TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Tuomo Kivelä

Increasing the Automation Level of Serial Robotic Manipulators with Optimal Design and Collision-free Path Control

Julkaisu 1509 • Publication 1509

Tampere 2017

Tampereen teknillinen yliopisto. Julkaisu 1509
Tampere University of Technology. Publication 1509

Tuomo Kivelä

Increasing the Automation Level of Serial Robotic Manipulators with Optimal Design and Collision-free Path Control

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Konetalo Building, Auditorium K1702, at Tampere University of Technology, on the 1st of December 2017, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2017

Doctoral candidate: Tuomo Kivelä
Laboratory of Automation and Hydraulic Engineering
Faculty of Engineering Sciences
Tampere University of Technology
Finland

Supervisor: Prof. Jouni Mattila
Laboratory of Automation and Hydraulic Engineering
Faculty of Engineering Sciences
Tampere University of Technology
Finland

Instructors: M.Sc. Sirpa Launis
Research Manager
Sandvik Mining and Construction Oy
Finland

M.Sc. Jussi Puura
Mining Technologies Specialist
Sandvik Mining and Construction Oy
Finland

Pre-examiners: Associate Prof. Shaoping Bai
Department of Mechanical and Manufacturing Engineering
Aalborg University
Denmark

Prof. Claudio Rossi
Centre for Automation and Robotics
Universidad Politecnica de Madrid
Spain

Opponents: Associate Prof. Shaoping Bai
Department of Mechanical and Manufacturing Engineering
Aalborg University
Denmark

Prof. Aki Mikkola
Laboratory of Machine Design
Lappeenranta University of Technology
Finland

ISBN 978-952-15-4043-1 (printed)
ISBN 978-952-15-4062-2 (PDF)
ISSN 1459-2045

Abstract

The current hydraulic robotic manipulator mechanisms for heavy-duty machines are a mature technology, and their kinematics has been developed with a focus on the human operator maneuvering a hydraulically controlled system without numerical control input. As the trend in heavy-duty manipulators is increased automation, computer control systems are increasingly being widely used, and the requirements for robotic manipulator kinematics are different. Computer control enables a different kind of robotic manipulator kinematics, which is not optimum for direct control by a human operator, because the joint motions related to the different trajectories are not native for the human mind. Numerically controlled robotic manipulators can accept kinematics that is more efficient at doing the job expected by the customer.

To increase the autonomous level of robotic manipulator, the optimal structure is not enough, but it is a part of the solution toward a fully autonomous manipulator. The control system of the manipulator is the main part of computer-controlled manipulators. A collision avoidance system plays an important role in the field of autonomous robotics. Without collision avoidance functionality, it is quite obvious that only very simple movements and tasks can be carried out automatically. With more complicated movement and manipulators, some kind of collision avoidance system is required. An unknown or changing environment increases the need for an intelligent collision avoidance system that can find a collision-free path in a dynamic environment.

This thesis deals with these fundamental challenges by optimizing the serial manipulator structure for the desired task and proposing a collision avoidance control system. The basic requirement in the design of such a robotic manipulator is to make sure that all the desired task points can be achieved without singularities. These properties are difficult to achieve with the general shape and type of robotic manipulators. In this research work, a task-based kinematic synthesis approach with the proper optimization method ensures that the desired requirements can be fulfilled.

To enable autonomous task execution for robotic manipulators, the control systems must have a collision avoidance system that can prevent different kinds of collisions. These collisions include self-collisions, collisions with other manipulators, collisions with obstacles, and collisions with the environment. Furthermore, there can be multiple simultaneous possible collisions that need to prevented, and the collision system must be able to handle all these collisions in real-time. In this research work, a real-time collision avoidance control approach is proposed to handle these issues. Overall, both topics, covered in this thesis, are believed to be key elements for increasing the automation of serial robotic manipulators.

Preface

This study was carried out from 2014 to 2017 at the Laboratory of Automation and Hydraulics (AUT) at Tampere University of Technology (TUT) in cooperation with Sandvik Mining and Construction Finland Oy.

I would like to express my deepest gratitude to my supervisor, Prof. Jouni Mattila, for his support, guidance, and advice during this thesis process. It has been my absolute privilege to prepare this thesis under his supervision. I would also like to thank the Head of AUT (formerly the Head of IHA), Prof. Kalevi Huhtala, for providing excellent facilities for the research.

I would like to express my sincere thanks to the personnel at the Sandvik Mining and Construction Finland Oy for setting up the experimental system and sharing valuable comments and suggestions. Special thanks go to Sirpa Launis and Jussi Puura for their valuable comments and productive discussions.

I wish to thank the preliminary examiners Prof. Shaoping Bai and Prof. Claudio Rossi for the evaluation of this thesis and their constructive feedback.

I am grateful also to all my colleagues at AUT (formerly IHA) who provided their existing knowledge and help during this work. I would like to thank, in particular, Dr. Janne Koivumäki, Dr. Janne Honkakorpi, and M.Sc. Liisa Aha, who helped me solve the challenges related to this dissertation. Special thanks also go to all members of the doctoral school, who have provided valuable discussions.

This work was funded by the Doctoral School of Industry Innovations (DSII) of Tampere University of Technology. This funding is greatly appreciated. Without it, this work could not have been done.

I want to thank my parents Marita and Kari and my siblings Raine, Marko, and Mervi for the support they have given me during these years.

The greatest thanks go to my beloved fiancée Jenni for her support and endless understanding during this work. She helped me cope at work with daily life. Finally, I am in debt to my sons Valtteri, Eemeli, Akseli, and Kasperi for their counterbalance to work.

Kangasala, November 2017.

Tuomo Kivelä

Contents

Acronyms

AABB	axis-aligned bounding box
APF	artificial potential field
BB	bounding box
CAD	computer-aided design
DH	Denavit-Hartenberg
DOF	degrees-of-freedom
GA	genetic algorithm
GPU	graphics processing unit
LM	Levenberg-Marquardt
OBB	oriented bounding box
PQP	proximity query package
RMMS	re-configurable modular manipulator system
RP	research problem
TCP	tool center point
TOW	total orientation workspace

List of Publications

P–I Tuomo Kivelä, Jouni Mattila, and Jussi Puura, "A Generic Method to Optimize a Redundant Serial Manipulator Structure", *Automation in Construction*, vol. 81, pp. 172–179, September 2017.

P–III Tuomo Kivelä, Jouni Mattila, Jussi Puura, and Sirpa Launis, (2018), "Redundant Robotic Manipulator Path Planning for Real-Time Obstacle and Self-Collision Avoidance", in *Advances in Service and Industrial Robotics. RAAD 2017. Mechanisms and Machine Science*, vol 49. Springer, Cham

P–IV Tuomo Kivelä, Jouni Mattila, Jussi Puura, and Sirpa Launis, "On-line Path Planning With Collision Avoidance for Coordinate Controlled Robotic Manipulators" in *Proceedings of the 2017 Bath/ASME Symposium on Fluid Power and Motion Control*, Sarasota, FL, USA, October 16-19, 2017.

P–V Tuomo Kivelä, Jouni Mattila, Jussi Puura, and Sirpa Launis, "Real-time Distance Query and Collision Avoidance for Point Clouds with a Heavy-duty Redundant Manipulator" in *Proceedings of the 2017 IEEE 8th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Ningbo, China, November 19-20, 2017.

Unpublished Manuscript

P– II Tuomo Kivelä, Jouni Mattila, Jussi Puura, and Sirpa Launis, "A Method for Task-based Optimization of a Mining Rig's Serial Manipulators with Arbitrary Topology", 2017.

1 Introduction

Robotics is a multidisciplinary field of study in which a robot means a complicated mechanical device that is usually controlled automatically to perform a task. The Robotics Institute of America has given the following definition for robots: *"A robot is a reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of task."* In general, robotic manipulators can be divided into two types: parallel manipulators and serial manipulators. A parallel manipulator is a mechanical system that uses several actuators to support a single end-effector. A Steward platform is one of the best-known parallel manipulators which is formed from six linear actuators. A serial manipulator, in contrast, is a mechanical system that is designed as a series of links connected by joints that extend from a base to an end-effector. Typical industrial manipulators are serial manipulators.

Robots vary in appearance from small-scale miniature robots to large-scale heavy-duty robots. For example, arm-like robots are a common type of robot, and they are also known as robotic manipulators. A robotic manipulator consists of links connected by joints with one fixed end and one end to perform a given task [1]. The joints of the robotic manipulator are components, which enables relative motion between adjoining links.

Hydraulic robotic manipulators are a special group of robotic manipulators where the joints of the manipulator are moved with hydraulic actuators (Figure 1.1). The overall complex hydraulic circuit consists of numerous components, including hydraulic valves, hydraulic actuators, filters, and hydraulic pumps that power the fluid circuitry through the prime mover.

The history of robots begins far from the past. Robots as we know them began to be developed during the Industrial Revolution. The first modern robots were industrial robots in factories. These robots were relatively simple machines capable of manufacturing and assembly tasks. They were programmed to repeat the task without the need for human assistance. The development of digital controllers enabled industrial robots to make use of artificial intelligence. Today, modern robots, for example, service robots and humanoid robots, are much more complicated than conventional six degrees-of-freedom (DOF) industrial robots. Therefore, the design concepts and control algorithms developed for conventional industrial robots no longer apply, and these methods must be updated and extended to meet the requirements of modern robots. In high-tech fields, modern robots have been already adapted with intelligent and complex control and decision algorithms. However, there are many fields of technology where the industry relies on conventional robotics, especially heavy-duty industry. These industrial fields have expressed interest in modernizing their products and production by increasing automation. This requires design concepts and control algorithms that can be implemented in the current products

Figure 1.1: Heavy-duty hydraulic manipulators on a drilling rig (Photo: Sandvik) [2].

and production without a huge investment.

The commercial success of industries that use robotic manipulators is due partly to the high automation level and high performance. Usually, these manipulators co-operate with other manipulators for assembly or positioning purposes. These features require an appropriate structure of the robotic manipulators, as well as an intelligent control algorithm, so that multiple robotic manipulators can work efficiently in the same workspace.

1.1 Research Problem

A recent survey confirmed that there is high industrial interest in digital and automated solutions [3]. Although this survey focused on the mining industry, the same trend can be seen in other heavy industries as well. It is predicted that this trend will continue as companies invest in new technology to improve their operations and competence. This interest creates a great opportunity for academics to co-operate with industry to solve problems in which they are investing. The main source of motivation for conducting research in this area is that research on industrially relevant methods for designing and intelligently controlling robotic manipulators are lacking in the literature. We examine two underdeveloped features for robotic manipulators: task-based kinematic synthesis of robotic manipulators and a collision avoidance control system for robotic manipulators. Even though these two topics are not necessarily directly coupled to each other, there is a strong coherence between them within autonomous robotics. For example, it is obvious that a collision-free control requires such a manipulator structure that collisions can be avoided while achieving the goal pose. Furthermore, if the manipulator structure

is optimized for an autonomous task execution, it most likely also requires a collision avoidance system to fulfill the task.

Therefore, the research problem (RP) addressed in this thesis involves the design of kinematic synthesis solutions for serial robotic manipulators with the following features:

RP1: **Optimal structure:** The kinematic synthesis solution should be able to optimize the kinematic structure of a device from the task description. This is referred to as a topology synthesis, and it is used to optimize the number, type, and arrangement of joints and the links connected by these joints.

RP2: **Optimal dimensions:** The kinematic synthesis solution should be able to optimize the dimensions of a device. This is referred to as a dimensional synthesis.

RP3: **Task-based design:** The kinematic synthesis solution should be able to fulfill the desired task space definitions for each task point.

RP4: **General applicability:** The kinematic synthesis solution should be applicable to multi-DOF robotic manipulators (joint space) and 6-DOF Cartesian space.

Furthermore, the RP addressed in this thesis also proposes a collision avoidance control approach for serial robotic manipulators with the following features:

RP5: **Distance query:** The collision avoidance system should detect possible collisions between the manipulator and obstacles or the manipulator's self-collisions in order to prevent these collisions. Possible collisions should be detected, and the shortest distance between these objects should be calculated so that there is enough time to react to these warnings.

RP6: **High performance:** The collision avoidance system should provide high performance while satisfying the accuracy of the manipulator. The performance of the collision avoidance system should be experimentally validated to meet the real-time requirement.

RP7: **General applicability:** The collision avoidance system should be applicable to kinematically different robotic manipulators. The solution should be suitable for use by multi-DOF manipulators that contain prismatic and revolute joints.

1.2 Justification for the Kinematic Synthesis of Serial Robotic Manipulators

The current robotic manipulator mechanisms for heavy-duty machines are a mature technology, and their kinematics has been developed with a focus on the human operator maneuvering a hydraulically controlled system without numerical control input. As the trend in heavy-duty machines is increased automation, computer control systems are increasingly being widely used, and the requirements for robotic manipulator kinematics

are different. Numeric control enables a different kind of robotic manipulator kinematics, which is not optimum for direct control by a human operator, because the joint motions related to the different trajectories are not native for the human mind. Numerically controlled robotic manipulators can accept kinematics that is more efficient in doing the job expected by the user.

Comprehensive studies have been performed with lightweight, electrically driven manipulators. These studies lack about information about the dynamic behavior of heavy-duty long-reach hydraulic robotic manipulators. For example, drilling manipulators have to be able to servo-control the drill bit position and orientation of heavy drilling machines at a more than 10 meter reach and to be able to support highly dynamic drill bit forces during percussive drilling. Due to high force requirements, these robotic manipulators are driven by hydraulic linear and rotary actuators. For robotic manipulator design and optimization purposes, transformations from joint space coordinates to actuator space coordinates are needed [4, 5]. The first target is to develop a serial robotic manipulator design method that is able to generate a set of robotic link structures defined by Denavit-Hartenberg (DH) parameters that fulfill use case requirements. The first design requirement is to ensure that the robotic manipulator has the minimum required reachability that includes some degree of additional manipulability. This means the ease of arbitrarily changing the position and orientation of the end-effector at the tool center point (TCP) at the desired target position and orientation of the drill bit. In addition, many practical tasks, like rock drilling, require position control not only of the robotic manipulator end-effector but also of the force exerted by the end-effector on the environment [6, 7].

With reference to electrically driven, lightweight robots with a small power-to-size ratio, much research has been done to quantify various performance criteria for robot design and advanced motion control. Performance criteria to be optimized include load capacity, force transmission from the joints to the end-effector, dynamic responsiveness, etc. Based on these performance measures, optimizations, such as mass distribution, actuator size, or link dimensions, can be improved. Zhou and Bai [8] developed a design method for lightweight robotic manipulator. The proposed method combines the kinematics, dynamics, and structural strength analysis. The main objective of this method is to minimize the weight of the manipulator. Graettinger and Krogh [9] proposed the acceleration radius as a global performance measure for robotic manipulators. The acceleration radius is a uniform lower bound on the magnitude of the acceleration that can be achieved at the end-effector from any state in the operating region. Yoshikawa [10] defined manipulability and dynamic manipulability for the evaluation of kinematics and dynamic robot performance. Local dynamic indices have also been investigated by [11] for sizing actuators. Klein [12] illustrated the meaning of some of these indices, which he calls a dexterity measure, in the context of redundant manipulability. Another important criterion in optimal manipulator design is that it can achieve isotropic configurations [13]. At these configurations, the best servo accuracy is achievable; the likelihood of error is equal in all directions, and equal forces may be exerted in all directions. Thus, these points may be considered the optimal working points of a given manipulator design.

1.3 Justification for Collision Avoidance of Serial Robotic Manipulators

Drilling or bolting manipulators of mining rigs are typical mechatronic systems that compromise a redundant mechanical structure (i.e., mechanical links in series or in parallel),

drive-train components (hydraulic linear or rotary actuators) and control software for motion planning and control. An additional challenge is that mining rigs are equipped with several multi-DOF robotic manipulators that carry heavy rock drilling and bolting machines at a more than 10 meter reach. In this multi-robotic manipulator environment, path-planning and collision avoidance are important features. The objective of this multi-robot system is to perform given task collaboratively and cooperatively. Ideally, subtasks assigned to a failed or slowly performing manipulator should be automatically transferred to other available manipulators without collisions. In this research, collision-free path planning in a multi-robot environment is treated as an optimization problem [14].

Wear and tear on robotic manipulators due to collisions, for example, with the side walls in underground mines, are very common during end-effector positioning, even at low speeds. This results in increased maintenance costs, and thus, collisions are considered a major disadvantage of end-effector positioning operation. Therefore, the aim of a real-time collision avoidance control system is not only to find a collision-free path to the goal pose in a dynamic environment but also to reduce the downtime and maintenance costs of the devices. In addition, the collision avoidance system enables several features that increase the performance of the robotic manipulator. The following features can be included into a collision avoidance system:

1. Avoid collisions
2. Find a collision-free path in real-time
3. Decrease machine downtime
4. Decrease maintenance costs
5. Automatic sequence
6. Automated operations

1.4 Research Methods and Restrictions

Kinematic synthesis is a broad field with various applicable approaches. This research work concentrates on task-based kinematic synthesis. A real-life, mining customer, drilling pattern of holes is used as a task, that a robotic manipulator should be able to fulfill. The dimensions of the existing manipulator are optimized to enhance the performance and profitability of the manipulator. Furthermore, we use an approach in which several different topologies are optimized in terms of link lengths. Thereafter, suitable performance measures are used to rank the optimized structures and select the most attractive one. The kinematic synthesis process is carried out by using a real mining customer case.

Task-based kinematic synthesis methods are generally chosen because they can guarantee optimal task execution. The kinematic synthesis method based on a Levenberg-Marquardt (LM) was chosen in publications **P−I** and **P−II** to optimize the serial manipulator structure for the given task. In publication **P−I**, an existing manipulator structure was used to describe the topology of the manipulator to be optimized because this topology has been proven by construction customers to be suitable for the desired task. However, this structure, composed of eight joints, was designed for a human operator controlling the manipulator manually. In publication **P−II**, the same task space was used to find a

simpler structure for the manipulator that is controlled by a computer. This research work led to the design of a new manipulator structure for autonomous control.

Collision avoidance addresses collision-free path planning for general serial robotic manipulators. Generality means that the number and type of joints are not restricted, and the work space can be n-dimensional. The collision avoidance method proposed in publications **P–III**, **P–IV**, and **P–V** is a real-time collision-free path planning approach that enforces the actuator's position and velocity limits. Solving the collision-free path from a starting pose to a goal pose with respect to moving and deformable obstacles is excluded from this work to limit the scope. Collision-free path generation is utilized using the null space motion concept [15]. Accurate collision point detection and distance query are the key elements of successful collision avoidance systems. In publications **P–III** through **P–V**, a novel approach to calculate the shortest distance between two objects with a complex structure at the real-time rate is shortly presented. The proposed method is based on point cloud data that is used to describe the objects. This method is used in publications **P–III**, **P–IV**, and **P–V** to calculate the direction of the repulsive forces that are used to prevent collisions. The proposed collision avoidance control approach is validated through simulations and experimental work.

1.5 Thesis Contributions

As this thesis is a compendium thesis, the main contributions of the publications and the unpublished manuscript can be summarized as follows:

P–I: A generic method for optimizing the structure of a redundant serial manipulator was presented. The proposed optimization method takes into account not only the desired task points but also practical design constraints for given task points, including the most important constraints, such as the available confined working envelope and joint ranges. The topology of the tunnel construction manipulator used in this case study was proven by construction customers to be suitable for tunnel construction. Therefore, a change in the manipulator's topology was not necessary in this study. Instead, the dimensions of the existing manipulator were optimized to enhance the performance and profitability of the manipulator [16].

P–II: A generic method for selecting and optimizing a redundant serial manipulator structure was presented. In **P–I**, an existing topology was optimized to find the optimal dimensions for that topology. In this paper, a set of pre-chosen topologies are optimized to find the optimal manipulator topology and the optimal manipulator dimensions for the given problem. In the generic approach proposed in this paper, due to practical reasons such as the resulting robot design cost-effectiveness and to maintain the readability, the set of robot topologies is limited to serial-type manipulators with a maximum number of seven actuators. Real customer application, used in this study, requires a long-reach manipulator (10-15 meters) due to the desired workspace. Therefore, manipulator topologies that have at least one prismatic joint are interesting; however, it is not clear if six joints are better than seven joints. The proposed method optimizes the set of robot topologies and ranks them according to the selected performance measure.

Our results indicate that this kind of method is effective in finding the optimal structure and dimensions for the serial-type manipulator from the task specifications based on the selected performance measure.

P−III: A collision avoidance system for preventing a manipulator's self-collisions was introduced. The proposed collision avoidance control system enables autonomous operation for a redundant serial manipulator. To verify this functionality, an experimental test was carried out with a heavy-duty hydraulic manipulator, which is used in underground tunneling. The experiment validated that the proposed method can be used in real-time control [17].

