Laser in Surgery and Medicine*, 37(2):144–148, 2005.

[43] M. Jakopec, S. J. Harris, F. R. y Baena, P. Gomes, J. Cobb, and B. L. Davies. Preliminary Results of an Early Clinical Experience with the Acrobot System for Total Knee Replacement Surgery. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002: 5th International Conference, Proceedings*, pages 256–263, Tokyo, Japan, 2002.

[44] L. A. Kahrs, J. Raczkowsky, and H. Wörn. Aufbau eines miniaturisierten Laserprojektors für die Erweitere Realität in der Chirurgie. In *Proceedings der Gemeinsame Jahrestagung der Deutschen, Österreichischen und Schweizerischen Gesellschaft für Biomedizinische Technik*, pages 1–2, Zurich, Switzerland, 2006.

[45] U. Kappert, R. Cichon, J. Schneider, V. Gulielmos, S. Tugtekin, K. Matschke, I. Schramm, and S. Schüler. Robotic Coronary Artery Surgery: The Evolution of a New Minimally Invasive Approach in Coronary Artery Surgery. *Thorac Cardiovasc Surg*, 48(4):193–197, 2000.

[46] L. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces. *IEEE Journal on Rob. and Aut.*, 12(4):566–580, 1996.

[47] R. Konietschke. *Aufbauoptimierung für Roboter in medizinischen Anwendungen.* Munich University of Technology, Munich, Germany, 2001. Diploma Thesis.

[48] R. Konietschke, A. Busam, T. Bodenmüller, T. Ortmaier, J. Wiechnik, M. Suppa, T. Welzel, G. Eggers, G. Hirzinger, and R. Marmulla. Potential, Limitations and Challenges of Markerless Registration with the DLR 3D-Modeller in Medical Applications. In *Proceedings of the 21st International Congress and Exhibition of Computer Assisted Radiology and Surgery*, Berlin, Germany, 2007.

[49] R. Konietschke, S. Frumento, T. Ortmaier, U. Hagn, and G. Hirzinger. Kinematic Design Optimization of an Actuated Carrier for the DLR Multi-Arm Surgical System. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2006*, pages 4381–4387, Beijing, China, 2006.

[50] R. Konietschke, G. Hirzinger, and Y. Yan. All Singularities of the 9-DoF DLR Medical Robot Setup for Minimally Invasive Applications. In *Advances in Robot Kinematics*, pages 193–200, Ljubljana, Slovenia, 2006.

[51] R. Konietschke, A. Knöferle, and G. Hirzinger. The Autopointer: A New Augmented-Reality Device for Transfer of Planning Data into the Operating Room. In *Proceedings of the 21st International Congress and Exhibition of Computer Assisted Radiology and Surgery*, Berlin, Germany, 2007.

[52] R. Konietschke, T. Ortmaier, C. Ott, U. Hagn, L. Le-Tien, and G. Hirzinger. Concepts of Human-Robot Cooperation for a new Medical Robot. In *Proceedings of Human Centered Robotic Systems – HCRS*, pages 1–6, Munich, Germany, 2006.

[53] R. Konietschke, T. Ortmaier, H. Weiß, R. Engelke, and G. Hirzinger. Optimal Design of a Medical Robot for Minimally Invasive Surgery. In *2. Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, Nürnberg, Germany, 2003.

[54] R. Konietschke, T. Ortmaier, H. Weiß, G. Hirzinger, and R. Engelke. Manipulability and Accuracy Measures for a Medical Robot in Minimally Invasive Surgery. In *Advances in Robot Kinematics*, Genua, Italy, 2004.

[55] R. Konietschke, H. Weiß, T. Ortmaier, U. Hagn, Y. Yan, and G. Hirzinger. A Registration-Free Intraoperative Procedure to Optimally Position the Ports and the Robots in Robotically Assisted Minimally Invasive Surgery. In *4. Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, Berlin, Germany, 2005.

[56] R. Konietschke, H. Weiß, T. Ortmaier, and G. Hirzinger. A Preoperative Planning Procedure for Robotically Assisted Minimally Invasive Interventions. In *3. Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, München, Germany, 2004.