P−IV: A collision avoidance system for preventing a manipulator to collide during the task execution was introduced. In **P−III**, only manipulator's self-collisions were considered. In this paper, a more complex system to avoid collisions is studied. This system comprises two manipulators which both can avoid self-collisions, collisions with obstacles, and collisions with each other. A novel architecture for detecting the exact collision points of two geometrically complex objects and how to use this information in a collision avoidance system was also proposed. The proposed collision avoidance method uses the calculated shortest distance between two point clouds to move the manipulator away from the obstacles and toward the desired goal. An experiment and simulation validated that the proposed method can be used in real-time control [18].

P−V: In this paper, the proposed collision avoidance control architecture (**P−III** and **P−IV**) was applied to completely different manipulator. Furthermore, the use of a laser scanning system to provide realistic environment and obstacles information to the collision avoidance system was studied. In short, the coordinated motion control system of the heavy-duty hydraulic manipulator resolves joint references so that the goal position can be reached in real-time without any collisions. The proposed method is able to detect and prevent different types of possible collisions, including self-collisions and collisions with obstacles. The collision server is used to retain static point clouds and to calculate the shortest distance between objects that are too close to each other. The point clouds on the server are kept up-to-date with the manipulators' joint sensors and laser scanner-based measurements. During coordinated motion control, the joint trajectories of the redundant manipulator are modified so that collisions can be avoided, while at the same time, the trajectory of the end-effector maintains its initial trajectory if possible. Results are given for a 4-DOF redundant heavy-duty hydraulic manipulator to demonstrate the capability of this collision avoidance control system [19].

1.6 The Author's Contribution to the Publications

This section clarifies the author's contribution to the publications and the unpublished manuscript.

P−I: The author wrote the paper and developed the approach for optimizing the existing serial robotic manipulator structure using the LM algorithm.

Jussi Puura helped with the system requirement definitions and reviewed the paper. Professor Jouni Mattila, the academic supervisor, reviewed the paper and suggested major improvements.

P–II: The author wrote the paper and developed the task-based kinematic synthesis method. Jussi Puura helped with the system requirement definitions and reviewed the paper and suggested improvements. Professor Jouni Mattila and Sirpa Launis, the industrial supervisor, reviewed the paper and suggested major improvements.

P–III: The author wrote the paper and developed and performed a wide range of experiments to validate the proposed collision avoidance system. Jussi Puura and Sirpa Launis helped with the system requirement definitions and the experimental setup, reviewed the paper, and suggested improvements. Professor Jouni Mattila reviewed the paper and suggested major improvements.

P–IV: The author wrote the paper and expanded the proposed collision avoidance system to prevent collisions with different types of collisions. Jussi Puura and Sirpa Launis helped with the system requirement definitions, reviewed the paper, and suggested improvements. Professor Jouni Mattila reviewed the paper and suggested major improvements.

P–V: The author wrote the paper and expanded the proposed collision avoidance system to prevent collisions with external obstacles that were detected and identified by using a laser scanner. Pauli Mustalahti helped with the experimental measurement system. Professor Jouni Mattila reviewed the paper and suggested minor improvements.

1.7 Outline of the Thesis

The introductory part of this compendium thesis is divided into six chapters. The contents of the remaining chapters are summarized below.

Chapter 2 is a review of the state-of-the-art in kinematic synthesis of serial robotic manipulators. The optimization principles of task-based kinematic synthesis are covered, and the typical serial robotic manipulator design scenario is briefly discussed.

Chapter 3 is a review of the state-of-the-art in collision avoidance and collision detection of serial robotic manipulators. Furthermore, general methods for solving collision avoidance control problems are introduced and discussed.

Chapter 4 summarizes the simulation and experimental results and discoveries shown in the individual publications and manuscript, comprising this compendium research work.

Chapter 5 discusses how the RPs, described in Section 1.1, are addressed.

Chapter 6 provides the conclusions and suggestions for future work.

2 State-of-the-Art Kinematic Synthesis of Serial Robotic Manipulators

In the introduction, the justification for the kinematic synthesis for serial robotic manipulators was described. The task of the kinematic synthesis is to develop a mechanism that meets the requirements of the given desired motion. The kinematic synthesis, for example, enables the design of such a serial robotic manipulator that is guaranteed to reach the desired task points with the desired end-effector orientations. Kinematic synthesis solutions for serial robotic manipulators have been developed in the literature. These solutions use optimization methods to find an optimal kinematic structure for the manipulator. Moreover, these solutions are increasingly task-based because the classical approach to optimize a general-purpose manipulator does not guarantee optimal task execution [20]. The constant rise in computation power in recent years has helped solve more complex and larger task-based kinematic synthesis problems.

This section covers state-of-the-art solutions in the kinematic synthesis of serial robotic manipulators, with a focus on task-based optimization. We first classify kinematic synthesis terms in Section 2.1 and follow with an introduction to the principles and approaches of the kinematic synthesis in Section 2.2. Finally, in Section 2.3, state-of-the-art kinematic synthesis solutions applicable to serial robotic manipulators are reviewed, to reveal the strengths and weaknesses of the solutions.

2.1 Kinematic Synthesis

The science of mechanism kinematics can be roughly divided into two separate parts: analysis and synthesis. Analysis is used to resolve or predict a defined mechanism's reaction to specific motions of the driver. In other words, analysis tries to determine the motion of the mechanism's bodies if one or more of the actuators of the mechanism is actuated. Kinematic synthesis, instead, is used to create a mechanism to attain a specific motion. In this sense, synthesis can be thought of as the inverse problem of analysis [21]. This leads to a fundamental question to which researchers over the years have tried to find an answer: What is the best mechanism configuration for the desired task? Obviously, there is no simple or unique answer to this question. Instead, in most cases there can be an infinite number of answers this question. In the literature, the kinematic synthesis problem is usually divided into two or three steps to make handling of the problem easier. These steps are as follows:

1. Type or topology synthesis

2. Number synthesis
3. Dimensional synthesis

Type synthesis and topology synthesis are synonyms, and both are used in the literature. In some references, the type or topology synthesis is combined with the number synthesis. The topology synthesis involves determining the structure of a device, for example, serial or parallel structure. This phase is also used to select the type of links and joints. The second phase, the number synthesis, determines the number of links of the mechanism. The last phase, the dimensional synthesis, is devoted to determining the dimensions of a device.

The selection of the type of mechanism needed to accomplish a desired task depends not only on the kinematic requirements but also on many other factors, such as, materials, manufacturing costs, etc. Because of many different factors, it is impossible to automatize the topology synthesis without losing generality [21, 22]. Therefore, an exhaustive search approach is widely used among researchers to systematically find the best mechanism for the application at hand. This approach also involves the number synthesis. This phase is followed by the dimensional synthesis for sizing of each feasible alternative using an optimization method.

We focus exclusively on serial robotic manipulators synthesis, which simplifies the topology synthesis. This simplification allows us to systematically test all possible topologies in the range of the minimum and maximum number of joints. In this thesis, the topology is assumed to be known, or the systematic approach is used. Therefore, the topology synthesis and the number synthesis do not deserve much attention. Instead, the third phase, the dimensional synthesis, presents challenging problems. The remainder of this chapter is addressed to the dimensional synthesis problems.

2.2 Kinematic Synthesis Methods

There are multiple ways to categorize kinematic synthesis methods, for example, by the optimization method or based on a general-purpose design or a task-specific design. Patel and Sobh [20] divided kinematic synthesis methods into three categories:

1. Geometric approach
2. Parametric optimization approach
3. Task-based design approach

2.2.1 Geometric Approach

In 1833, Grashof proposed a simple condition to understand the mobility of a four-link mechanism. This finding is commonly known as Grashof's criterion. This criterion is used to estimate the rotatability of the links in a four-bar mechanism. The criterion has the following well-known definition: *"If the sum of the shortest and longest link of a planar quadrilateral linkage is less than or equal to the sum of the remaining two links, then the shortest link can rotate fully with respect to a neighboring link."* A mathematical formula for the previous definition is as follows:

$$s + l \leq p + q, \tag{2.1}$$

where s and l are the shortest and longest links, respectively. The other two parts are designated p and q (see Figure 2.1).

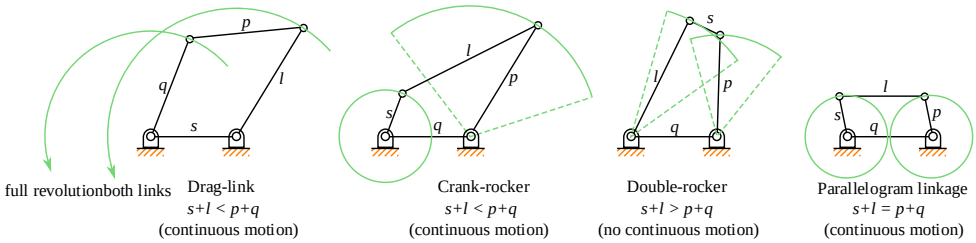


Figure 2.1: Types of four-bar linkage, where s and l stand for the shortest and longest links, respectively. The other two parts are described with p and q [23].

Many researchers have further extended Grashof's criterion and used it for determining the optimal link lengths. Even though Grashof's criterion can be applied to closed loop kinematic chains, it can be used for serial robotic manipulator design by assuming the distance between the base of the manipulator and the task point is one link in the closed mechanical chain. With this approach, it is possible to design a manipulator with high dexterity at the given task points. Here, dexterity refers to the ability of the manipulator to achieve as large as possible orientation about a given task point. However, if the task definition defines only one required orientation, then an alternative approach should be considered. This method cannot be used to optimize a manipulator at multiple task points. Therefore, task satisfaction cannot be guaranteed. In addition, this method cannot be applied to a design problem where the robotic manipulator requires prismatic joints.

2.2.2 Parametric Optimization Approach

Parametric optimization is a classic approach to solving an optimization problem. The main idea is to create an objective function and constraint functions and then sum them to form a cost function, which can be minimized or maximized to find the optimal solution. Constraint functions can be weighted to address the solution in the specified direction. Different performance criteria that quantify the performance of the robotic manipulator, such as the structural length index, manipulability measure, condition number, and global conditioning index, can be used to design the optimal structure, according to the criterion. A comprehensive survey of manipulator performance parameters and their limitations was carried out by Patel and Sobh [24].

Although the parametric optimization approach has dominated the kinematic synthesis of design process for robotic manipulators, it has some drawbacks. One such drawback is that this approach is not part of the design process of the manipulator's structure; this approach improves only an existing manipulator's structure and does not generate new structures. Another significant drawback of the parametric optimization approach is the manipulator's Jacobian matrix non-homogeneity. The Jacobian matrix is non-homogeneous due to the different unit used to represent the link lengths and the joint values [24]. Therefore, the majority of the proposed parametric optimization approaches focus on robotic manipulators without prismatic joints or in cases where end-effector orientation is not important. However, in spite of many scaling techniques that have been

proposed to treat the non-homogeneity of the Jacobian matrix [25, 26], they have not received very wide endorsement.

2.2.3 Task-based Design Approach

The task-based optimization or synthesis of a robotic manipulator consists of finding a set of manipulator design parameters so that the required task points and kinematic requirements are fulfilled. The kinematic requirements, such as a constrained workspace, design parameter limits, and joint limits, are task specifications that affect the kinematic structure of the robotic manipulator.

Usually, most common industrial 6-DOF robotic manipulators with a spherical wrist are designed for the desired rated payload while maximizing the manipulator workspace envelope. In more advanced design scenarios, the manipulator total orientation workspace can be specified as a set of range of rotation angles of the end-effector in a position inside the bounded workspace [27].

Clearly for a restricted workspace this is not an adequate design approach. Instead, in more specific robotic manipulator design scenarios, the reachable workspace should be divided into several bounded subspaces that each have different requirements for the required range of wrist orientation angles. For example, in a tunnel drilling task, the drilling pattern near the center of the tunnel is more dense and requires a wide range of orientation angles to be reached. However, near the boundaries of the tunnel (walls, ceiling, and ground), the drilling pattern execution requires a lower range of orientation angles to be reached. Therefore, this design problem consists of well-defined robotic manipulator design requirements with a large number of defined task positions to be reached in the workspace. Due to the complexity of the given design optimization problem, it is not obvious which robotic topology should be chosen for this given robotic manipulator design problem.

In general, the task for the task-based design approach is given in terms of the task points that the manipulator has to reach with a specified end-effector rotation. Let $\mathbf{P}$ be the set of task points; then the task can be described as follows:

$$\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots \mathbf{p}_m\} \in \mathbf{TS}, \tag{2.2}$$

where m is the number of task points, and $\mathbf{TS}$ is the task space that defines the position and the orientation of the manipulator's end-effector. Typically, a manipulator's joint is a revolute or prismatic joint with one DOF. By using commonly known DH parameter notation, each link of the manipulator can be described with four parameters, where three parameters are design parameters and one is a joint variable, depending on the joint type. In the case of a revolute joint, the design parameters are $\{a, \alpha, d\}$, and in the case of a prismatic joint, the design parameters are $\{a, \alpha, \theta\}$. Now, the serial robotic manipulator configuration can be given as:

$$\mathbf{DH} = \{a_1, \alpha_1, d_1 \ or \ \theta_1, \dots, a_n, \alpha_n, d_n \ or \ \theta_n\} \in \mathbf{CS}, \tag{2.3}$$

where n is the DOF of the manipulator, $\mathbf{DH}$ is the serial manipulator configuration set, and $\mathbf{CS}$ is the configuration space. The joint vector for the n-DOF manipulator is defined as follows:

$$\mathbf{q} = [\mathbf{q}_1, \mathbf{q}_2, \dots \mathbf{q}_n] \in \mathbf{QS}, \tag{2.4}$$

where **QS** is the joint space. Mapping between the manipulator's configuration set (**DH**), with a joint vector (**q**), and a specific task point can be described as follows:

$$f(\mathbf{DH}, \mathbf{q}) = \mathbf{p}. \tag{2.5}$$

Now, the optimization problem is stated as follows:

$$\exists \mathbf{DH} : \ \forall \mathbf{p} \in \mathbf{TS} \ ; \exists \mathbf{q} \in \mathbf{QS} | f(\mathbf{DH}, \mathbf{q}) = \mathbf{p}. \tag{2.6}$$

2.3 State-of-the-Art in Kinematic Synthesis

Current heavy-duty robotic manipulator mechanisms are a mature technology, and their kinematics has been developed with a focus on the human operator maneuvering a manually controlled system without numerical control input. As the trend in heavy-duty robotic manipulators is increased automation, computer control systems are increasingly being widely used, and the requirements for robotic manipulator kinematics are different. Numeric control enables a non-intuitive kind of robotic manipulator kinematics, which is not optimum for direct control by a human operator, because the joint motions related to the different trajectories are not native for the human mind. Numerically controlled robotic manipulators can accept kinematics that is more efficient in executing the desired task.

In the field of kinematic synthesis, different synthesis approaches (Section 2.2) have been used to optimize robotic manipulator structures. Grashof's idea, extended by Paul [28], has been widely used to design 3-DOF manipulators. Vijaykumar et al. [29] subdivided the structure of the manipulator into a regional structure and an orientation structure, which enables use of the Grashof criterion with 6-DOF manipulators. The dexterity criterion was applied to design a manipulator by maximizing the dexterity. Dexterity refers to the ability of the manipulator to realize any orientation about the given point. This kind of design methodology is applicable to designing a general-purpose manipulator, where the objective is to maximize the reachable workspace, as well as maximize the dexterous workspace.

Li and Dai [30] claimed that the orientation angle workspace indicates the flexible degree of the robotic manipulator and developed a method for calculating the orientation angle workspaces and output variation curves of the orientation angle workspace within the position workspace by using Grashof's criterion. Patel and Sobh [31] also used Grashof's criterion to design an optimal serial three-link manipulator. They proposed an optimization algorithm which optimizes the manipulator in terms of the dexterity index for the desired region of interest. This region can either be a set of task points or a trajectory of the end-effector of the manipulator. The optimal design was achieved by generating the optimal link lengths for a three-link planar manipulator.

Most of the studies that use Grashof's criterion are devoted to optimizing mechanisms that consist of three links or mechanisms that can be divided into groups of three links. However, Grashof's criterion can be extended to cover mechanisms that have more than three links [32]. Moreover, the majority of the reported works are for crank-driven linkages. Recently, Bai [33] introduced a motion analysis method for the coupler link of planar four-bar linkages. This method can be applied to kinematic synthesis, for example, the conditions of full rotations of the coupler or reaching to a specified range.

Manipulator performance parameters have been widely used to optimize the robotic manipulator structure, and Patel and Sobh [24] made a comprehensive survey of these

parameters and their limitations. This survey did not provide recommendations about which performance measure should be used to design a robotic manipulator structure. Instead, the survey provided a review of the definition, classification, scope, and limitations of some of the widely used performance measures, whose significance and limitations have not always been well discussed. According to the survey, performance indices can be classified based on the following three groups:

1. Scope (local or global indices)
2. Characteristic (kinematic, dynamic, or neither)
3. Application (intrinsic or extrinsic)

Local performance indices are performance metrics that are dependent on the pose of the manipulator, whereas global indices are pose-independent indices. The condition number is one of the most widely used local performance indices to obtain the optimal manipulator structure [34, 35]. The condition number is a local kinematic conditioning index used to measure the degree of ill-conditioning of the manipulator or the kinematic isotropy of the Jacobian. The condition number is defined as the ratio of the maximum and minimum singular values of the Jacobian.

Another widely used performance index is the manipulability. This kinematic performance index was proposed by Yoshikawa [10]. Later, Sobh and Toundykov [36] used this performance index to optimize three-link robotic manipulators with a given set of task points. This design process utilized a numerical optimization based on the steepest-descent algorithm together with an objective function which was based on the manipulability index. The manipulability index itself is based on the manipulator's Jacobian matrix, and calculation of the manipulability index depends on whether the manipulator is redundant or not. If the manipulator is redundant, then the manipulability index is defined as the square root of the determinant of the product of the Jacobian matrix and its transpose. In the case of a non-redundant manipulator, the manipulability is simply the absolute value of the determinant of the Jacobian.

With the parametric optimization approach, it possible to use more than just one performance index. Kucuk and Bingul [37] constructed an objective function that had two objectives: the maximum workspace volume and maximization of the product of the manipulability measure and the condition number of the Jacobian matrix. Sixteen different combinations of three-link manipulators were optimized using the proposed performance indices. The results showed that a high local performance index does not always guarantee a high global performance index. The idea of using multiple performance indices was further developed by Kucuk and Bingul [38], who used conventional global and local indices to optimize the robotic manipulator structure. These indices were the structural length index, manipulability measure, condition number, and global conditioning index. The workspaces of sixteen fundamental robotic manipulators were optimized and compared to each other.

The variation in tasks that robotic manipulators should be able to accomplish is huge. A dominant design approach for robotic manipulators is based on the parametric optimization approach because it can be used to optimize a general-purpose manipulator structure applicable for different tasks. However, this approach cannot guarantee task satisfaction [20]. The task-based design approach is the opposite of the other design approaches and

does not try to maximize the manipulator's workspace. Instead, this method can guarantee that the task can be completed. Paredis and Khosla [39] used a task-based approach to find the optimal structure for a six-DOF spherical wrist manipulator. The proposed method was able to optimize the DH parameters of the manipulator by minimizing the objective function. However, they have to make several assumptions to make the problem solvable, including only rotary manipulators are considered, spherical wrist, no self-collision, etc.

Kim and Khosla [40] studied task-based robotic manipulator optimization with a reconfigurable modular manipulator system (RMMS). They stated that the optimal design of a robotic manipulator even for a simple task is difficult and may lead to a combinatorial explosion. Therefore, they proposed a high-level design framework that decomposes the complexity of the design problem. They called this framework Progressive Design. Progressive Design is composed of three steps: kinematic design, planning, and kinematic control. The first step is to find the optimal values for the DOF, type, dimension, and base position. The second step optimizes the positions between two adjacent task points of the kinematic design. In the final step, task points are connected along the given trajectory. The proposed method was applied to develop an optimal manipulator for changing tiles on the space shuttle [40, 41]. This framework was developed considering its extension and can be extended to more complex problems with small modifications.

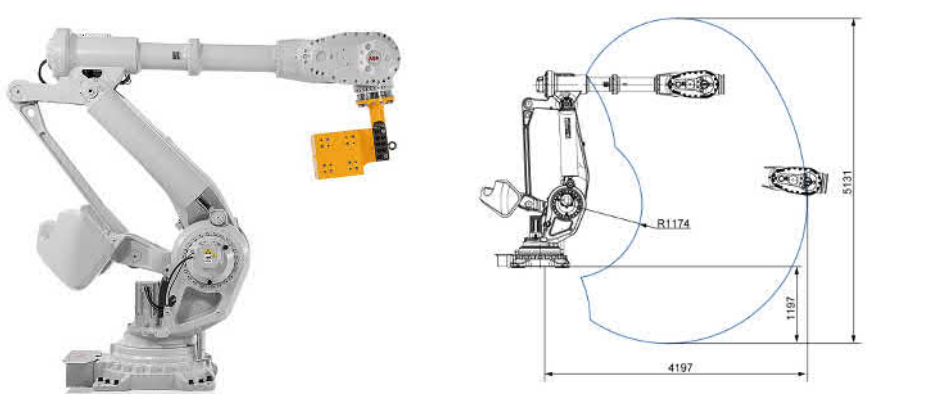
Most optimization approaches for robotic manipulators in the literature handle non-redundant manipulators. However, in some applications, especially if the workspace is cluttered, redundant manipulators have better possibilities of completing the desired task without collisions. Therefore, task-based design approaches for redundant robotic manipulators are needed. Singla et al. [42] proposed an optimization method for redundant rotary robotic manipulators in a cluttered environment. They formulated a constrained optimization problem minimizing the positional error and simultaneously avoiding any collision of the manipulator with either the obstacles or within its links. The augmented Lagrangian method was used to solve the optimization problem. However, the problem formulation was developed independently from the optimization algorithm. This enables use of different optimization algorithms. This approach, similar to many other approaches, avoids dealing with prismatic joints.

Patel and Sobh [20] proposed a method for developing the optimal manipulator structure of a non-redundant 6-DOF serial robotic manipulator from task descriptions with a spherical wrist. The proposed method focuses on the regional structure and finds all possible manipulator configurations that can reach all the task points with the required orientations. Then among these configurations, the configuration which can attain maximum end-effector velocities in arbitrary directions with the joint constraints was considered the optimal structure, and it provides the best kinematic performance. They also optimized the manipulator's structure based on the least power-consuming configuration. This method is capable of handling prismatic joint in any of the three first joints.

Gradient-based methods for solving manipulator optimization problems have been popular among researchers. However, these methods suffer a local minimum problem, and they are not guaranteed to find a global minimum. This is due to the manipulator configuration search space which is excessively large. Therefore, alternative methods have been proposed for use as manipulator optimization algorithms. The two approaches most frequently used for solving the optimization problems are random line search and genetic algorithm (GA). Use of the GA was proposed by several researchers [43 – 45]. Shiakolas et al. [46] used a evolutionary optimization approach to optimize the serial robotic manipulator structure.

2.4 Practical Task-based Kinematic Synthesis of Serial Robotic Manipulators

Usually, the most common industrial 6-DOF robotic manipulators with a spherical wrist are designed for the desired rated payload while maximizing the manipulator's workspace envelope (Figure 2.2). For example, the IRB 8700 manipulator form ABB is designed to handle 550 kg with maximum reach of 4,20 m. In more advanced design scenarios, the manipulator's total orientation workspace (TOW) can be specified as a range of rotation angles the end-effector in a position inside the bounded workspace [27].



(a) The ABB IRB 8700 industrial manipulator (Photo: ABB) [47].

(b) A working range of the IRB 8700 manipulator (Photo: ABB) [47].

Figure 2.2: The IRB 8700 is a general purpose manipulator designed to fulfill requirements of different areas of industry. It is designed with focus on high production capacity, compact design and, low maintenance cost [47].

Clearly, for example, for tunneling construction this is not an adequate design approach. Instead, in more specific robotic manipulator design scenarios, the reachable workspace should be divided into several bounded subspaces that each has different requirements for the required range of wrist orientation angles. For example, in the tunnel drilling task (see Figure 2.3), the drilling pattern near the center point of the tunnel is more dense and requires a wide range of orientation angles to be reached. However, near the boundaries of the tunnel (walls, ceiling, and ground), the drilling pattern execution requires a smaller range of orientation angles to be reached. Therefore, this design problem consists of well-defined robotic manipulator design requirements with a large number of defined task positions to be reached in the workspace position that depends on a range of wrist orientation angles. Due to the complexity of the given design optimization problem, it is not obvious which robotic topology should be chosen for this given generic robotic manipulator design problem.

For example, most of the current heavy-duty hydraulic manipulator mechanisms are an aged technology, and their kinematics has been developed with a focus on the human operator controlling a hydraulically controlled system without numerical control input. As the trend in hydraulic manipulators is increased automation and usability, the requirements for robotic manipulator kinematics are different. Computer control enables a different kind of robotic manipulator kinematics, which is not optimum for direct control by a human operator, because the joint motions related to the different trajectories are not

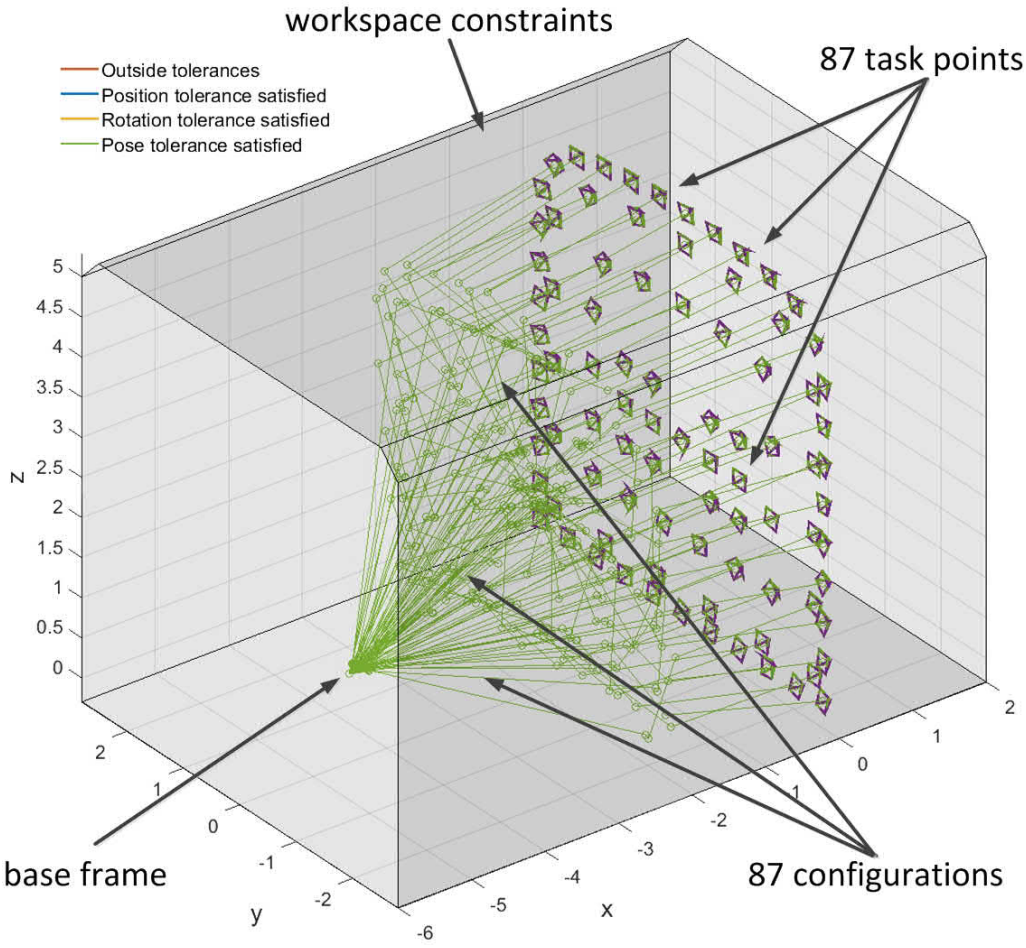


Figure 2.3: Final manipulator configurations for each task point.

native for the human mind. Numerically controlled robotic manipulators can accept kinematics that is more efficient in doing the job expected by the customer.

The task-based optimization method takes into account not only the desired task points but also the practical design constraints for the given task and the most important constraints, such as the available confined working envelope inside the workspace and joint ranges. In the generic approach, due to practical reasons, such as the resulting robot design cost-effectiveness, the set of robot topologies is limited to serial-type manipulators, and the maximum number of actuators must be defined. The studied task can consist of multiple task points where several real tasks are combined into a single set of task points. If the application requires a long-reach manipulator (10 – 15 meters) due to the size of the task space, manipulator topologies that have at least one prismatic joint are interesting; however, it is not clear whether six joints are better than seven joints. The selected set of robot topologies is optimized using an optimization method and ranked according to the selected performance measure. Our results indicate that this kind of method is effective in finding the optimal structure and dimensions for the serial-type manipulator from the

task specifications in terms of the selected performance measure.

3 State-of-the-Art Collision Avoidance of Serial Robotic Manipulators

In the introduction, the motivation for collision avoidance systems on serial robotic manipulators was described. These collision avoidance systems, for example, enable autonomous operations for robotic manipulators in unknown or in dynamic environments. Collision avoidance solutions for robotic manipulators, thus, have been developed in the literature. These solutions are more and more on-line collision avoidance path planning because the classical collision detection methods do not enable fully autonomous operations [48]. The considerable gain in computational power from control processing units in recent years enables the implementation of on-line collision avoidance control of robotic manipulators. Advanced sensor technology available on robotic systems for sensing the environment has recently improved to support this development.

This section covers state-of-the-art solutions in collision avoidance of serial robotic manipulators, with a focus on shortest distance query and real-time collision avoidance control. We first classify collision detection and shortest distance query terms and approaches in Section 3.1 and follow with an introduction to the principles and approaches of collision avoidance in Section 3.2. Next, the fundamental problem and the conventional solution for collision avoidance of serial robotic manipulators are covered in Section 3.3. Finally, in Section 3.4, state-of-the-art collision avoidance solutions applicable to serial robotic manipulators are reviewed.

3.1 Collision Detection and Shortest Distance Query

To prevent the robotic manipulator from colliding with obstacles or to prevent self-collision, a manipulator control system must detect obstacles before collisions. Another important feature of the collision avoidance system is finding the minimum distance between different parts of the manipulator and the obstacles, and between different parts of the manipulator itself. Finding the exact shortest distances between the manipulator and obstacles, as well as finding the exact possible collision location, are crucial parts of an accurate collision avoidance system. Collision detection has been well studied over the last few decades, and several methods for detecting collisions are reported in the literature, such as I-Collide [49], RAPID [50], Cullide [51], DeformDC [52], BoxTree [53], and Drop-tree [54]. However, these methods detect whether two objects interact or not. These methods do not provide exact distance queries which is one of the most important subjects of an efficient collision avoidance system. Therefore, these methods might not be applicable for real-time collision avoidance.

Several studies used bounding boxes to represent obstacles and the manipulator and then found the shortest distance between these bounding boxes; for example, Puiu and Moldoveanu [55] and Lee et al. [56] used oriented bounding boxes (OBBs). The shape of the moving obstacle, and the shape of the robotic manipulator parts, can be very complex, and the computation of the shortest distance between complex geometries is time-consuming. To simplify the collision detection procedure, the obstacles and the manipulator parts are represented using simple OBBs [50, 57]. This allows faster collision detection between the manipulator parts and the obstacles. However, with this approach, the exact shortest distance between these parts cannot be calculated, and also the exact collision location is lost. For continuous motions, several approaches have been developed to detect collisions, such as swept-volume methods [58, 59], the trajectory parameterization method [60], and feature-tracking methods [49, 61]. However, these methods are suitable only for simple geometries.

Sometimes bounding boxes (BBs) are used in the preliminary state to find an approximate location of the shortest distance. After this, the underlying actual geometry (usually a polygon model) is used to calculate the actual shortest distance. In the literature, several different libraries are presented for calculating the shortest distance between two objects. These libraries include methods such as V-Collide, PQP, and FreeSolid. What is common in all these methods is that they all use some kind of BBs to simplify the calculation process.

V-Collide, proposed by Hudson et al. [62], uses conservative approximation to find potential collision pairs of objects among the entire data set. This is achieved by using axis-aligned bounding boxes (AABBs) for every object in the scene. The pairs of objects whose BBs overlap are considered potential collision pairs. In the second phase, OBBs are used to check for collisions between a pair of objects. This methods can handle only medium-sized trees, because OBBs need to be rebuilt after the objects have moved. This method uses advantages of other libraries, such as I-Collide [49] and RAPID [50], which are mainly used for collision detection, not for distance queries.

Another popular method for detecting collisions and calculating the shortest distance between objects is a proximity query package (PQP) [50, 63], which is also based on the RAPID library. Therefore, a PQP uses also OBBs to represent objects. For distance queries, a PQP uses swept sphere volumes as bounding volumes. The PQP library supports three types of queries on a pair of geometric models: collision detection, distance computation, and tolerance verification.

FreeSolid is a collision detection library developed by den Bergen [64]. This method uses AABB hierarchies for collision detection. This implementation uses the Gilbert-Johnson-Keerthi (GJK) algorithm for computing the distance between convex objects. The algorithm uses various types of simple geometry primitives for collision detection. An advantage of this method is that it can also handle deformation of the geometry. This method also provides a fast collision test between objects, by using an acceleration scheme introduced in [65].

Another popular method for finding the shortest distance between obstacles is to use vision-based techniques [66, 67]. Although vision-based methods are widely used and are effective methods for calculating the shortest distance, they cannot be used in applications with dirty and harsh environments, such as in underground mining applications. Vision-based techniques are appearance-based or feature-based methods. Both techniques require a good enough image that can be used for recognition. Another feature of vision-based

techniques is that they most often require teaching or objects that should be recognized must be defined beforehand. Therefore, unexpected objects cannot be recognized.

In applications with challenging environments (such as lightning conditions) or where the shape of the obstacles can change during the operation, a technique that does not rely on predefined obstacles must be used. A system that can produce point clouds is preferable, such as a laser-based measurement system. A point cloud that represents an obstacle can be updated if the shape of the obstacle changes during the operation. The system is also able to create new point clouds if new unforeseen obstacles appear in the workspace. The use of a point cloud-based distance calculation method ensures that there is no extra volume around the obstacles that could lead to an incorrect shortest distance calculation.

Calculating the shortest distance between two point clouds is a challenge for a real-time control system. A method for detecting collisions and calculating the shortest distance between more complex geometries using point clouds is presented by Kaldestad et al. [68], in which calculations based on the graphics processing unit (GPU) are required to use the algorithm in real-time. Another solution to the computation resources problem is where the calculation procedure is divided into different stages, which ensures that the shortest distance can be found in real-time. The method for calculating the shortest distance between two point clouds can be divided into two parts: the initialization phase and the calculation phase. The initialization phase, as its name suggests, is performed in order to find the shortest distance from preliminary actions. Initialization actions are performed only once before the actual calculation phase. The calculation phase, in turn, is used to calculate the actual shortest distance between the point clouds. The poses of the point clouds are also updated based on a measurement or a surveillance system.

3.2 Collision Avoidance

Collision avoidance is an action or series of actions to prevent a collision after a possible collision is detected. Detection possible collisions is a crucial part of the collision avoidance system. The possible collision has to be detected early enough so that the collision avoidance system is able to prevent the actual collision. When making an attempt to avoid a collision, there are many decisions to make within a very short period of time. The decision how to avoid the collision depends on the available options the manipulator can make.

Collision avoidance is used in different areas of industry, such as automobile, robotics, spacecraft, telecommunications, etc. Collision avoidance systems have different meanings in different industries. What is common among these systems is that they are designed to prevent different kinds of collisions. For example, in the automobile and telecommunication industries, collisions are quite distinct from each other. Therefore, the rest of this thesis focuses only on collision avoidance of serial robotic manipulators.

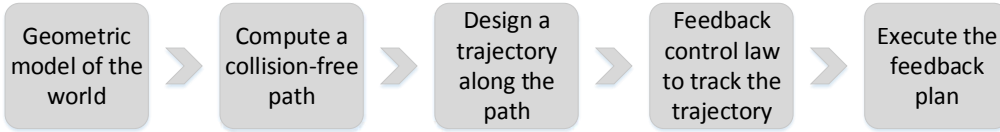


Figure 3.1: Global path planning strategy.

Collision avoidance methods can be categorized in a number of ways: Two main strategy types are global strategies and local strategies [69, 70]. Global collision avoidance methods find a collision-free path from the start pose to the goal pose if such a path exists (Figure 3.1). However, global methods are computationally heavy and may not be applicable for real-time control [59]. Furthermore, global methods usually do not rely on sensor feedback, and therefore, they cannot be used with dynamic environments. The environment has to be static and well-defined.

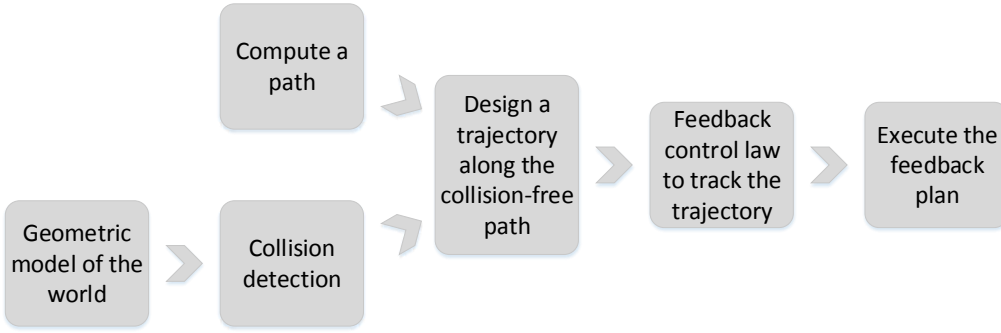
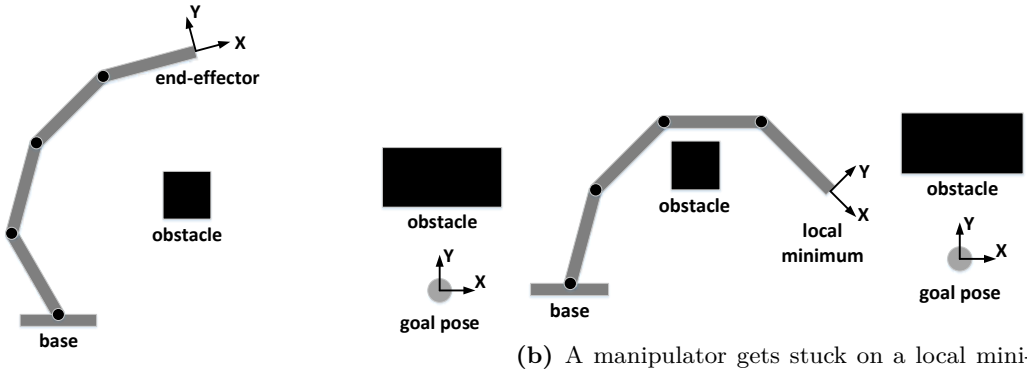


Figure 3.2: Local path planning strategy.

Local strategies treat obstacle avoidance as a control problem. The idea is to use low-level controllers to change the path of the manipulator if the obstacle appears in the current path of the manipulator (Figure 3.2). They use sensor information to detect obstacles in the workspace. Local methods are computationally lighter than global methods, and therefore, local methods can be used in real-time during the task execution. Usually, local methods cannot guarantee the optimal path for the manipulator, and they may even get stuck on a local minimum (Figure 3.3). Therefore, a combination of global and local methods, called hybrid systems, might be a good solution to avoid collisions [71, 72].



(a) A starting pose of a manipulator.

(b) A manipulator gets stuck on a local minimum.

Figure 3.3: A planar manipulator gets stuck on a local minimum when reaching from the starting pose to the goal pose using simple collision avoidance method based on artificial potential fields.

Local minimum issues are general in the use of local path control methods. These are situations where the attractive and repulsive forces affecting on the manipulator

compensate each other (Figure 3.3). This scenario causes a zero resultant force on the manipulator. Consequently, the movement of the manipulator stops and thus results in a failure of reaching the goal pose by ending into a local minimum.

3.3 Fundamental Problem and the Conventional Solution

At the velocity level, the fundamental problem of collision avoidance for robotic manipulators we are trying to solve is the inversion of

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}_G(\mathbf{q})\,\dot{\mathbf{q}}, \tag{3.1}$$

where $\dot{\mathbf{p}}$ and $\boldsymbol{\omega}$ represent the end-effector linear and angular velocities, respectively. The geometric Jacobian matrix $\mathbf{J}_G$ relates the joint velocity vector $\dot{\mathbf{q}}$ to end-effector velocity vector $\dot{\mathbf{x}}$ through the mapping Eq. (3.1). To achieve the desired end-effector velocity, we have to solve this equation while avoiding collisions.

The solution for Eq. (3.1) can be found through the inverse kinematics by inverting the Jacobian matrix

$$\dot{\mathbf{q}} = \mathbf{J}_G^{-1}(\mathbf{q})\,\dot{\mathbf{x}}. \tag{3.2}$$

Equation (3.2) is valid only for manipulators that have the same dimension in the operational space and the joint space ($m = n$). When the manipulator is redundant ($m < n$), the Jacobian matrix has more columns than rows, and an infinite number of solutions exist for Eq. (3.1). Therefore, there cannot be a direct Jacobian matrix inverse. If the Jacobian matrix is a non-square matrix, then a pseudo-inverse can be adopted [73]:

$$\dot{\mathbf{q}} = \mathbf{J}_G^{\dagger}(\mathbf{q})\,\dot{\mathbf{x}}, \tag{3.3}$$

and the matrix,

$$\mathbf{J}_G^{\dagger} = \mathbf{J}_G^T\left(\mathbf{J}_G\mathbf{J}_G^T\right)^{-1}, \tag{3.4}$$

is the right pseudo-inverse of $\mathbf{J}_G(\mathbf{q})$.