[57] W. Korb, T. Bodenmüller, G. Eggers, T. Ortmaier, M. Schneberger, M. Suppa, J. Wiechnik, R. Marmulla, and S. Hassfeld. Surface-Based Image-to-Patient-Registration Using a Hand-Guided Laser-Range Scanner System. In *CARS 2004; Computer Assisted Radiology and Surgery, 18th International Congress and Exhibition*, page 1328, Chicago, USA, 2004.

[58] K. Kreutz-Delgado, M. Long, and H. Seraji. Kinematic Analysis of 7-DoF Manipulators. *Int. J. Rob. Res.*, 11(5):469–481, 1992.

[59] J. Kuffner and S. LaValle. RRT-Connect: An Efficient Approach to Single-Query Path Planning. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA 2000)*, pages 995–1001, San Francisco, CA, 2000.

[60] Kuka. KUKA.Sim. [web page] http://www.kuka.com. [Accessed on January 3rd, 2007.].

[61] G. Lehmann, A. Chiu, D. Gobbi, Y. Starreveld, D. Boyd, M. Drangova, and T. M. Peters. Towards dynamic planning and guidance of minimally invasive robotic cardiac bypass surgical procedures. In *Proc MICCAI 2001. 4th International Conference*, pages 368–375, Utrecht, The Netherlands, 2001.

[62] P. Lemke and G. Hirzinger. Auslegungsoptimierung einer hochpoligen einzelpolbewickelten permanenterregten Synchronmaschine. In *Elektrisch-mechanische Antriebssysteme, Innovationen - Trends - Mechatronik*, pages 107–126, Fulda, Germany, 2004. VDI/VDE.

[63] K. Levenberg. A Method for the Solution of Certain Problems in Least Squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[64] H. C. Lin, K. Mills, P. Kazanzides, G. D. Hager, P. Marayong, A. M. Okamura, and R. Karam. Portability and Applicability of Virtual Fixtures Across Medical and Manufacturing Tasks. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 225– 230, Orlando, Florida, USA, 2006.

[65] M. Lourakis. Levmar: Levenberg-Marquardt Nonlinear Least Squares Algorithms in C/C++. [web page] http://www.ics.forth.gr/~lourakis/levmar/. [Accessed on January 31st, 2006.].

[66] R. Marmulla, S. Hassfeld, T. Lüth, and J. Mühling. Laser-Scan-Based Navigation in Cranio-Maxillofacial Surgery. *Journal for Craniomaxillofacial Surgery*, 31:267–277, 2003.

[67] D. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM J. Appl. Math.*, 11(2):431–441, 1963.

[68] MathWorks. Simulink – Simulation and Model-Based Design. [web page] `http://www.mathworks.com/products/simulink/`. [Accessed on April 22th, 2007.].

[69] M. Mayr. *Matching zweier Punktwolken zur intraoperativen Patientenregistrierung.* Munich University of Applied Sciences, Munich, Germany, 2004. Diploma Thesis.

[70] M. Menon and A.Tewari. Robotic Radical Prostatectomy and the Vattikuti Urology Institute Technique: An Interim Analysis of Results and Technical Points. *Urology*, 61:15–20, 2003.

[71] J-P. Merlet. Kinematics is not Dead. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation ICRA*, pages 1 – 6, San-Francisco, USA, 2000.

[72] J-P. Merlet. Still a Long Way to Go on the Road for Parallel Mechanisms. In *ASME 27th Biennial Mechanisms and Robotics Conf.*, Montréal, Canada, 2002.

[73] J.-P. Merlet. Optimal Design of Robots. In *Proceedings of Robotics: Science and Systems*, pages 311–318, Cambridge, USA, 2005.

[74] J. M. Müller. Chirurgenmanual. [web page] `http://www.charite.de`. [Accessed on June 15th, 2007.].

[75] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Boca Raton, USA, 1994.

[76] S. Y. Nof. *Handbook of Industrial Robotics.* John Wiley & Sons, New York, USA, 1999.