The pseudo-inverse has stability problems in a neighborhood of singularities. At a singularity, the Jacobian matrix no longer has a full row rank. If the configuration is close to a singularity, then the pseudo-inverse method will lead to large changes in the joint angles, even for small movements in the target position. The damped least squares method avoids many of the pseudo-inverse method's problems with singularities and can provide a numerically stable method for selecting $\dot{\mathbf{q}}$.

$$\dot{\mathbf{q}} = \mathbf{J}_G^T\left(\mathbf{J}_G\mathbf{J}_G^T + \lambda^2\mathbf{I}\right)^{-1}\dot{\mathbf{x}}, \tag{3.5}$$

where λ is a non-zero damping constant, and the right damped pseudo-inverse assumes the form

$$\tilde{\mathbf{J}}_G^{\dagger} = \mathbf{J}_G^T\left(\mathbf{J}_G\mathbf{J}_G^T + \lambda^2\mathbf{I}\right)^{-1}. \tag{3.6}$$

The damping constant depends on the details of the multibody and target positions and must be chosen carefully to make Eq. (3.5) numerically stable. If $\lambda = 0$, Eq. (3.3) becomes identical to Eq. (3.5), which is ill-conditioned near a singularity.

In [15], the general form of the solution for Eq. (3.1) is shown to be

$$\dot{\mathbf{q}} = \tilde{\mathbf{J}}_G^{\dagger}\dot{\mathbf{x}} + \left(\mathbf{I} - \tilde{\mathbf{J}}_G^{\dagger}\mathbf{J}_G\right)\mathbf{z}, \tag{3.7}$$

where $\mathbf{z}$ is an arbitrary vector in the $\dot{\mathbf{q}}$-space. A projection operator $\left(\mathbf{I} - \tilde{\mathbf{J}}_G^{\dagger}\mathbf{J}_G\right)$ describes the redundancy of the system and can be used to map an arbitrary $\dot{\mathbf{q}}$ into the null space of the transformation.

3.3.1 Robotic Manipulator Path Control

The path control for the end-effector of the robotic manipulator is achieved using two proportional controllers, one for the position and one for the orientation. Let $\mathbf{p}_d$ designate the goal position of the manipulator's end-effector; then the position controller is described as follows:

$$\dot{\mathbf{p}} = k_v \left(\mathbf{p}_d - \mathbf{p}\right), \tag{3.8}$$

where $\mathbf{p}$ is the current position of the end-effector and k_v is the position control coefficient.

In order to devise a trajectory generator for the end-effector orientation, a suitable orientation error must be defined. The end-effector orientation error depends on the choice of the orientation description. Direct computation of the orientation error requires the extraction of the Euler angles from the end-effector rotation matrix, which suffers from representation singularities [74]. Therefore, the orientation error is calculated based on the unit quaternion representation. The drawbacks of the Euler parameters can be overcome with the unit quaternion [73]. Let $\boldsymbol{\sigma}_d = \{\eta_d, \boldsymbol{\epsilon}_d\}$ and $\boldsymbol{\sigma} = \{\eta, \boldsymbol{\epsilon}\}$ represent the quaternions associated with $\mathbf{R}_d$ and $\mathbf{R}$, respectively. The orientation error is defined as

$$\mathbf{e}_O = \eta\boldsymbol{\epsilon}_d - \eta_d\boldsymbol{\epsilon} - \mathbf{S}(\boldsymbol{\epsilon}_d)\boldsymbol{\epsilon}, \tag{3.9}$$

where $\mathbf{e}_O$ is the orientation error and $\mathbf{S}(\boldsymbol{\epsilon}_d)$ is the skew-symmetric matrix for the $\boldsymbol{\epsilon}_d$. The end-effector's orientation controller can be described as

$$\boldsymbol{\omega} = k_w \mathbf{e}_O, \tag{3.10}$$

where k_w is the angular velocity control coefficient. Using Eqs. (3.1), (3.8), and (3.10), the manipulator path control takes the following form:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} k_v \left(\mathbf{p}_d - \mathbf{p}\right) \\ k_w \mathbf{e}_O \end{bmatrix}. \tag{3.11}$$

The linear and angular of velocities of the manipulator's end-effector are bounded. This allows the use of larger controller coefficients for the velocities, which ensures that the demanded end-effector velocities are also suitable near the goal pose. The end-effector velocities can be bounded

$$\dot{\mathbf{p}} = \begin{cases} \dot{\mathbf{p}} \frac{\dot{p}_{max}}{max(|\dot{\mathbf{p}}|)}, & \text{if } max(|\dot{\mathbf{p}}|) > \dot{p}_{max} \\ \dot{\mathbf{p}}, & \text{otherwise,} \end{cases} \tag{3.12}$$

where $\dot{p}_{max}$ is the maximum linear velocity of the end-effector and

$$\boldsymbol{\omega} = \begin{cases} \boldsymbol{\omega} \frac{\omega_{max}}{max(|\boldsymbol{\omega}|)}, & \text{if } max(|\boldsymbol{\omega}|) > \omega_{max} \\ \boldsymbol{\omega}, & \text{otherwise,} \end{cases} \tag{3.13}$$

where ω_{max} is the maximum angular velocity of the end-effector.

The collision-free path control is based on Eq. (3.7) and the approach proposed by [75]. The primary goal is to reach the goal pose with the manipulator's end-effector;

the secondary goal is to avoid collisions. In [75], the authors presented a method for avoiding collisions with one obstacle avoidance point. This approach is extended to cover multiple obstacle avoidance points. The primary and secondary goals are described by the equations

$$\mathbf{J}_G\dot{\mathbf{q}} = \dot{\mathbf{x}}, \tag{3.14}$$

$$\mathbf{J}_{O_i}\dot{\mathbf{q}} = \dot{\mathbf{x}}_{O_i}, \tag{3.15}$$

where $\mathbf{J}_{O_i}$ is the obstacle avoidance point Jacobian, and $\dot{\mathbf{x}}_{O_i}$ is the obstacle avoidance point velocity.

The obstacle avoidance point velocity can be calculated by using the shortest distance calculation method described in the Collision Detection section and by using the artificial potential field proposed by [76]:

$$\mathbf{U}_{O_i} = \begin{cases} \frac{1}{2}\mu\left(\frac{1}{\rho_i} - \frac{1}{\rho_O}\right)^2, & \text{if } \rho_i \leq \rho_O \\ 0, & \text{if } \rho_i > \rho_O, \end{cases} \tag{3.16}$$

where $\mathbf{U}_{O_i}$ is the repulsive artificial potential field, ρ_O is the limit for the collision avoidance, ρ_i is the distance between two obstacles, and μ is a scalar coefficient. Now the $\dot{\mathbf{x}}_{O_i}$ can be formulated to be equal with the negative gradient of the repulsive potential field defined in Eq. (3.16),

$$\dot{\mathbf{x}}_{O_i} = -\nabla\mathbf{U}_{O_i} = \begin{cases} \mu\left(\frac{1}{\rho_i} - \frac{1}{\rho_O}\right)\frac{1}{\rho_i^2}\frac{\delta\rho_i}{\delta\dot{\mathbf{x}}_{O_i}}, & \text{if } \rho_i \leq \rho_O \\ 0, & \text{if } \rho_i > \rho_O, \end{cases} \tag{3.17}$$

where $\frac{\delta\rho_i}{\delta\dot{\mathbf{x}}_{O_i}}$ is the direction of the collision line.

The secondary goal solution for $\dot{\mathbf{q}}$ is used as an arbitrary vector $\mathbf{z}$ to modify the solution to Eq. (3.7). By combining Eq. (3.7) and Eq. (3.15) with multiple obstacle avoidance points, the following solution can be derived:

$$\dot{\mathbf{q}} = \tilde{\mathbf{J}}_G^{\dagger}\dot{\mathbf{x}} + \sum_{i=1}^{n_o}\left[\left(\mathbf{I} - \tilde{\mathbf{J}}_G^{\dagger}\mathbf{J}_G\right)\tilde{\mathbf{J}}_{O_i}^{\dagger}\dot{\mathbf{x}}_{O_i}\right], \tag{3.18}$$

where n_o is the number of obstacle avoidance points, and $\tilde{\mathbf{J}}_{O_i}^{\dagger}$ is the damped right-pseudo-inverse solution for $\mathbf{J}_{O_i}$. If the desired path cannot be maintained, Eq. (3.18) gives up the desired velocity $\dot{\mathbf{x}}$ to ensure a collision-free path.

3.4 State-of-the-Art in Collision Avoidance

This section describes state-of-the-art collision avoidance by local strategies. Local control strategy solutions are very popular collision avoidance methods [70]. These methods threat obstacle avoidance as a control problem instead of a path planning problem. Thus, we change the path if obstacles are detected on the path or if obstacles move on the path. Therefore, these methods are suitable for unknown scenarios where there is no information about where the obstacles are during the desired manipulator operations.

Over the last few decades, many papers have described local collision avoidance methods for robotic manipulators. The general features of these methods are well-known, and

different approaches to solving this problem have been reported in the literature. Most of the proposed methods handle the local collision avoidance problem at the kinematic level.

Maciejewski and Klein [75] proposed a method that identifies for each period in time the point on the manipulator that is closest to an obstacle. This point is called the obstacle avoidance point. Then a desired velocity is assigned directly away from the obstacle surface. The desired end-effector velocity is treated as the primary goal and the secondary goal is obstacle avoidance. These two goals are combined with a null space motion concept [15]. This solution covers only a single obstacle avoidance point, and it must be extended to cover scenarios with multiple simultaneous obstacle avoidance points.

Another local collision avoidance method was proposed by Colbaugh et al. [77] and later further developed by Glass et al. [78]. In this approach, collision avoidance was treated as a configuration control problem. The main idea is to represent the collision avoidance requirements as a set of kinematic constraints that are satisfied during the desired end-effector trajectory. The simplicity and computational efficiency ensure that this approach can be used with a dynamically varying environment in real-time.

Guo and Hsia [79] proposed a local method for a manipulator to avoid obstacles. This method uses the optimization of an objective function that maximizes the distance between the manipulator and the obstacles. This algorithm is based on an extension of the joint trajectory command generator [80]. More recently, this same objective function optimization technique for collision avoidance was further developed by Dede et al. [81], where the authors used a redundant service robot arm to demonstrate the proposed method.

One of the most well-known approaches for collision avoidance is based on artificial potential fields (APFs). This approach was pioneered by Khatib [76]. Thereafter, the principle of the APF method was extended to support different collision avoidance algorithms. Within the APF method, the manipulator and obstacles can be thought to have the same charge, and the goal position of the manipulator acts as a different charge. Therefore, the manipulator and the obstacles repel each other by generating a repulsive force between them, and the goal position attracts the manipulator due to the opposite charge. Recently, Padula and Perdereau [69] further developed the APF method. One major drawback of this method is the local minima problem, which can trap the manipulator before it reaches its goal. The avoidance of local minima has been an active research topic with APFs. Kim and Khosla [82] used harmonic potential functions to avoid the local minima problem, and Volpe and Khosla [83] used superquadric potential functions to avoid the local minima problem. Furthermore, McLean and Cameron [84] used the virtual spring method to produce a trajectory for the entire manipulator, using the original path as a reference. Recently, some other effective methods for avoiding the local minima problem have been proposed by several authors [85 – 87].

Local methods that are based on potential fields in the manipulator's operational space are further divided into two categories based on how the methods use potential fields. When the collision avoidance method uses potential fields to generate forces on the manipulator [68, 76, 88 – 91], the solution requires position-based impedance control (PBIC) and a dynamic model of the system.

Khansari-Zadeh and Billard [90] presented an approach to real-time obstacle avoidance based on dynamical systems (DSs) that can prevent collisions with multiple convex-shaped objects. This approach is based on modulating the original dynamics of the controller. To guarantee collision-free task execution, the method requires an analytical equation

of the obstacles. Therefore, in the presence of unforeseen objects, this method cannot guarantee safe avoidance of the obstacles. Another limitation of the proposed collision avoidance method is that it considered obstacle avoidance only for a point manipulator. However, they suggested some algorithms that can be integrated in their method which may then be capable of avoiding link collisions. This integration was outside their work, and it is not yet proven to work.

Saveriano and Lee [91] proposed an algorithm that allows the manipulator to avoid obstacles and to reach the assigned goal. They used an improved DS modulation approach that ensures the avoidance of convex and concave obstacles. They called this method discrete modulation (DM) separate from previous continuous modulation (CM). The advantage of their method is that a modulation matrix is calculated directly from the point cloud data, without the need for an analytical representation of the obstacles. They used a Microsoft Kinect RGB-D sensor for monitoring the scene and detecting obstacles. Their experimental results showed that the proposed method can be used to avoid collisions between a manipulator's end-effector and static obstacles. They considered only collisions between a point manipulator and the obstacles. Collisions between a manipulator's links and obstacles and self-collisions were beyond the scope of their work.

Another approach for collision avoidance that uses potential fields, compared to DS modulation, is to generate joint velocities directly [69, 82]. This approach does not require a dynamic model of the manipulator and therefore is easier to implement. Kim and Khosla [82] used the APF method together with a null space modification ([15]) to generate joint trajectories and at the same time, avoid collision with obstacles. They used the panel method ([92]) to represent arbitrary shaped obstacles and to derive the potential over the whole workspace. With this method, they were able to use harmonic potential, which is free from local minima. They demonstrated the proposed solution with a mobile robot and a planar 3-DOF redundant manipulator. According to the authors, this method can be expanded to cover a three-dimensional (3D) workspace. However, the proposed method assumes that the obstacles are static and their panels are calculated off-line.

Padula and Perdereau [69] used potential fields in order to control the joint velocities to make the manipulator avoid collisions with obstacles. They used the null space-projection method with a weighted pseudo-inverse matrix which improves the manipulator's capability to find a feasible path. This method does not require calculation of the dynamics of the manipulator, which makes the implementation of the proposed method simpler. They demonstrated the proposed solution with 7-DOF planar and spatial redundant manipulators with simulations. The detection of the obstacles and the calculation of the shortest distance between the manipulator and obstacles, as well as the experimental verification, were not discussed in their work.

4 Summary of Publications

This thesis as a compendium comprises four research publications and an unpublished manuscript, each of which is briefly summarized below. Publications **P−I** and **P−II** deal with the kinematic synthesis of serial robotic manipulators, publications **P−III**, **P−IV**, and **P−V** deal with the collision avoidance of serial robotic manipulators, and discusses a collision detection and distance query between two objects.

4.1 P-I: A Generic Method to Optimize a Redundant Serial Manipulator Structure

In this paper, a generic method for optimizing a redundant serial manipulator structure is presented. The method finds an optimal solution in terms of an LM method for a constrained optimization problem. The case study shows that the proposed method can be used to optimize a serial manipulator structure.

To verify that the developed algorithm can optimize a serial manipulator's structure, the algorithm was tested with an 8-DOF redundant serial manipulator, the Sandvik SB60 hydraulic manipulator. The SB60 manipulator has six revolute joints and two prismatic joints. Furthermore, real-world underground tunneling drilling patterns for holes were used to find the optimal structure for a manipulator to drill these holes. The objective was to optimize the manipulator's structure by modifying the manipulator link lengths and the position of the coordinate system of the manipulator base in order to satisfy a number of design constraints. Thus, several hundreds of parameters in total were optimized with the proposed method.

By looking the results, we notice that some of the changes in the DH parameters are very significant. There are several reasons for this, but one thing is that the working area of the original manipulator is somewhat larger than the working area for which it is optimized. Therefore, the optimized manipulator can be smaller than the original one which can be seen as a change in a DH parameter. However, if the task points in the example reflect what is to be achieved by the manipulator, then the original manipulator is too large for the desired task; therefore, it is not the optimal manipulator for the task. Due to the optimal design, the new manipulator could save time and energy.

4.2 P-II: A Method for Task-based Optimization of a Mining Rig's Serial Manipulators with Arbitrary Topology

In this paper, a generic method for selecting and optimizing a redundant serial manipulator structure was presented. The method finds an optimal solution in terms of the LM method, which was presented in **P−I**, for a constrained optimization problem. The case study

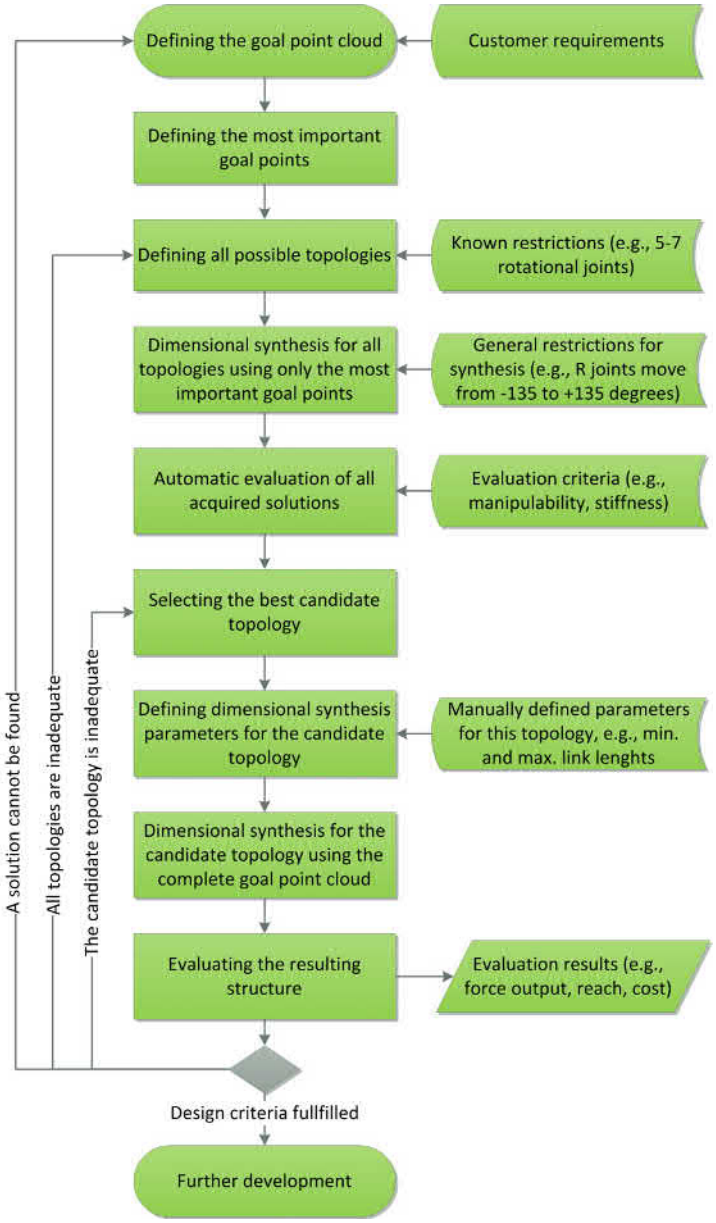


Figure 4.1: Simplified synthesis process of a robotic manipulator

showed that the proposed method could be used as a design tool for finding the best structure for a manipulator with specific constraints.

The paper presents an overview and implementation of the complete process flow from the user requirements to an acceptable mechanical structure with a defined number of links of known lengths and connected by joints of known types. In $\mathbf{P}-\mathbf{I}$, an existing topology was optimized to find the optimal dimensions for that topology. In this paper, a set of pre-chosen topologies are optimized to find the optimal manipulator topology and the optimal manipulator dimensions for the given problem. The number of different

topologies are limited to maintain the readability of the paper.

The flow of the proposed synthesis process is depicted in Fig. 4.1. The synthesis process is started by defining the task that the manipulator should be able to fulfill. In general, the task consists of several poses that need to be reached by the manipulator with an attached tool. In addition to the locations and orientations given with the required degrees of freedom, tasks may include other required attributes, such as the required force output and structural stiffness.

The next step is to define all possible topologies. This is referred to as the topology synthesis. Topology in this context means the number of links and the number, types, and ordering of the joints that connect the links to each other. To limit the infinite number of topologies imaginable, known restrictions should be taken into account before or during the topology synthesis.

Once the list of possible topologies has been defined, it is time to perform a dimensional synthesis for all topologies. A dimensional synthesis is used to calculate the required link lengths, given the goal points with respect to the machine's base coordinates and the topology of the structure. At this point in the process, constraints that are common to all topologies should be considered in the synthesis algorithm. For example, it may be desirable to limit the motion range of all rotational joints to some value and assign minimum lengths for all links. If environmental entities, such as obstacles or walls, are to be included in the evaluation, they should be included in the synthesis algorithm at this stage of the process.

After the dimensional synthesis has been performed for all topologies and feasible solutions (those able to reach all task points) have been evaluated and ranked, the designer should select a group of the best candidate solutions from the ranked list for a further synthesis. An evaluation of the resulting structure should be automatically generated by the design software used. The designer can then choose to change some of the specific synthesis parameters and run the evaluation again.

This process assumes that the initially generated number of topologies is high, that the task contains a large number of points, and that the kinematic synthesis algorithm is resource-intensive. If these assumptions can be relaxed, the method gives the best kinematic structure for building the manipulator. Thereafter, the link arms must be sized with the proper material strengths, and the joints need to find suitable actuators. If this is not possible for the structure produced by the optimization method, then one has to add such constraints that the link lengths and joint actuators can be implemented and do the optimization again with the proper constraints.

The result showed that the drilling pattern of the holes of the case study could be achieved with many topologies. Which topology is the best depends on the application. In this case, the global conditioning index and the kinematic conditioning index were used to rank the optimized topologies to find the best manipulator structure. If the application is different from this case study, different performance measures [24] can be used to rank the manipulators and to find the best manipulator structure.

In this paper, six topologies were optimized to show how the proposed method could be used to find the most attractive manipulator structure. However, in general, it is possible to automate the kinematic synthesis process so that all possible topologies are generated automatically and then optimized. Then the performance indices are calculated for every topology, and the topologies are ranked according to the selected performance method. This procedure ensures that the suitable manipulator structure is found.

4.3 P-III: Redundant Robotic Manipulator Path Planning for Real-Time Obstacle and Self-Collision Avoidance

Collision avoidance systems are one of the most challenging problems for autonomous manipulators to accomplish the given task. An improved shortest distance-based collision avoidance method was proposed to obtain a collision-free path from the starting pose to the goal pose. Multiple simultaneous collision avoidance points can be used to avoid self-collisions and any part of the manipulator from colliding with obstacles or the environment. The proposed method can be used with a real-time control system, and the method does not require any path planning time, because all decisions are made on the fly. An advanced shortest distance calculation method ensures that the shortest distance between point cloud-type objects can be calculated in real-time. The shortest distance calculation method also minimizes the calculation power needed to find the shortest distance.

In order to validate the collision avoidance system, we conducted the following test. The task was to move the manipulator from a starting pose to a goal pose by avoiding self-collision. The starting pose was at the left side of the manipulator's base with a diagonally upward rotated end-effector. The goal pose was at the other side of the manipulator with the same end-effector pose except the y-coordinate was the inverse. With a conventional control system, the end-effector of the manipulator collided with the lift part of the manipulator, if it tried to move from the starting pose to the goal pose. The proposed collision avoidance control system of the manipulator executed the desired task without self-collision by finding a collision-free path.

4.4 P-IV: On-line Path Planning with Collision Avoidance for Coordinate Controlled Robotic Manipulators

For autonomous manipulators to accomplish the given task, one of the most challenging problems is collision avoidance. An improved shortest distance-based collision avoidance method was proposed to obtain a collision-free path from the starting pose to the goal pose. Multiple simultaneous collision avoidance points can be used to prevent self-collisions and collisions with other manipulators, and any part of the manipulator from colliding with obstacles or the environment. The proposed method can be used with a real-time control system, and the method does not require any path planning time because all decisions are made spontaneously.

To verify the functionality of the collision avoidance control system, simulation and experimental tests were carried out with a heavy-duty hydraulic manipulator, an 8-DOF redundant serial manipulator, the Sandvik SB60 hydraulic manipulator. During the simulation and the experiment, the collision server, which was used to calculate the distances between collision pairs, ran on a laptop with an Intel Core i7 processor. The server program to calculate the shortest distance between point clouds was programmed with C++/CLI and compiled to a standalone program. The average time to calculate the distance was 8 ms. This ensured that the proposed method could be used in a real-time control system.

In the simulation case, two manipulators were used simultaneously to carry out the drilling task. The task was to drill four holes in a vertical line, the yz-plane. The vertical distance between the two holes was 1 m, and they were horizontally in the middle of the bases of the manipulators. The holes had to be drilled in order starting from the lowest. If the manipulator was in the drilling state, its pose was locked, and the manipulator could

not change the configuration to avoid collisions. Instead, other manipulators had to avoid this manipulator. When the drilling of the current hole was complete, the manipulator moved on to the next available hole. During movement from one pose to another, each manipulator avoided self-collisions and collisions with other manipulators.

4.5 P-V: Real-time Distance Query and Collision Avoidance for Point Clouds with a Heavy-duty Redundant Manipulator

The collision avoidance control system was developed to enable autonomous operation of a redundant serial manipulator. To verify this functionality, an experimental test was carried out with a heavy-duty 4-DOF hydraulic serial manipulator, the Hiab XS 033. The Hiab XS 033 manipulator is a hydraulic multi-purpose manipulator for almost a limitless range of applications. This manipulator has three revolute joints and one prismatic joint. In the case study, the manipulator was equipped with a grabber that included two free non-actuated joints. During the experiment, the TCP of the manipulator was controlled, and the grabber hung from the TCP parallel to the gravity.

For the experiment, the 3D part of the grabber that could collide with the obstacle was converted into a point cloud from a computer-aided design (CAD) model, containing 1000 points. The environment and the obstacle were scanned with the laser scanning system, containing 5500 points. Thereafter, we created rules for monitoring the shortest distance between possible collision pairs. If the distance between the collision pair was larger than the collision margin, then the distance had no influence on the control of the manipulator, but if the distance was smaller than the collision margin, we modified the trajectory of the manipulator based on the calculated shortest distance between point clouds.

5 Discussion

This chapter presents a summary of how the RP, divided into RP1-RP7 as described in Section 1.1, is addressed.

5.1 Optimal Structure (RP1)

The kinematic synthesis solution should be able to optimize the kinematic structure of a device from the task description. This is referred to as a topology synthesis, and it is used to optimize the number, type, and arrangement of joints and the links connected by these joints.

The topology synthesis, also called the structure optimization, involves determining the structure of a device, excluding link lengths, from the task description. The word structure, in this context, is used to describe the number, type, and arrangement of joints and the links connected by these joints. Ideally, the topology synthesis results in the optimal topology for the specific task, and this topology is then subjected to the dimensional synthesis to determine the link lengths.

There is no general approach, in the literature, for the topology synthesis of open-chain serial manipulators. The reason is that it is extremely difficult to apply general rules to optimize the number and type of joints for the manipulator. Therefore, instead of trying to automatically optimize the most suitable topology, we optimized several different topologies in terms of link lengths in **P−II**. Thereafter, suitable performance measures can be used to rank the optimized structures and select the most attractive one.

In **P−II**, six topologies were optimized to show how the proposed method can be used to find the most attractive manipulator structure. However, in general, it is possible to automate the kinematic synthesis process so that all possible topologies are generated automatically and then optimized. Then the performance indices are calculated for every topology, and the topologies are ranked according to the selected performance method. This procedure ensures that the suitable manipulator structure is found.

5.2 Optimal Dimensions (RP2)

The kinematic synthesis solution should be able to optimize the dimensions of a device. This is referred to as a dimensional synthesis.

All the kinematic synthesis solutions proposed in research publications **P−I** and **P−II** are dimensionally optimal. In these research papers, a generic method for selecting and optimizing a redundant serial manipulator structure was presented. The presented method for the dimensional synthesis uses a nonlinear LM method to solve the synthesis problem.

The LM algorithm is an iterative technique that finds the minimum of a multivariate function. The LM interpolates between the Gauss-Newton method and the steepest descent gradient method.

The proposed method gives only the skeletal structure of the manipulator. After the optimization, the link arms must be sized with the proper material strengths, and the joints need to have suitable actuators. In some cases, this may not be possible to achieve for the structure produced by the optimization process. Based on engineering knowledge, it is possible to add constraints, so that the link lengths and joint actuators can be implemented.

5.3 Task-based Design (RP3)

The kinematic synthesis solution should be able to fulfill the desired task space definitions for each task point.

Most commonly, industrial robotic manipulators with six degrees of freedom and spherical wrists are designed to achieve the desired payload while maximizing the robot's workspace envelope. In more advanced design scenarios, the robot's total orientation workspace can be specified as the range of the end-effector's rotation angles inside a bounded workspace [27]. This type of design can be classified as general-purpose design. Although using a general-purpose design is generally acceptable, it does not guarantee the optimal design for task execution. Therefore, task-based optimization is preferred when a task is predefined.

The task-based optimization or synthesis of a robotic manipulator consists of finding a set of manipulator design parameters so that the required task points and kinematic requirements are fulfilled. The kinematic requirements, such as a constrained workspace, design parameter limits, and joint limits, are task specifications that affect the kinematic structure of the robotic manipulator.

In research papers **P−I** and **P−II**, we proposed an optimization method that takes into account not only the desired task points but also practical design constraints for given task points, including the most important constraints, such as the available confined working envelope and the joint ranges.

5.4 General Applicability (RP4)

The kinematic synthesis solution should be applicable to multi-DOF robotic manipulators (joint space) and 6-DOF Cartesian space.

In spite of the number of previous studies, the use of the task-based kinematic synthesis for an arbitrary manipulator structure with practical relevance has remained restricted. Thus, additional studies of the task-based optimization for an arbitrary manipulator to find the optimal values for the design parameters, the base frame, and the joint values are needed. This means that there are several hundreds of parameters in total that need to be optimized. Therefore, the aim of research papers **P−I** and **P−II** was to extend the previous works to cover a more general serial manipulator optimization problem with practical relevance. The proposed optimization method can handle non-redundant and redundant manipulators. Furthermore, it can handle revolute and prismatic joints and n-dimensional task space, including the end-effector position and orientation. The proposed method can also handle a large number of task points.

In research papers **P−I** and **P−II**, we proposed an optimization method that takes into account not only the desired task points but also the practical design constraints for the given task and the most important constraints, such as the available confined working envelope and the allowed joint ranges. In the generic approach proposed in these papers, due to practical reasons, such as the resulting robot design cost-effectiveness, the set of robot topologies was limited to serial-type manipulators with a maximum number of seven actuators. The studied task space consisted of multiple task points where several real-life tasks were combined into a single set of task points.

5.5 Distance Query (RP5)

The collision avoidance system should detect possible collisions between the manipulator and obstacles or the manipulator's self-collisions in order to prevent these collisions. Possible collisions should be detected, and the shortest distance between these objects should be calculated so that there is enough time to react to these warnings.

Accurate collision point detection and distance query are the key elements of advanced collision avoidance systems. The research papers **P−III** through **P−V** presented a novel approach for calculating the shortest distance between two objects with a complex structure at the real-time rate. The proposed method is based on point cloud data that is used to describe the objects. Rigid point clouds can be converted from 3D CAD models whereas dynamic point clouds, such as surrounding obstacles, can be received from point cloud sensors. Calculation of the shortest distance is performed by using bounding boxes and the octree data structure for efficient computing. The proposed shortest distance query algorithm is applied to a real-time experiment and a simulation case in research papers **P−III** through **P−V** to validate the algorithm. Results demonstrated that the proposed method can be used in real-time control applications.

5.6 High Performance (RP6)

The collision avoidance system should provide high performance while satisfying the accuracy of the manipulator. The performance of the collision avoidance system should be experimentally validated to meet the real-time requirement.

A high-performance collision avoidance control system was one of the main objectives in this thesis. This requirement for high performance included that the control system of the manipulator determines the joint references so that the goal pose can be reached without any collisions, in real-time. The control system checks whether any part of the manipulator is at risk of colliding with itself or with any obstacles. If there is a risk of collision, the joint trajectories of the manipulator are modified so that collisions will be avoided while at the same time, the trajectory of the end-effector maintains its initial trajectory if possible.

High performance of the proposed collision avoidance method was observed in the simulations presented for an 8-DOF underground drilling manipulator in **P−IV** and the experiments presented for an 8-DOF underground drilling manipulator in **P−III** and **P−IV** and for a 4-DOF forestry hydraulic manipulator in **P−V**. All simulations and experiments were able to find a collision-free path from the starting pose to the goal pose, compared to the conventional control system without collision avoidance capabilities, which would have failed to reach the goal pose. The experiments validated, with different

collision scenes and with different manipulators, that the proposed method can be used in a variety of applications.

5.7 General Applicability (RP7)

The collision avoidance system should be applicable to kinematically different robotic manipulators. The solution should be suitable for use by multi-DOF manipulators that contain prismatic and revolute joints.

To achieve as broad as possible applicability for the proposed collision avoidance system, the system was designed for an 8-DOF robotic manipulator that has prismatic and revolute joints in **P−III** and **P−IV**. In these research papers, 6-DOF Cartesian space was used as the task space. The proposed collision avoidance system uses generic kinematic definitions so that the kinematics does not need to be hard coded. This ensures general applicability for different kinds of serial manipulator structures, only by changing the kinematic parameter definitions, namely, the DH parameters. General applicability was tested with another completely different manipulator, a 4-DOF kinematically redundant robotic manipulator with 3-DOF Cartesian space, in **P−V**.

Two kinematically different manipulators, the RRPRRRRP type for 6-DOF Cartesian space and the RRRP type for 3-DOF Cartesian space (xyz-space), were used for experimental verification of the general applicability. The proposed collision avoidance control system takes the manipulator's kinematic parameters as an input and makes all the required kinematic calculations based on these parameters.

6 Conclusions and Future Work

This compendium thesis introduces existing approaches and proposes suitable approaches for kinematic synthesis methods and collision avoidance control methods for serial robotic manipulators. These advanced design and control solutions increase the performance and the automation level of robotic manipulators. These two solutions are tightly connected, and they will play an important role in future intelligent autonomous serial robotic manipulators.

6.1 Kinematic Synthesis of Serial Robotic Manipulators

Current serial robotic manipulator mechanisms are out-of-date technology, when talking about heavy-duty industry with hydraulically driven manipulators, and their kinematics has been developed with a focus on a human operator maneuvering a hydraulically controlled system. As the trend is increasing automation, the requirements for manipulator kinematics are different. Computer control enables a different kind of manipulator kinematics, which is not optimum for direct control by a human operator, because the joint motions related to the different trajectories are not native to the human mind. Therefore, it is beneficial to design a new manipulator structure for autonomous control.

Kinematic synthesis is a large research field, in which a number of different approaches have been proposed. Task-based methods are increasingly gaining more attention compared to other methods. This trend can be seen as original equipment manufacturers' (OEMs') interest in serving their customers with better products that can be guaranteed to carry out the desired task. However, not all of these task-based methods are suitable for optimizing other than conventional industrial 6-DOF manipulators, which are not suitable for all tasks. Consequently, a novel solution was developed for the kinematic synthesis of a serial robotic manipulator that is not a standard industrial manipulator, in this thesis. The thesis results demonstrate that a serial robotic manipulator structure and dimensions can be optimized from task specifications to fulfill customer requirements. However, this method gives a kinematic structure for building the manipulator. Thereafter, the link arms must be sized with the proper material strengths, and the joints need to find suitable actuators. If this is not possible for the structure produced by the optimization method, one has to add such constraints that the link lengths and joint actuators can be implemented and perform the optimization again with the proper constraints. Although the proposed method uses optimization techniques, it might require the iteration rounds to find the solution that satisfies other requirements outside kinematic requirements.

6.2 Collision Avoidance of Serial Robotic Manipulators

Recent surveys confirmed that there is high industrial interest in digital and automated solutions among robotic manipulators. Although these surveys focused on a small area of industry, the same trend can be seen in other industries as well. It is predicted that this trend will continue as companies invest in new technology to improve their operations and competence. Automated control strategies would obviously be an important step toward unmanned autonomous machines.

Collision avoidance plays an important role in the development of fully autonomous control system for a robotic manipulator. Increasing the intelligence of the control system of the manipulator enables manipulators to reach the desired goal even if the optimal path is blocked by an obstacle. An advanced collision avoidance system is able to detect possible collisions early enough, so that the collisions can be prevented and a new collision-free path to the goal pose can be calculated.

In harmony with these trends, a collision avoidance control system for robotic manipulators is considered. The coordinated motion control system of the robotic manipulator resolves joint references so that a goal pose can be reached in real-time without any collisions. The proposed method is able to detect and prevent different types of possible collisions, including self-collisions and collisions with obstacles. During coordinated motion control, the joint trajectories of the redundant manipulator are modified so that the collisions can be avoided, while at the same time, the trajectory of the end-effector maintains its initial trajectory if possible. Experimental results for two kinematically different serial robotic manipulators demonstrated the capability of the proposed collision avoidance control system.

6.3 Future Work

Because of the limited scope of this thesis, some parts of kinematic synthesis and collision avoidance were excluded. These properties of both topics can be developed in future work.

6.3.1 Kinematic Synthesis of Serial Robotic Manipulators

Concerning the task-based kinematic synthesis of serial robotic manipulators, some properties were left for future work. Selecting task points for the kinematic synthesis were omitted. Selection of the proper and descriptive task points is an important part of kinematic synthesis. If the task space is not defined with great detail or it contains too many task points, it is not clear how the task point should be selected for the synthesis. Task points can be distributed homogeneously over the whole workspace, or more intelligent interval analysis methods can be used. However, these solutions were left for future work.

The proposed kinematic synthesis method, in this research work, gives only the skeletal structure of the manipulator, and the placement or the implementation of joints was not considered. For example, in heavy-duty robotic manipulators, the joints have to be hydraulic due to the high torque requirements. The most desirable hydraulic joint is driven by a hydraulic cylinder, which is more cost-effective and lighter in weight than hydraulic motors. Therefore, it is profitable to use hydraulic cylinders whenever possible.

The optimal placement of these actuators can be taken into account during the kinematic synthesis.

6.3.2 Collision Avoidance of Serial Robotic Manipulators

Concerning collision avoidance of a serial robotic manipulator, some properties were also left for future work. Allowing environmental obstacles to deform or move during the execution of the task was omitted. Allowing obstacles to deform their shapes and move during the task's execution is an important feature of some applications. However, the proposed collision avoidance system was designed to include this feature also. For future work, the implementation of collision avoidance for a dynamic environment is straightforward.

Determining the collision avoidance path, if multiple paths exist, is important for an efficient collision avoidance system, which was omitted in this thesis. It might be useful to check if the selected path does not lead into a local minimum or a death end.

Bibliography

[1] R. Jazar, *Theory of Applied Robotics: Kinematics, Dynamics, and Control (2nd Edition)*. Springer US, 2010. [Online]. Available: https://books.google.fi/books?id=hHS40c6sYucC

[2] Sandvik Mining and Construction, "Sandvik dd321," http://mediabase.sandvik.com/smc/, 2017, accessed: 2017-05-22.

[3] "Technological transformation," http://minestories.com/technological-transformation/, 2017, accessed: 2017-05-31.

[4] L. Beiner and J. Mattila, "An improved pseudoinverse solution for redundant hydraulic manipulators," *Robotica*, vol. 17, no. 2, p. 173–179, 1999.

[5] J. Mattila and T. Virvalo, "Energy-efficient motion control of a hydraulic manipulator," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 3, 2000, pp. 3000–3006 vol.3.

[6] S. Chiaverini, B. Siciliano, S. Member, and L. Villani, "A survey of robot interaction control schemes with experimental comparison," in *IEEE Int. Conf. Robotics and Automation, 2003*, 1999, pp. 273–285.

[7] B. Heinrichs, N. Sepehri, and A. B. Thornton-Trump, "Position-based impedance control of an industrial hydraulic manipulator," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, Apr 1996, pp. 284–290 vol.1.

[8] L. Zhou and S. Bai, "A new approach to design of a lightweight anthropomorphic arm for service applications," *Journal of Mechanisms and Robotics*, vol. 7, no. 3, 2015.

[9] T. J. Graettinger and B. H. Krogh, "The acceleration radius: a global performance measure for robotic manipulators," *IEEE Journal on Robotics and Automation*, vol. 4, no. 1, pp. 60–69, Feb 1988.

[10] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[11] M. Thomas, H. C. Yuan-Chou, and D. Tesar, "Optimal actuator sizing for robotic manipulators based on local dynamic criteria," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 107, no. 2, pp. 163–169, 1985.

[12] C. A. Klein, "Use of redundancy in the design of robotic systems," in *Robotics Research, The Second International Symposium*, 1984, pp. 207–214.

[13] M. V. Kircanski, "Robotic isotropy and optimal robot design of planar manipulators," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, May 1994, pp. 1100–1105 vol.2.

[14] E. Freund, M. Schluse, and J. Rossmann, "Dynamic collision avoidance for redundant multi-robot systems," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 3, 2001, pp. 1201–1206 vol.3.