[77] T. Ortmaier. *Motion Compensation in Minimally Invasive Robotic Surgery.* PhD thesis, Munich, Germany, 2003.

[78] T. Ortmaier and G. Hirzinger. Cartesian Control Issues for Minimally Invasive Robot Surgery. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2000*, pages 565 – 571, Takamatsu, Japan, 2000.

[79] T. Ortmaier, D. Reintsema, U. Seibold, U. Hagn, and G. Hirzinger. The DLR Minimally Invasive Robotics Surgery Scenario. In G. Färber and J. Hoogen, editors, *Proceedings of the Workshop on Advances in Interactive Multimodal Telepresence Systems*, pages 135–147, Munich, Germany, 2001.

[80] T. Ortmaier, H. Weiß, U. Hagn, M. Grebenstein, M. Nickl, A. Albu-Schaeffer, C. Ott, S. Jörg, R. Konietschke, G. Hirzinger, and R. Wohlgemuth. A New Robot for Navigated Placement of Pedicle Screws. In *4. Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, Berlin, Germany, 2005.

[81] T. Ortmaier, H. Weiß, U. Hagn, M. Nickl, A. Albu-Schäffer, C. Ott, S. Jörg, R. Konietschke, L. Le-Tien, and G. Hirzinger. A Hands-On-Robot for Accurate Placement of Pedicle Screws. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4179 – 4186, Orlando, USA, 2006.

[82] D. J. Ostlie, K. A. Miller, R. K. Woods, and G. W. Holcomb. Single Cannula Technique and Robotic Telescopic Assistance in Infants and Children Who Require Laparoscopic Nissen Fundoplication. *Journal of Pediatric Surgery*, 38(1):111–115, 2003.

[83] F. B. Ouezdou and R. Mavroidis. Kinematic Synthesis of Manipulators Using a Distributed Optimization Method. *Journal of mechanical design, Transactions of the ASME*, 121(44), 1999.

[84] D. L. Pieper. *The Kinematics of Manipulators Under Computer Control*. PhD thesis, Stanford, USA, 1968.

[85] H. Reichenspurner, D. H. Boehm, A. Welz, C. J. Schulze, H. Gulbins, and S. M. Wildhirt. Three-Dimensional Video and Robot-Assisted Port-Access Mitral Valve Operation. *Annals of Thoracic Surgery*, 69(4):1176–1182, 1999.

[86] C. Riviere and P. Jensen. A Study of Instrument Motion in Vitreoretinal Microsurgery. In *Proceedings of the 22nd International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 59 – 60, 2000.

[87] B. Roth. Performance Evaluation of Manipulators from a Kinematics Viewpoint. *NBS Special Publication*, pages 39–61, 1975.

[88] J. M. Sackier and Y. Wang. Robotically Assisted Laparoscopic Surgery: From Concept to Development. 8(1):63–66, 1994.

[89] Scanlab. The Scancube. [web page] http://www.scanlab.de. [Accessed on April 11th, 2007.].

[90] K. Schittkowski. *Mathematische Grundlagen von Optimierungsverfahren*. Department of Mathematics, University of Bayreuth (Script, in German), 1999.

[91] K. Schittowski. NLQPL: A FORTRAN-Subroutine Solving Constrained Nonlinear Programming Problems. *Annals of Operations Research*, 5:485–500, 1985.

[92] W. Schroeder, K. M. Martin, and W. E. Lorensen. *The Visualization Toolkit (2nd ed.): An Object-Oriented Approach to 3D Graphics*. Prentice-Hall, Inc., Upper Saddle River, USA, 1998.

[93] R. Schwarz, A. Schmitgen, G. Grunst, T. Berlage, C. Schmitz, W. Schiller, and A. Welz. Gewebsverschiebung in roboter-assistierter endoskopischer Herzchirurgie: Auswirkungen auf Planung, Simulation und Führung. In *2. Jahrestagung der*

Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC), Nürnberg, Germany, 2003.

[94] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Advanced Textbooks in Control and Signal Processing Series. Springer-Verlag, London, New York, 1996.