[15] T. N. E. Greville, "The pseudoinverse of a rectangular or singular matrix and its application to the solution of systems of linear equations," *SIAM Review*, vol. 1, no. 1, pp. 38–43, 1959. [Online]. Available: http://www.jstor.org/stable/2028031

[16] T. Kivelä, J. Mattila, and J. Puura, "A generic method to optimize a redundant serial robotic manipulator's structure," *Automation in Construction*, vol. 81, pp. 172 – 179, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580517305150

[17] T. Kivelä, J. Mattila, J. Puura, and S. Launis, *Redundant Robotic Manipulator Path Planning for Real-Time Obstacle and Self-Collision Avoidance*, ser. Advances in Service and Industrial Robotics. RAAD 2017. Mechanisms and Machine Science. Cham: Springer International Publishing, 2018, vol. 49, pp. 208–216. [Online]. Available: https://doi.org/10.1007/978-3-319-61276-8_24

[18] T. Kivelä, J. Mattila, J. Puura, and S. Launis, "On-line path planning with collision avoidance for coordinate controlled robotic manipulators," in *Proceedings of ASME/BATH 2017 Symposium on Fluid Power and Motion Control, FPMC2017, October 16-19, 2017, Sarasota, Florida*, ser. Symposium on Fluid Power and Motion Control, 2017, p. ??

[19] T. Kivelä, J. Mattila, J. Puura, and S. Launis, "Real-time distance query and collision avoidance for point clouds with heavy-duty redundant manipulator," in *2017 IEEE 8th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Nov 2015, pp. 1–6.

[20] S. Patel and T. Sobh, "Task based synthesis of serial manipulators," *Journal of Advanced Research*, vol. 6, no. 3, pp. 479 – 492, 2015.

[21] R. S. R. S. Hartenberg and J. Denavit, *Kinematic synthesis of linkages*. New York : McGraw-Hill, 1964, includes index.

[22] M. A. Pucheta and A. Cardona, "Topological and dimensional synthesis of planar linkages for multiple kinematic tasks," *Multibody System Dynamics*, vol. 29, no. 2, pp. 189–211, Feb 2013. [Online]. Available: http://dx.doi.org/10.1007/s11044-011-9294-3

[23] "The fourbar mechanism," http://iitkgp.vlab.co.in/?sub=40&brch=126&sim=1273&cnt=1, 2012, accessed: 2017-05-25.

[24] S. Patel and T. Sobh, "Manipulator performance measures - a comprehensive literature survey," *Journal of Intelligent Robotic Systems*, vol. 77, no. 3, pp. 547–570, 2015.

[25] L. J. Stocco, S. E. Salcudean, and F. Sassani, "On the use of scaling matrices for task-specific robot design," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 958–965, Oct 1999.

[26] J. H. Lee, K. S. Eom, and I. I. Suh, "Design of a new 6-dof parallel haptic device," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 1, 2001, pp. 886–891 vol.1.

[27] J.-P. Merlet, C. M. Gosselin, and N. Mouly, "Workspaces of planar parallel manipulators," *Mechanism and Machine Theory*, vol. 33, no. 1, pp. 7 – 20, 1998.

[28] B. Paul, "A reassessment of grashof's criterion," *Journal of Mechanical Design*, vol. 101, no. 3, pp. 515 – 518, 1979.

[29] R. Vijaykumar, K. J. Waldron, and M. J. Tsai, "Geometric optimization of serial chain manipulator structures for working volume," *Int. J. Rob. Res.*, vol. 5, no. 2, pp. 91–103, jul 1986.

[30] R. Li and J. S. Dai, "Orientation angle workspaces of planar serial three-link manipulators," *Science in China Series E: Technological Sciences*, vol. 52, no. 4, pp. 975–985, 2009. [Online]. Available: http://dx.doi.org/10.1007/s11431-009-0083-7

[31] S. H. Patel and T. Sobh, "Optimal design of three-link planar manipulators using grashof's criterion," *Prototyping of Robotic Systems: Applications of Design and Implementation*, pp. 533–537, 1986.

[32] K.-L. Ting, "Five-bar grashof criteria," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 108, no. 4, p. 70, 2012. [Online]. Available: http://dx.doi.org/10.1115/1.3258765

[33] S. Bai, "Geometric analysis of coupler-link mobility and circuits for planar four-bar linkages," *Mechanism and Machine Theory*, vol. 118, no. Supplement C, pp. 53 – 64, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094114X16304864

[34] J. Angeles and C. S. López-Cajún, "Kinematic isotropy and the conditioning index of serial robotic manipulators," *Int. J. Rob. Res.*, vol. 11, no. 6, pp. 560–571, dec 1992.

[35] J. K. Salisbury and J. J. Craig, "Articulated hands," *The International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982. [Online]. Available: http://dx.doi.org/10.1177/027836498200100102

[36] T. M. Sobh and D. Y. Toundykov, "Optimizing the tasks at hand [robotic manipulators]," *IEEE Robotics Automation Magazine*, vol. 11, no. 2, pp. 78–85, June 2004.

[37] S. Kucuk and Z. Bingul, "Robot workspace optimization based on a novel local and global performance indices," in *Proceedings of the IEEE International Symposium on Industrial Electronics, 2005. ISIE 2005.*, vol. 4, June 2005, pp. 1593–1598.

[38] S. Kucuk and Z. Bingul, "Comparative study of performance indices for fundamental robot manipulators," *Robotics and Autonomous Systems*, vol. 54, no. 7, pp. 567 – 573, 2006.

[39] C. J. J. Paredis and P. K. Khosla, "Kinematic design of serial link manipulators from task specifications," *The International Journal of Robotics Research*, vol. 12, no. 3, pp. 274–287, 1993. [Online]. Available: http://dx.doi.org/10.1177/027836499301200306

[40] J. O. Kim and P. K. Khosla, "A formulation for task based design of robot manipulators," in *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*, vol. 3, Jul 1993, pp. 2310–2317 vol.3.

[41] J. O. Kim and P. K. Khosla, "Design of space shuttle tile servicing robot: an application of task based kinematic design," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, May 1993, pp. 867–874 vol.3.

[42] E. Singla, S. Tripathi, V. Rakesh, and B. Dasgupta, "Dimensional synthesis of kinematically redundant serial manipulators for cluttered environments," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 585–595, 2010.

[43] J. O. Kim, "A multi-population genetic algorithm and its application to design of manipulators," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Raleigh, NC, 1992*, pp. 279–286, 1992.

[44] W. K. Chung, J. Han, Y. Youm, and S. H. Kim, "Task based design of modular robot manipulator using efficient genetic algorithm," in *Proceedings of International Conference on Robotics and Automation*, vol. 1, Apr 1997, pp. 507–512 vol.1.

[45] C. West, A. Montazeri, S. Monk, and C. Taylor, "A genetic algorithm approach for parameter optimization of a 7dof robotic manipulator," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1261 – 1266, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896316309648

[46] P. Shiakolas, D. Koladiya, and J. Kebrle, "Optimum robot design based on task specifications using evolutionary techniques and kinematic, dynamic, and structural constraints," *Inverse Problems in Engineering*, vol. 10, no. 4, pp. 359–375, 2002.

[47] "Irb 8700 - industrial robots from abb robotics," http://new.abb.com/products/robotics/industrial-robots/irb-8700/, 2015, accessed: 2017-08-09.

[48] S. Stavridis, D. Papageorgiou, and Z. Doulgeri, "Dynamical system based robotic motion generation with obstacle avoidance," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 712–718, April 2017.

[49] J. D. Cohen, M. C. Lin, D. Manocha, and M. Ponamgi, "I-collide: An interactive and exact collision detection system for large-scale environments," in *Proceedings of the 1995 Symp. on Inter. 3D Graphics*, ser. I3D '95. New York, NY, USA: ACM, 1995, pp. 189–ff.

[50] S. Gottschalk, M. C. Lin, and D. Manocha, "Obbtree: A hierarchical structure for rapid interference detection," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 171–180. [Online]. Available: http://doi.acm.org/10.1145/237170.237244

[51] N. K. Govindaraju, S. Redon, M. C. Lin, and D. Manocha, "Cullide: Interactive collision detection between complex models in large environments using graphics hardware," in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, ser. HWWS '03. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2003, pp. 25–32. [Online]. Available: http://dl.acm.org/citation.cfm?id=844174.844178

[52] M. Tang, S. Curtis, S.-E. Yoon, and D. Manocha, "Interactive continuous collision detection between deformable models using connectivity-based culling," in *SPM '08: Proceedings of the 2008 ACM symposium on Solid and physical modeling*. New York, NY, USA: ACM, 2008, pp. 25–36.

[53] G. Zachmann, "The boxtree: Exact and fast collision detection of arbitrary polyhedra," in *SIVE Workshop*, jul 1995, pp. 104–112.

[54] G. Zachmann, "Rapid collision detection by dynamically aligned dop-trees," in *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No.98CB36180)*, 1998, pp. 90–97.

[55] D. Puiu and F. Moldoveanu, "Real-time collision avoidance for redundant manipulators," in *2011 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, May 2011, pp. 403–408.

[56] I. Lee, K. K. Lee, O. Sim, K. S. Woo, C. Buyoun, and J. H. Oh, "Collision detection system for the practical use of the humanoid robot," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov 2015, pp. 972–976.

[57] G. van den Bergen, "Efficient collision detection of complex deformable models using aabb trees," *J. Graph. Tools*, vol. 2, no. 4, pp. 1–13, Jan 1998.

[58] S. Cameron, "Collision detection by four-dimensional intersection testing," *IEEE Transactions on Robotics and Automation*, vol. 6, pp. 291–302, 1990.

[59] J. Kwon and O. Khatib, "Adaptive collision checking for continuous robot motions within motion constraints," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 5365–5372.

[60] J. Canny, "Collision detection for moving polyhedra," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 2, pp. 200–209, Feb 1986.

[61] B. Mirtich, "V-clip: Fast and robust polyhedral collision detection," *ACM Trans. Graph.*, vol. 17, no. 3, pp. 177–208, Jul 1998.

[62] T. C. Hudson, M. C. Lin, J. Cohen, S. Gottschalk, and D. Manocha, "V-collide: Accelerated collision detection for vrml," in *Proceedings of the Second Symposium on Virtual Reality Modeling Language*, ser. VRML '97. New York, NY, USA: ACM, 1997, pp. 117–ff. [Online]. Available: http://doi.acm.org/10.1145/253437.253472

[63] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, "Fast distance queries with rectangular swept sphere volumes," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 4, 2000, pp. 3719–3726 vol.4.

[64] G. V. den Bergen, "A fast and robust gjk implementation for collision detection of convex objects," *Journal of Graphics Tools*, vol. 4, no. 2, pp. 7–25, 1999. [Online]. Available: http://dx.doi.org/10.1080/10867651.1999.10487502

[65] G. van den Bergen, "Efficient collision detection of complex deformable models using aabb trees," *Journal of Graphics Tools*, vol. 2, no. 4, pp. 1–13, 1997. [Online]. Available: http://dx.doi.org/10.1080/10867651.1997.10487480

[66] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, and R. Dillmann, "Toward humanoid manipulation in human-centred environments," *Robotics and Autonomous Systems*, vol. 56, no. 1, pp. 54 – 65, 2008, human Technologies: "Know-how".

[67] S. M. Grigorescu, D. Ristić-Durrant, and A. Gräser, "Rovis: Robust machine vision for service robotic system friend," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 3574–3581.

[68] K. B. Kaldestad, S. Haddadin, R. Belder, G. Hovland, and D. A. Anisi, "Collision avoidance with potential fields based on parallel processing of 3d-point cloud data on the gpu," in *2014 IEEE Int. Conf. Robotics Autom. (ICRA)*, May 2014, pp. 3250–3257.

[69] F. Padula and V. Perdereau, "A new pseudoinverse for manipulator collision avoidance," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 14 687 – 14 692, 2011.

[70] L. Žlajpah and T. Petrič, *Obstacle Avoidance for Redundant Manipulators as Control Problem*. InTech, 2012. [Online]. Available: https://www.intechopen.com/books/serial-and-parallel-robot-manipulators-kinematics-dynamics-control-and-optimization/obstacle-avoidance-for-redundant-manipulators-as-a-control-problem

[71] M. Barbehenn, P. C. Chen, and S. Hutchinson, "An efficient hybrid planner in changing environments," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, May 1994, pp. 2755–2760 vol.4.

[72] E. Yoshida and F. Kanehiro, "Reactive robot motion using path replanning and deformation," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 5456–5462.

[73] L. L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*, ser. Advanced textbooks in control and signal processing. London, New York: Springer, 2000.

[74] G. G. Slabaugh, "Computing euler angles from a rotation matrix," *Technical Report*, 1999. [Online]. Available: http://www.staff.city.ac.uk/~sbbh653/publications/euler.pdf

[75] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.

[76] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, Mar 1985, pp. 500–505.

[77] R. Colbaugh, H. Seraji, and K. L. Glass, "Obstacle avoidance for redundant robots using configuration control," *Journal of Robotic Systems*, vol. 6, no. 6, pp. 721–744, 1989. [Online]. Available: http://dx.doi.org/10.1002/rob.4620060605

[78] K. Glass, R. Colbaugh, D. Lim, and H. Seraji, "Real-time collision avoidance for redundant manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 448–457, Jun 1995.

[79] Z. Y. Guo and T. C. Hsia, "Joint trajectory generation for redundant robots in an environment with obstacles," in *Proceedings., IEEE International Conference on Robotics and Automation*, May 1990, pp. 157–162 vol.1.

[80] R. J. Vaccaro and S. D. Hill, "A joint-space command generator for cartesian control of robotic manipulators," *IEEE Journal on Robotics and Automation*, vol. 4, no. 1, pp. 70–76, Feb 1988.

[81] M. I. C. Dede, O. W. Maaroof, and E. Tatlicioglu, "A new objective function for obstacle avoidance by redundant service robot arms," *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, p. 48, 2016. [Online]. Available: http://dx.doi.org/10.5772/62471

[82] J. O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, Jun 1992.

[83] R. Volpe and P. Khosla, "Manipulator control with superquadric artificial potential functions: theory and experiments," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 6, pp. 1423–1436, Nov 1990.

[84] A. McLean and S. Cameron, "The virtual springs method: Path planning and collision avoidance for redundant manipulators," *The International Journal of Robotics Research*, vol. 15, no. 4, pp. 300–319, 1996. [Online]. Available: http://dx.doi.org/10.1177/027836499601500401

[85] M. G. Park and M. C. Lee, "A new technique to escape local minimum in artificial potential field based path planning," *KSME International Journal*, vol. 17, no. 12, pp. 1876–1885, 2003. [Online]. Available: http://dx.doi.org/10.1007/BF02982426

[86] W. Guan, Z. Weng, and J. Zhang, "Obstacle avoidance path planning for manipulator based on variable-step artificial potential method," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*, May 2015, pp. 4325–4329.

[87] S. Byrne, W. Naeem, and R. S. Ferguson, *Efficient Local Sampling for Motion Planning of a Robotic Manipulator*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 164–175. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32527-4_15

[88] O. Khatib, "Dynamic control of manipulator in operational space," in *Proc. 6th IFToMM World Congress on Theory of Machines and Mechanisms*, 1983, pp. 1128–1131.

[89] C. Y. Chung, B. H. Lee, and J. H. Lee, "Obstacle avoidance for kinematically redundant robots using distance algorithm," in *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, vol. 3, Sep 1997, pp. 1787–1793 vol.3.

[90] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, no. 4, pp. 433–454, May 2012. [Online]. Available: http://dx.doi.org/10.1007/s10514-012-9287-y

[91] M. Saveriano and D. Lee, "Point cloud based dynamical system modulation for reactive avoidance of convex and concave obstacles," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 5380–5387.

[92] C. Chow, *An introduction to computational fluid mechanics.* John Wiley and Sons, Inc.,New York, NY, Jan 1979.

Publication I

Tuomo Kivelä, Jouni Mattila, and Jussi Puura, "A Generic Method to Optimize a Redundant Serial Robotic Manipulator's Structure", *Automation in Construction*, vol. 81, pp. 172–179, September 2017.

The original print of this publication is available online (DOI: 10.1016/j.autcon.2017.06.006).

Automation in Construction 81 (2017) 172–179

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

A generic method to optimize a redundant serial robotic manipulator's structure

Tuomo Kivelä[a,*], Jouni Mattila[a], Jussi Puura[b]

[a] Tampere University of Technology, Laboratory of Automation and Hydraulics, P.O. Box 589 (Korkeakoulunkatu 6), Tampere FIN-33101, Finland
[b] Sandvik Mining and Construction Oy, Pihtisulunkatu 9, Tampere FIN-33310, Finland

ARTICLE INFO

Keywords:
Serial manipulator
Robotic
Optimization
Synthesis
Redundant
Computer-aided design
Constraint
Workspace

ABSTRACT

In this paper, an optimization method for a redundant serial robotic manipulator's structure is proposed in order to improve their performance. Optimization was considered in terms of kinematics using the proposed objective function and the non-linear Levenberg-Marquardt algorithm for multi-variate optimization. Range limits of the joints, bounds of the design parameters, and a constrained workspace are enforced in the proposed method. A desired manipulator can be optimized to cover the required task points using dimensional synthesis. This approach effectively optimizes the link lengths of the manipulator and minimizes the position and orientation errors of the tool center point. A commercial heavy-duty hydraulic, underground tunneling manipulator was used to demonstrate the capability of the proposed optimization method. The obtained results encourage the use of the proposed optimization method in automated construction applications, such as underground tunneling, where the confined environment and the required task add challenges in the design of task-based robotic manipulators.

1. Introduction

Construction project must always battle against time and cost. One has to get the project right and get it right from the start. This requires very deep knowledge in construction but also a range of equipment that can handle the required tasks. Particularly, tunneling construction involves very demanding projects with confined workspaces. Therefore, the appropriate structural design of manipulators plays an important role in designing the optimal manipulator.

Most commonly, industrial robotic manipulators with six degree of freedom and spherical wrists are designed to achieve the desired payload while maximizing the robot's workspace envelope. In more advanced design scenarios, the robot's total orientation workspace can be specified as the range of the end-effector's rotation angles inside a bounded workspace [1]. This type of design can be classified as general purpose design. Although using a general purpose design is generally acceptable, it does not guarantee the optimal design for task execution. Therefore, task-based optimization is preferred when a task is predefined.

The task-based optimization or synthesis of a robotic manipulator consists of finding a set of manipulator design parameters so that the required task points and kinematic requirements are fulfilled. The kinematic requirements, such as a constrained workspace, design parameter limits and joint limits, are task specifications that affect the kinematic structure of the robotic manipulator.

The most important requirement of the manipulator is the ability to reach to the desired task points. For example, in tunnel construction, the purpose of a drill plan is to describe the set descriptive poses that must be reached. This may not be used by the customer in a single drilling phase. Instead, it may contain selected drill plans combined into a single set of task points. Other priorities include obstacle avoidance, singularity avoidance, manipulator dexterity and several other objectives that affect the kinematic design of manipulators directly or indirectly.

Comprehensive studies have been conducted on the kinematic synthesis of serial robotic manipulators [2–11]. A classic approach to task-based dimensional synthesis is to create objective and constraint functions and then sum them to form a cost function, which can be minimized to find an optimal solution [2–4]. Constraint functions can be weighted to address the solution in the specified direction. Although previous studies have shown how different optimization approaches can be used to solve task-based dimensional synthesis problems, they are limited and solve only a part of our optimization problem. For example, [5–7] focus on non-redundant manipulators, and some works concentrate the optimization of serial manipulators with only three joints [4,6,8]. Furthermore, some works are restricted and optimize the

* Corresponding author.
E-mail addresses: tuomo.kivela@tut.fi (T. Kivelä), jouni.mattila@tut.fi (J. Mattila), jussi.puura@sandvik.com (J. Puura).

http://dx.doi.org/10.1016/j.autcon.2017.06.006
Received 22 September 2016; Received in revised form 15 May 2017; Accepted 10 June 2017
Available online 20 June 2017

manipulator's structure based on end-effector's position rather than on both position and orientation [4,9]. Usually, there is no explanation of where the task points came from. Further, the number of task points is often limited to only a few points [2,5,9]. As a result, no comprehensive real-world scale study appears to exist.

In spite of previous studies, the task-based optimization of an arbitrary manipulator's structures with practical relevance has yet to be achieved. Hence, additional studies of the task-based optimization of arbitrary manipulators to find the appropriate values for the design parameters, the base frame, and the joint values are needed. Typically, this type of optimization problem required several hundreds of parameters to be optimized. Therefore, the aim of this paper is to extend the aforementioned works to cover more general serial manipulators optimization with practical relevance for construction manipulators. The proposed optimization method can handle non-redundant and redundant manipulators. Furthermore, it can handle both revolute and prismatic joints and n-dimensional task spaces as well as end effector position and orientation. In addition, the proposed method can be used to optimize hundreds of parameters efficiently, and the number of parameters to be optimized is not restricted by the proposed method.

In this paper, we propose an optimization method that takes into account not only the desired task points but also practical design constraints for given task points, including most important constraints, such as the available confined working envelope and joint ranges. The topology of the tunnel construction manipulator used in this case study is proven by construction customers to be suitable for tunnel construction. Therefore, a change in the manipulator's topology was not necessary in this study. Instead, the dimensions of the existing manipulator were optimized to enhance the performance and profitability of the manipulator.

Studied tunnel drilling patterns consist of multiple task points where several real drill plans are combined into a single set of task points. This real customer application requires a long-reach manipulator (10–15 m) due to the desired tunnel size. Therefore, a manipulator with at least one prismatic joint is of interest. Our results indicate that this method is effective in optimizing the manipulator's dimensions compared to the original manipulator's design according to the selected performance measures.

We have organized the rest of this paper as follows. In Section 2, the kinematic synthesis process is described according to the task specification. In Section 3, the dimensional synthesis method based on selectively damped least squares is described with optimization constraints and initial design parameters. The proposed kinematic synthesis method was applied to a case study; the results are given in Section 4, and the conclusions are given in Section 5.

2. Kinematic synthesis

In general, kinematic synthesis can be divided into two separate types: topology synthesis and dimensional synthesis. Topology synthesis involves determining the structure of a device, whereas dimensional synthesis is devoted to determining the dimensions of a device. The synthesis process begins with defining the task that the manipulator's structure should be able to fulfill. This step is preferably performed with the customer to maximize the benefit the machine will bring him or her. For example, if the machine must be carefully repositioned before starting a new work cycle, it may be desirable to reach a certain number of goal points before repositioning is needed. In general, the task consists of several poses that need to be reached by the manipulator together with a tool attached to it. Each of these task points can be expressed with a homogeneous transformation matrix, which is described as follows:

$$\mathbf{T}_{tcp}^{w} = \begin{bmatrix} \mathbf{n}_{tcp}^{w} & \mathbf{s}_{tcp}^{w} & \mathbf{a}_{tcp}^{w} & \mathbf{p}_{tcp}^{w} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where $\mathbf{n}_{tcp}$, $\mathbf{s}_{tcp}$, and $\mathbf{a}_{tcp}$ are the unit vectors of a frame attached to the end effector and $\mathbf{p}_{tcp}$ is the position vector of the origin of the frame with respect to the origin of the world frame. In addition to locations and orientations given with the required degrees of freedom, tasks may also include other required attributes, such as a required force output or structural stiffness.

Usually, the number and type of joints are given as input parameters for kinematic synthesis. In some methods, the angles between two consecutive links can be design parameters, and the optimization method decides these angles. For example, Singla et al. [2] optimized redundant serial manipulators for cluttered environments, and Ouezdou et al. [11] optimized manipulators with task specifications. They defined the number of joints, type and order of the joints, and then used the optimization process to find the optimal link length and joint locations. This type of optimization is preferred for cluttered environments where it is mandatory to use complicated snake-like manipulators. For example, in [2], 8–10 joints were needed to achieve satisfactory results. In applications where the environment is not cluttered, it is preferable to keep the design as simple as possible and the number of joints as small as possible. With twist angles fixed between two consecutive links, it is possible to optimize simple structures, which makes it easier to produce than complex structures. If the joints of the manipulator have to be hydraulic due to high torque requirements, the most desirable hydraulic joint is driven by a hydraulic cylinder, which is more cost-effective and lighter in weight than hydraulic motors. Therefore, it is profitable to use hydraulic cylinders whenever possible. Hydraulic cylinders are much easier to implement if the twist angle between the joints is multiple with 90 °.

2.1. Topology synthesis

Topology synthesis involves determining the structure of a device, excluding link lengths, from the task description. The word *structure* is used to describe the number, type, and arrangement of joints and the links connected by these joints. Ideally, topology synthesis results in an optimal topology for the task, and this topology is the subjected to dimensional synthesis to determine link lengths (see Section 2.2).

There is no general approach for the topology synthesis of open-chain serial manipulators. Graph theory [12] is widely used in research involving topology synthesis, but most of these studies are related to closed-chain kinematic structures [13,14]. For open-chain kinematic structures, a logical approach using existing knowledge is appropriate for topology synthesis.

2.2. Dimensional synthesis

A classic approach for dimensional synthesis begins with assigning weight factors for several criteria and then totalling them to form a cost function, which can be then minimized to find an optimal solution [3,4]. The multi-variable, multi-constraint and multi-objective aspects of the problem are further discussed in [10]. Other methods include stochastic algorithms, distributed optimization techniques [11] and parameter space approaches, which are used for the parallel Gough platform in [15].

The augmented Lagrangian method is a constrained optimization method [2]. A pure penalty function method penalizes an objective function in order to discourage constraint violation, typically using a large penalty parameter. This causes a poor rate of convergence, since the Hessian of the Lagrangian becomes ill-conditioned. In the augmented Lagrangian method, the Lagrangian is combined with a modest penalty term. In [2], the augmented Lagrangian method was used, and it proved to be efficient and robust for the synthesis of serial redundant manipulators with obstacle avoidance. In that research, the authors achieved synthesis with 6-, 8-, and 10-joint manipulators.

The genetic algorithm method can be used to optimize a multi-objective cost function that contains many local minimum points. Barissi

et al. presented a multi-objective cost function that elaborates different constraints as well as an optimality criterion for the design of serial robotic manipulators [4]. They showed that the inclusiveness and flexibility of the proposed method make it suitable for the geometric design optimization of robotic manipulators.

Collisions between the manipulator and environmental obstacles, as well as manipulator self-collision, can be prevented by taking these collisions into account in the dimensional synthesis algorithm. One way to avoid obstacles is to create obstacle avoidance points on the manipulator and simulate forces that repel these points away from obstacles if any are found in the vicinity [16]. Another method to check for collisions is to use geometric-based computer-aided design models and then prevent them.

Dimensional synthesis is a nonlinear parameter estimation problem, and it can therefore be realized as a kinematic calibration problem [17]. Kinematic calibration techniques are devoted to finding accurate estimates of the Denavit-Hartenberg (DH) parameters for the robotic manipulator. There are several different methods to carry out kinematic calibration (dimensional synthesis); selectively damped least squares (SDLS) is one such method [18]. This method is based on calculating the inverse matrix of the Jacobian of the transformation between the parameter space and the operational space. If the manipulator is redundant, the direct inverse of the Jacobian is not possible, and a pseudo-inverse solution must be used instead of a direct inverse solution, which is described in the following section. The parameter estimates can be calculated by multiplying the inverse matrix with the pose error. Since this method is an iterative method, the procedure is iterated until the change in parameter values converges within a given threshold. At each iteration, the inverse matrix is updated with the parameter estimates and pose errors through direct kinematics. An unweighted inverse solution gives the best possible solution to the pose error in the sense of least squares. A different way to realize desired performance characteristics is using a weighted inverse. An appropriately chosen weighting matrix can be used to optimize the solution towards the desired properties. The dimensional synthesis method proposed in this paper is based on SDLS.

3. Dimensional synthesis method

A robot manipulator consists of links connected by joints. Joints are essentially of two types: revolute or prismatic. The whole manipulator's structure forms a kinematic chain. The kinematic chain can be an open chain, a closed chain or a combination of both. One end of the kinematic chain is constrained to a base, and the other end is connected to an end effector, which allows for the manipulation of objects in space.

The mechanical structure of a manipulator can be described by its degrees of mobility (DOM), which uniquely determine its configuration. Each DOM has a joint variable. Direct kinematics are used to compute the position and orientation of the end effector of the manipulator as a function of the joint variables.

The DH convention is a well-known way to describe the kinematic and geometric structures of a robotic manipulator. This method defines the relative positions and orientations of two consecutive links. There are two different ways to represent DH parameters: classic DH parameters [17] and modified DH parameters [19]. The difference between these two representations is the location of the coordinate system attachment to the links and the order of the performed transformations. With modified DH convention, transformations between two frames can be described by four parameters (Fig. 1):

α_{i-1} angle between z_{i-1} and z_i, about x_{i-1}
a_{i-1} distance on the common normal of axis $i-1$ and axis i along x_{i-1}
θ_i angle between x_{i-1} and x_i, about z_i
d_i distance from the common normal to point O_i along z_i

Using this notation, the coordinate transformation between two frames can be described by a transformation matrix as follows:

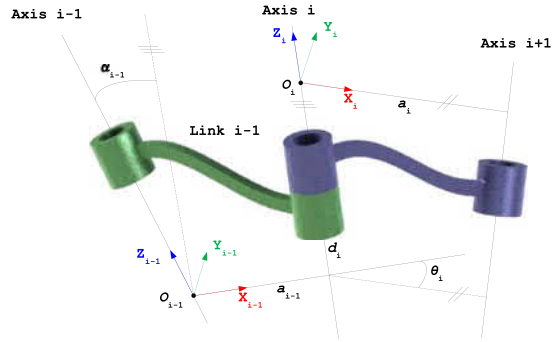


Fig. 1. Coordinate frame locations of the modified Denavit-Hartenberg parameter notation.

$$\mathbf{T}_i^{i-1} = Rot_{x_{i-1}}(\alpha_{i-1})\cdot Trans_{x_{i-1}}(a_{i-1})\cdot Rot_{z_i}(\theta_i)\cdot Trans_{z_i}(d_i), \tag{2}$$

and the transformation matrix is transformed into the following equation:

$$\mathbf{T}_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where s and c denote sin and cos functions, respectively. Then, the coordinate transformation describing the position and orientation of frame n with respect to frame 0 is given as follows:

$$\mathbf{T}_n^0(\mathbf{q}) = \mathbf{T}_1^0(q_1)\mathbf{T}_2^1(q_2)...\mathbf{T}_n^{n-1}(q_n). \tag{4}$$

Combining the position and orientation of an end effector describes a manipulator pose as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \boldsymbol{\phi} \end{bmatrix}, \tag{5}$$

where $\mathbf{p}$ describes the end effector position and $\boldsymbol{\phi}$ describes its orientation. Now, the relationship between the joint value vector $\mathbf{q} \in \mathbb{R}^n$ and the end effector pose vector $\mathbf{x} \in \mathbb{R}^m$ is given as follows:

$$\mathbf{x} = f(\mathbf{q}), \tag{6}$$

where $f(\cdot)$ is the non-linear direct kinematic function of the manipulator. The geometric Jacobian matrix $\mathbf{J}_G$ relates the joint velocity vector $\dot{\mathbf{q}}$ to the end effector velocity vector $\dot{\mathbf{x}}$ through mapping the following:

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{pmatrix} = \mathbf{J}_G(\mathbf{q})\dot{\mathbf{q}}, \tag{7}$$

where $\dot{\mathbf{p}}$ and $\boldsymbol{\omega}$ represent the end effector's linear and angular velocities, respectively. This allows us to solve the inverse kinematic function by inverting the Jacobian matrix as follows:

$$\dot{\mathbf{q}} = \mathbf{J}_G^{-1}(\mathbf{q})\dot{\mathbf{x}}. \tag{8}$$

Eq. (8) is valid only for manipulators that have the same number of DOM (n) and operational variables necessary to specify a given task (r), that is, manipulators in which $n = r$. When the manipulator is redundant ($r < n$) the Jacobian matrix has more columns than rows, and an infinite number of solutions exist for Eq. (7). Therefore, a direct inverse of the Jacobian matrix cannot be achieved. If the Jacobian matrix is a non-square matrix, then a suitable pseudo-inverse can be adopted as follows [17]

$$\dot{\mathbf{q}} = \mathbf{J}_G^{\dagger}(\mathbf{q})\dot{\mathbf{x}}, \tag{9}$$

and the following matrix

$$\mathbf{J}_G^{\dagger} = \mathbf{J}_G^T(\mathbf{J}_G\mathbf{J}_G^T)^{-1} \tag{10}$$

is the right pseudo-inverse of $\mathbf{J}_G$ ($\mathbf{q}$).

The pseudo-inverse method sets the value of $\dot{\mathbf{q}}$ to be equal to $\mathbf{J}_G^{\dagger}$ (Eq. (9)). This pseudo-inverse method gives the best possible solution to Eq. (7) in terms of least squares. Unfortunately, the pseudo-inverse tends to have stability problems in the neighborhood of singularities. At a singularity, the Jacobian matrix no longer has a full row rank. If the configuration is close to a singularity, then the pseudo-inverse method causes very large changes in the joint angles, even for small movements in the target position.

The damped least-squares method prevents many of the pseudo-inverse method's problems with singularities and has a numerically stable method of selecting $\dot{\mathbf{q}}$. Rather than just finding the minimum vector $\dot{\mathbf{q}}$ that gives the best solution, the damped least-squares method finds the value of $\dot{\mathbf{q}}$ that minimizes the quan tity of the following equation:

$$\|\mathbf{J}_G\dot{\mathbf{q}}-\dot{\mathbf{x}}\| + \lambda^2 \|\dot{\mathbf{q}}\|^2, \tag{11}$$

where λ is a non-zero damping constant. Now, the damped least-squares inverse kinematics solution is equal to the following:

$$\dot{\mathbf{q}} = \mathbf{J}_G^T(\mathbf{J}_G\mathbf{J}_G^T+\lambda^2\mathbf{I})^{-1}\dot{\mathbf{x}}. \tag{12}$$

The damping constant depends on the details of the multi-body and the target positions and must be chosen carefully to make Eq. (12) numerically stable. The damping constant should be large enough so that the solutions for $\dot{\mathbf{q}}$ are well-behaved near singularities, but, if it is too large, then the convergence rate is slow. A number of methods have been proposed to select the damping constant dynamically; these methods are called SDLS methods [18]. The damping constant of SDLS methods depends on the configuration of the manipulator, the relative pose of the end effector and the task pose.

The singular value decomposition (SVD) [20] is an useful method for analyzing the damped least-squares method. This method can be also used to select a damping coefficient for the SDLS method. According to SVD theory, in some $\mathbf{J}$ ($m \times n$) orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ of dimensions $m \times n$ and $n \times n$ respectively exist, so that:

$$\mathbf{J} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{13}$$

where $\mathbf{D}$ is the $n \times n$ diagonal matrix formed by singular values of $\mathbf{J}$, which are arranged in descending order, that is, $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_m \geq 0$.

Damping the pseudo-inverse method near a singular region is necessary. Therefore, the damping factor can be defined so that damping is applied only when entering a singular region. The singular region can be defined on the basis of the estimate of the smallest singular value of a Jacobian matrix using SVD. The damping factor can be defined based on smallest singular value, as in [21]:

$$\lambda^2 = \begin{cases} 0, & \text{if} \quad \sigma_m \geq \epsilon \\ \left(1-\left(\frac{\sigma_m}{\epsilon}\right)^2\right)\lambda_{\max}^2, & \text{if} \quad \sigma_m < \epsilon, \end{cases} \tag{14}$$

where σ_m is the smallest singular value of the Jacobian matrix, ϵ defines the width of the singular region in which the damping factor gets a non-zero value, and $\lambda_{\max}$ is the maximum damping factor value.

3.1. Optimization process

The direct kinematic equation in Eq. (6) can be rewritten so that all fixed DH parameters are emphasized as follows:

$$\mathbf{x} = f(\mathbf{a}, \mathbf{d}, \boldsymbol{\alpha}, \boldsymbol{\theta}), \tag{15}$$

where $\mathbf{a} = [a_1 \ldots a_n]^T$, $\mathbf{d} = [d_1 \ldots d_n]^T$, $\boldsymbol{\alpha} = [\alpha_1 \ldots \alpha_n]^T$ and $\boldsymbol{\theta} = [\theta_1 \ldots \theta_n]^T$. These are the DH parameters for the whole structure.

Let $\mathbf{x}_t$ be the pose of the end effector defined by a task, and let $\mathbf{x}_c$ be the current pose that can be computed with the DH parameters. The deviation $\Delta\mathbf{x} = \mathbf{x}_t - \mathbf{x}_c$ gives a measure of accuracy at the given pose. On the assumption of small deviations in the first approximation, it is possible to derive the following:

$$\Delta\mathbf{x} = \frac{\partial f}{\partial \mathbf{a}}\Delta\mathbf{a}\frac{\partial f}{\partial \mathbf{d}}\Delta\mathbf{d}\frac{\partial f}{\partial \boldsymbol{\alpha}}\Delta\boldsymbol{\alpha}\frac{\partial f}{\partial \boldsymbol{\theta}}\Delta\boldsymbol{\theta}. \tag{16}$$

Eq. (16) can be expressed in a more compact form as follows:

$$\Delta\mathbf{x} = \boldsymbol{\Phi}\Delta\boldsymbol{\zeta}, \tag{17}$$

where

$$\boldsymbol{\Phi} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{a}} & \frac{\partial \mathbf{f}}{\partial \mathbf{d}} & \frac{\partial \mathbf{f}}{\partial \boldsymbol{\alpha}} & \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} \end{bmatrix} \tag{18}$$

$\boldsymbol{\Phi} \in \mathbb{R}^{m \times 4n}$ and m is number of operational space variables. The DH parameters can be grouped into the following vector:

$$\boldsymbol{\zeta} = [\mathbf{a}^T \quad \mathbf{d}^T \quad \boldsymbol{\alpha}^T \quad \boldsymbol{\theta}^T], \tag{19}$$

where $\boldsymbol{\zeta} \in \mathbb{R}^{4n \times 1}$. Then, the parameter variations, with respect to the current parameter values, can be expressed as follows:

$$\Delta\boldsymbol{\zeta} = \boldsymbol{\zeta}_t - \boldsymbol{\zeta}_c, \tag{20}$$

where $\boldsymbol{\zeta}_c$ denotes the current parameters and $\boldsymbol{\zeta}_t$ denotes the parameters required by the task.

When the number of task locations is k, it yields the following:

$$\Delta\overline{\mathbf{x}} = \begin{bmatrix} \Delta\mathbf{x}_1 \\ \vdots \\ \Delta\mathbf{x}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_k \end{bmatrix}\Delta\boldsymbol{\zeta} = \overline{\boldsymbol{\Phi}}\Delta\boldsymbol{\zeta}. \tag{21}$$

The solution for $\Delta\boldsymbol{\zeta}$ can be found using Eq. (12)

$$\Delta\boldsymbol{\zeta} = \overline{\boldsymbol{\Phi}}^T(\overline{\boldsymbol{\Phi}}\overline{\boldsymbol{\Phi}}^T+\lambda^2\mathbf{I})^{-1}\Delta\overline{\mathbf{x}}. \tag{22}$$

Since this method is an iterative method, the procedure is iterated until the change in parameter values converges within a given threshold. At each iteration, the pseudo-inverse matrix is to be updated with the parameter estimates ($\boldsymbol{\zeta}_c = \boldsymbol{\zeta}_c + \Delta\boldsymbol{\zeta}$), as well as pose errors using direct kinematics.

3.2. Optimization constraints

In kinematic optimization problems, there are usually some kinematic constraints that need to be taken into account during the optimization process. Kinematic constraints can include limits, parameter limits, or obstacle avoidance. These constraints have an impact on the optimization results.

The pseudo-inverse has a self-motion property that can be used to consider optimization constraints. The matrix $(\mathbf{I}-\mathbf{J}_G^{\dagger}\mathbf{J}_G)$ projects onto the null-space of $\mathbf{J}_G$. The pseudo-inverse solution in Eq. (9) can be re-written as follows [22]:

$$\dot{\mathbf{q}} = \mathbf{J}_G^{\dagger}\dot{\mathbf{x}} + (\mathbf{I}-\mathbf{J}_G^{\dagger}\mathbf{J}_G)\dot{\mathbf{q}}_0, \tag{23}$$

where $\dot{\mathbf{q}}_0$ is the arbitrary joint-space velocity. The null-space projection ensures that, for all vectors, $\dot{\mathbf{q}}_0$:

$$\mathbf{J}_G(\mathbf{I}-\mathbf{J}_G^{\dagger}\mathbf{J}_G)\dot{\mathbf{q}}_0 = \mathbf{0}. \tag{24}$$

By selecting a suitable joint-space velocity, one can manipulate internal motions of the manipulator and therefore prevent constraint violations.

3.3. Initial joint and design parameter values

Finding a solution to a Newton-based optimization method depends, in part, on the initial values given. It is therefore very important to choose the initial values as accurately as possible. Poorly selected initial values may cause the problem to not converge and place it in a local minimum in which the problem cannot be resolved. In this paper, we used a method based on random numbers to select the initial values. First, we specified the number of design parameter sets based on beta distribution. Thereafter, joint values for each set of parameters were

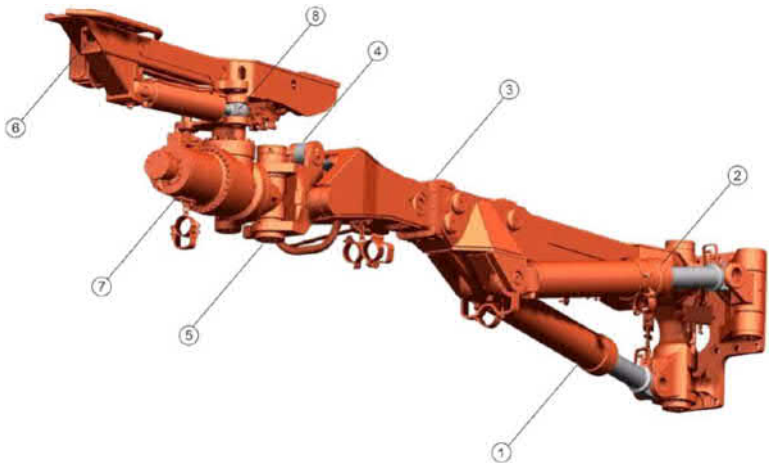


Fig. 2. Sandvik SB60 hydraulic rock-drilling boom used in underground mining jumbos. Boom movements: (1) boom lift, (2) boom swing, (3) boom extension, (4) feed tilt, (5) feed swing, (6) feed extension, (7) feed rotation, and (8) feed tilt for bolting.

solved using inverse kinematics for each task point. The parameter set closest to each task point was selected as an initial value.