[95] S. Selha, P. Dupont, R. Howe, and D. Torchiana. Dexterity Optimization by Port Placement in Robot-Assisted Minimally Invasive Surgery. In *Proceedings of SPIE – Volume 4570 Telemanipulator and Telepresence Technologies VIII*, pages 97–104, Newton, USA, 2002.

[96] M. Shimizu, W.-K. Yoon, and K. Kitagaki. A Practical Redundancy Resolution for 7 DoF Redundant Manipulators with Joint Limits. In *2007 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4510–4516, Rome, Italy, 2007.

[97] S. Singh and C. Riviere. Physiological Tremor Amplitude During Vitreoretinal Microsurgery. In *Proceedings of the IEEE 28th Annual Northeast Bioengineering Conference*, pages 171–172, Piscataway, USA, 2002.

[98] M. Suppa. *Simultaneous Robotic Exploration and Inspection of Flexible Workcells Using Multisensory Eye-in-Hand Systems*. PhD thesis, Hannover, Germany, 2007.

[99] M. Suppa, S. Kielhöfer, J. Langwald, F. Hacker, K. H. Strobl, and G. Hirzinger. The 3D-Modeller: A Multi-Purpose Vision Platform. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 781–787, Rome, Italy, 2007.

[100] H. A. Tabaie, J. A Reinbolt, W. P. Graper, T. F. Kelly, and M. A. Connor. Endoscopic Coronary Artery Bypass Graft (ECABG) Procedure with Robotic Assistance. *The Heart Surgery Forum*, 2(4):310–317, 1999.

[101] R. H. Taylor and D. Stoianovici. Medical Robotics in Computer-Integrated Surgery. *IEEE Transactions on Robotics and Automation*, 19(5):765–781, 2003.

[102] Tecnomatix. eM-Workplace. [web page] http://www.ugs.com/products/tecnomatix/assembly_planning/em_workplace_pc.shtml. [Accessed on June 15th, 2007.].

[103] J. G. Trafton, N. L. Cassimatis, M. D. Bugajska, D. P. Brock, F. E. Mintz, and A. C. Schultz. Enabling Effective Human-Robot Interaction Using Perspective-Taking in Robots. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 35(4):460–470, 2005.

[104] J. G. Trafton, A. C. Schultz, D. Perzanowski, M. D. Bugajska, W. Adams, N. L. Cassimatis, and D. P. Brock. Children and Robots Learning to Play Hide and Seek. *Human Robot Interaction*, pages 242–249, 2006.

[105] J. Traub, M. Feuerstein, M. Bauer, E. Schirmbeck, H. Najafi, R. Bauernschmitt, and G. Klinker. Augmented reality for port placement and navigation in robotically assisted minimally invasive cardiovascular surgery. In *Proceedings of Computer Assisted Radiology and Surgery (CARS 2004)*, pages 735–740, Chicago, USA, 2004.

[106] G. van den Bergen. *Collision Detection in Interaction 3D Environments*. Morgan Kaufmann, Breda, The Netherlands, 2003.

[107] G.-Q. Wei, K. Arbter, and G. Hirzinger. Real-Time Visual Servoing for Laparoscopic Surgery. *IEEE Engineering in Medicine and Biology*, 16(1):40 – 45, 1997.

[108] M. Wiering and F. Mignogna. Learning to Control Forest Fires with ESP. In *Proceedings of the Sixth European Workshop on Reinforcement Learning*, pages 22–23, Nancy, France, 2003.

[109] Wikipedia. Wikipedia, The Free Encyclopedia. [web page] http://en.wikipedia.org/, note=[Accessed on April 11th, 2007.].

[110] H. Wörn. The European Project AccuRobAs. [web page] http://www.accurobas.org/. [Accessed on January 3rd, 2007.].

[111] T. Yoshikawa. *Foundations of Robotics: Analysis and Control*. The MIT Press, London, England, 1990.

[112] M. Zinn, O. Khatib, B. Roth, and J.K. Salisbury. Playing it Safe [Human-Friendly Robots]. *IEEE Robotics and Automation Magazine*, 11(2):12 – 21, 2004.