4. Case study of manipulator optimization

The proposed optimization algorithm was used to optimize the serial redundant manipulators. To verify that the developed algorithm can optimize a manipulator's structure, the algorithm was tested with an eight DOM redundant serial manipulator, the Sandvik SB60 hydraulic manipulator. The SB60 manipulator is a hydraulic universal roll-over manipulator for mechanized tunneling, drifting, cross-cutting and bolt hole drilling. It can also be used for production drilling using cut-and-fill and room-and-pillar mining methods. The SB60 manipulator has six revolute joints and two prismatic joints (Fig. 2).

Real-world underground tunneling drilling patterns for holes was used to find the optimal structure for a manipulator to drill these holes. The purpose of a single drill plan (task point space) is to describe the poses that must be reached. The objective was to optimize the manipulator's structure by modifying the manipulator link lengths as well as the position of the coordinate system of the manipulator base in order to satisfy a number of design constraints. The drilling pattern consisted of 87 task points (n_{tp}); 10 real drill plans were chosen from actual customer cases and combined into a single set of task points. The aim was to reach all the positions with the desired orientation of the tool. Six ($n_r = 6$) operational space variables were considered: three variables for the positions of the drilling holes and three variables for the orientations of the drilling holes. The manipulator included seven design parameters (n_{dp}) (see Fig. 3 and Table 1) to be optimized that were changed during optimization. In addition, the manipulator's base frame (n_{bf}) is freely movable. Furthermore, the manipulator has eight joints, seven (n_{fj}) of which can be moved during the optimization process. The joint number seven (end effector tilt) was fixed to a constant joint value. The optimization task was to find the appropriate values for the design parameters, the base frame and the joint values. Thus, several hundreds of ($n_{np} = n_{dp} + n_{bf} + n_{fj} * n_{tp}$) parameters ($\zeta_o \in \mathbb{R}^{n_{np}}$) in total were optimized with the proposed method.

The process flow to optimize the serial manipulator's structure is shown in Table 2. Step 4 was repeated until the change in parameter values converged within a given threshold.

4.1. Joint and design parameter constraints

Several design constraints limit the manipulator's structure, and these constraints must be taken considered in the optimization process.

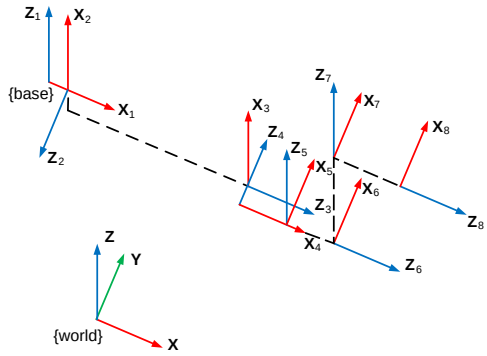


Fig. 3. Coordinate frame locations based on a modified Denavit-Hartenberg convention.

Table 1
Denavit-Hartenberg parameters for the manipulator used in the example. The joint variables are bolded. Parameter d_3 consists of an offset value d_{3p} and a joint value $\mathbf{d}_{3j}$.

Frame	a_{i-1}	d_i	α_{i-1}	θ_i
1	0	0	0	$\boldsymbol{\theta}_1$
2	a_1	0	$\pi/2$	$\boldsymbol{\theta}_2 + \pi/2$
3	a_2	$d_{3p} + \mathbf{d}_{3j}$	$\pi/2$	0
4	0	d_4	$\pi/2$	$\boldsymbol{\theta}_4 + \pi/2$
5	a_4	0	$\pi/2$	$\boldsymbol{\theta}_5 + \pi/2$
6	0	d_6	$\pi/2$	$\boldsymbol{\theta}_6$
7[1]	0	d_7	$-\pi/2$	θ_7
8	0	$\mathbf{d}_8$	$\pi/2$	0

[1] This joint is kept fixed during the optimization.

Table 2
Dimensional synthesis algorithm.

Dimensional synthesis
1. Input: task poses, manipulator's structure
2. Parameters to be optimized ← the DH parameters that can be optimized are given in Table 1
3. Initial parameter values ← setting initial parameter values is described in Section 3.3
4. ζ ← manipulator's structure optimization
4.1 The objective function is given in Eq. (32) with the constraints in Eqs. (25) and (31)
5. LCI ← the local conditioning index described in Eq. (35).

The first constraints considered were joint constraints. In practice, all joints are bound within their limits. The reason for joint limits include mechanical design limitations and cost constraints. The second constraints considered were parameter limits. This constraint type is very similar to joint limit constraints and is handled similarly. Several studies have investigated joint limit constraints [23,24]. These researchers used performance criteria to prevent joint limits. In their work, they proposed the performance criterion ($\mathbf{H}(\boldsymbol{\theta})$) to limit joint movement outside the center position. This performance criterion provides heavier weight to the joints near their limits and exceeds infinity at the joint bounds. This performance criterion is not appropriate for our example because the ranges of each parameter and joint must be freely usable. Therefore, a slightly different performance criterion was used in this example. The proposed performance criterion remains zero within the ranges of the parameters and rises rapidly outside the bounds. This ensures that the parameter's whole range is always usable. The performance criterion is taken into account using the gradient projection method, which is shown in Eq. (23):

$$\dot{\boldsymbol{\zeta}}_{\mathbf{o}} = \mathbf{J}_G^{\dagger}\dot{\mathbf{x}} + k_1(\mathbf{I}-\mathbf{J}_G^{\dagger}\mathbf{J}_G)\nabla\mathbf{H}, \tag{25}$$

where k_1 is a real scalar coefficient. Using the gradient projection method to prevent parameters limits and resulting problems is discussed in [24]. The performance criterion used in this example is as follows:

$$H_i(\theta) = \begin{cases} \zeta_{o,i}^{\max} - \zeta_{o,i}, & \text{if} \quad \zeta_{o,i} > \zeta_{o,i}^{\max} \\ \zeta_{o,i}^{\min} - \zeta_{o,i}, & \text{if} \quad \zeta_{o,i} < \zeta_{o,i}^{\min} \\ 0, & \text{otherwise,} \end{cases} \tag{26}$$

where $\boldsymbol{\zeta}_o^{\max}$ and $\boldsymbol{\zeta}^{\min}$ are upper and lower bound vectors, respectively. $i = 1 \ldots np$ and $H_i(\boldsymbol{\theta})$ is ith component of $\mathbf{H}(\boldsymbol{\theta})$.

4.2. Workspace constraints

The third constraints considered were workspace constraints. For example, in this case study, the manipulator was located in an underground tunnel, which means that there were walls, a floor and a ceiling around the manipulator which is unable to go through or collide with these workspace constraints. The null-space motion concept (Section 3.2) is used to avoid manipulator collisions within a workspace. The workspace can be handled as an external object, and collisions of the manipulator's internal structure with external objects are represented. An artificial potential field is used to create a repulsive force between the manipulator and obstacles that pushes the manipulator away from the obstacles. The method of using an artificial potential field to prevent collisions ha been used in several studies [25,26].

Let $\mathbf{x}_d$ be the minimum distance between the manipulator and the obstacle. Thus, the desired velocity vector [25] in a pure velocity servo-control scheme is as follows:

$$\dot{\mathbf{x}}_d = k_2\mathbf{x}_d, \tag{27}$$

where k_2 is a real scalar velocity coefficient. $\mathbf{x}_d$ can be considered as a direct kinematics equation up until the collision point as follows:

$$\mathbf{x}_d = \mathbf{x}_o - f_d(\boldsymbol{\zeta}), \tag{28}$$

where $f_d(\cdot)$ is continuous nonlinear function from the manipulator base coordinate system up to the collision point and $\mathbf{x}_o$ is the point on the obstacle. This equation can be solved like Eq. (7), which leads to the following:

$$\dot{\mathbf{x}}_d = \mathbf{J}_d(\boldsymbol{\zeta})\dot{\boldsymbol{\zeta}}, \tag{29}$$

where $\mathbf{J}_d(\boldsymbol{\zeta})$ is workspace constraint Jacobian matrix. The joint velocity vector can be solved by a pseudo-inversion as follows:

$$\dot{\boldsymbol{\zeta}}_{\mathbf{0}} = \mathbf{J}_d^{\dagger}(\boldsymbol{\zeta})\dot{\mathbf{x}}_d, \tag{30}$$

where $\dot{\boldsymbol{\zeta}}_0$ is the joint-space velocity vector for the internal motion of the manipulator. Eq. (23) can be rewritten to prevent manipulator collisions with workspace collisions as follows:

$$\dot{\boldsymbol{\zeta}}_{\mathbf{o}} = \mathbf{J}_G^{\dagger}\dot{\mathbf{x}} + (\mathbf{I}-\mathbf{J}_G^{\dagger}\mathbf{J}_G)\mathbf{J}_d^{\dagger}\dot{\mathbf{x}}_d. \tag{31}$$

4.3. Parameter update function

In the previous sections, the optimality criterion and constraints were formulated separately. In order to optimize a given manipulator's structure, a final parameter update function must be created. This can be done by adding the optimality criterion and the different constraints together. Therefore, combining Eqs. (22), (26) and (31), the final parameter update function is as follows:

$$\dot{\boldsymbol{\zeta}}_{\mathbf{o}} = \mathbf{J}_G^{\dagger}\dot{\mathbf{x}} + k_1(\mathbf{I}-\mathbf{J}_G^{\dagger}\mathbf{J}_G)\nabla\mathbf{H} + (\mathbf{I}-\mathbf{J}_G^{\dagger}\mathbf{J}_G)\mathbf{J}_d^{\dagger}\dot{\mathbf{x}}_d. \tag{32}$$

4.4. Results

As stated in previous sections, the accessibility of task points is considered to be an objective function for optimization. This is measured as the total cumulative error, which includes position and orientation errors. Therefore, the optimization's objective is to minimize the objective function. When the objective function value is zero, all task points were reached perfectly. The objective function is given as follows:

$$e_{obj} = \sum_{i=1}^{n_{tp}} \|\mathbf{x}_t - \mathbf{x}_c\|, \tag{33}$$

where n_{tp} is the number of task points.

Fig. 4 shows the case study's manipulator configurations during the optimization process. As shown in the figure, all joint values were set to zero, and the initial design parameter values were set to be equivalent to a real manipulator. Fig. 4 shows the manipulator configurations after a few iteration steps and how the manipulator reach towards corresponding task locations. The final manipulator configurations for every second task point are shown in Fig. 5. Green indicates that the task pose was achieved. Table 3 shows the initial design parameters and optimized design parameter values from one run. These values were normalized according to each parameter's upper and lower bound limits. Table 3 shows also how much each design parameter changed during optimization.

No optimization method is guaranteed to always find the global minimum. It is therefore important to test optimization methods multiple times with different initial values to verify the results. Table 4

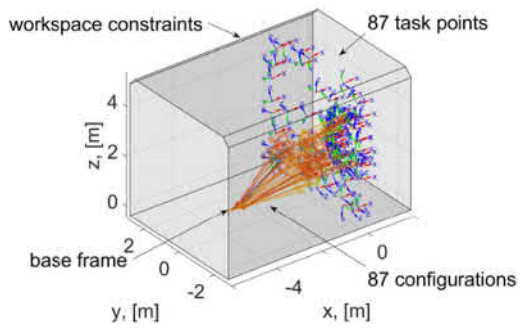


Fig. 4. Manipulator configurations with different joint values after a few iteration steps. The manipulator's state is described with different colors: outside tolerances with red (–), position tolerance satisfied with blue (–), rotation tolerance satisfied with yellow (–) and pose tolerance satisfied with green (–). To make the figure clearer, only every second task point and manipulator are shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

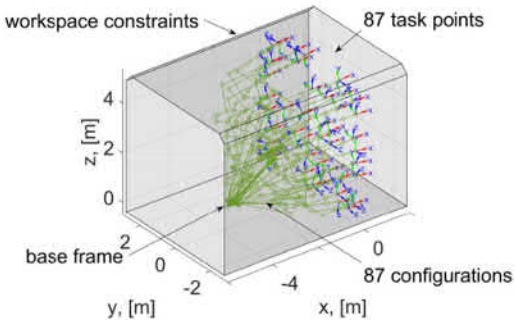


Fig. 5. Final manipulator configurations for each task point. The green color (–) indicates that the desired task point pose was fully reached. To make the figure clearer, only every second task point and manipulator are shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Initial and optimized design parameters. The values of the initial and optimized design parameters were normalized between the parameters' bounds.

Parameter	Initial value [0…1]	Optimized value [0…1]	Value changed [%]
a_1	0.3400	0.8207	48.07
a_2	0.7100	0.6376	−7.24
d_3	0.6600	0.7456	8.56
d_4	0.6250	0.9929	36.80
a_4	0.4000	0.3454	−5.46
d_6	0.7400	0.4287	−31.13
d_7	0.6400	0.6581	1.81
$base_x$	0.7500	0.3857	−36.43
$base_y$	0.5000	0.3167	−18.33
$base_z$	0.6667	0.7004	3.38

Table 4
Optimization results. The average values and standard deviations of the design parameters.

Parameter	Average [0…1]	stdDev
a_1	0.7539	0.0909
a_2	0.6900	0.2135
d_3	0.6581	0.2464
d_4	0.7637	0.2652
a_4	0.6289	0.2848
d_6	0.2029	0.2100
d_7	0.6234	0.0898
$base_x$	0.3216	0.1909
$base_y$	0.0922	0.1156
$base_z$	0.5903	0.1730

shows the mean and standard deviations of 10 different optimization runs. The initial values for these runs were generated as described in Section 3.3. In each case, the optimization method finds the minimum, and the solutions are similar.

4.5. Performance measures

Performance measures are required to describe the behavior of a manipulator. There are number of performance measures that have been studied since the early days of robotics [27]. All performance indices can be categorized based on their scope, performance characteristic or application. The performance measure that should be used to compare manipulators depends on the application. For example, for the optimized manipulator in this paper, it was not useful to use performance index that measures a workspace of the manipulator, since the manipulator was optimized to fulfill a specific task.

The condition number is a local kinematic conditioning index used to measure the degree of ill-conditioning of the manipulator or the kinematic isotropy of the Jacobian [28]. The condition number is defined as the ratio of maximum and minimum singular values of the Jacobian [29]. The condition number is calculated as follows:

$$\kappa = \frac{\sigma_{\max}}{\sigma_{\min}}. \tag{34}$$

The condition number does not have an upper bound, $\kappa \in [1, \infty]$. Therefore, its inverse, known as the local conditioning index (LCI) is more commonly used. The LCI is bounded, $LCI = [0,1]$. The LCI is calculated as follows:

$$LCI = \frac{1}{\kappa}. \tag{35}$$

When the Jacobian is non-homogeneous due to the different units used to represent the link lengths and joint angles, the condition number does not accurately represent the ill-conditioning of the manipulator's Jacobian [27]. Many scaling techniques have been used to treat the non-homogeneity of the Jacobian [30]. In this paper, the method proposed by [31] was used to scale the Jacobian matrix. The scaling method is based on the nominal link (l_{NL}), whose length is defined as the distance from the base frame to tool center point. The scaled form of the Jacobian used to calculate the LCI is as follows:

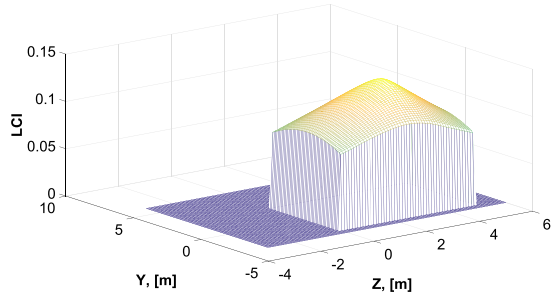
$$\tilde{\mathbf{J}}_G = \mathbf{S}\mathbf{J}_G, \tag{36}$$

where

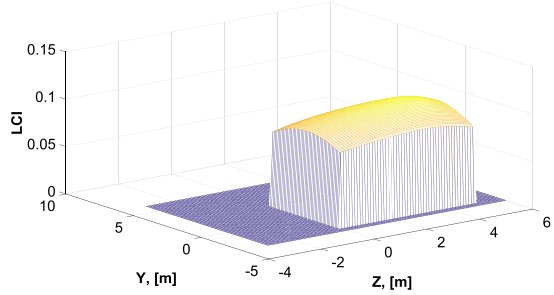
$$\mathbf{S} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{l}_{NL} \end{bmatrix}. \tag{37}$$

Fig. 6 shows the distribution of the LCI for the SB60 manipulator. The manipulator pose for the each LCI calculation can be obtained using Eq. (12). In this paper, the LCI is used to compare the optimized manipulator's structure with the original manipulator's structure and to analyse the ill-conditioning of the manipulator.

As shown in this figure, the LCI for both manipulators have similar



(a) LCI for the SB60 with the original parameters.



(b) LCI for the SB60 with the optimized parameters.

Fig. 6. Distribution of the LCI for (a) original SB60 and (b) the optimized SB60.

values. However, it is higher for the optimized manipulator than the original manipulator. This indicates that the new manipulator has the better ability to generate output force and torque. This is very important because, in rock-drilling processes, the manipulator needs to support significant force to keep the drill against the tunnel's face. Furthermore, the LCI for the optimized manipulator is almost constant, which means that the performance of the manipulator is also constant over the whole workspace and that there are no ill-conditioned spots.

5. Conclusions

In this paper, a generic method to optimize a redundant serial manipulator's structure was presented. The method finds an optimal solution in sense of a least squares to a constrained optimization problem. The case study showed that the proposed method can be used as an optimization tool.

It is important to note that this method gives only the skeletal structure of the manipulator. The link arms must be sized with proper material strengths, and the joints need to have suitable actuators. This may not be possible to achieve for the structure produced by the optimization process. Based on engineering knowledge, it is possible to add constraints, so that the link lengths and joint actuators can be implemented.

In reviewing the obtained results, we noticed that some of the changes in the DH-parameters are very significant. The reason for this is beyond the scope of this paper, but it should be noted that the working area of the original SB60 manipulator was somewhat larger than the working area for which it was optimized. Therefore, the optimized manipulator can be smaller than the original one. However, if the drilling task in the example reflects what is to be drilled, then the original manipulator is oversized for the desired task; therefore, its use may not be very profitable. Still, due to its optimized design, the new manipulator could save time and energy.

The interval analysis effect to the optimization results will be analyzed in future studies. The proposed optimization method will also be extended in order to determine the best possible structure for serial manipulators.

Acknowledgments

This work was funded by the Doctoral School of Industry Innovations at Tampere University of Technology (83026).

References

[1] M. Sobh, Y. Toundykov, Kinematic synthesis of robotic manipulators from task descriptions, Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003, vol. 2, 2003, pp. 1018–1023, , http://dx.doi.org/10.1109/CCA.2003.1223150.

[2] Y. Sun, H. Liu, Z. Luo, F. Wang, Robot mechanical structure optimization design, Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on, 2007, pp. 1919–1923.

[3] F. Ouezdou, S. Régnier, C. Mavroidis, Kinematic synthesis of manipulators using a distributed optimization method, J. Mech. Des. 121 (2007) 492–501.

[4] J.-P. Merlet, Democrat: A design methodology for the conception of robots with parallel architecture, Robotica 15 (1997) 367–373.

[5] E. Singla, S. Tripathi, V. Rakesh, B. Dasgupta, Dimensional synthesis of kinematically redundant serial manipulators for cluttered environments, Robot. Auton. Syst. 58 (5) (2010) 585–595.

[6] S. Barissi, H. Taghirad, Task based optimal geometric design and positioning of serial robotic manipulators, Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on, 2008, pp. 158–163.

[7] J.T. Wunderlich, Simulating a robotic arm in a box: Redundant kinematics, path planning, and rapid prototyping for enclosed spaces, SIMULATION 80 (6) (2004) 301–316.

[8] L.L. Sciavicco, B. Siciliano, Modelling and control of robot manipulators, Advanced textbooks in control and signal processing, Springer, London, New York, 2000.

[9] S.R. Buss, J.-S. Kim, Selectively damped least squares for inverse kinematics, J. Graph. Tools 10 (2004) 37–49.

[10] L.-W. Tsai, Mechanism Design: Enumeration of Kinematic Structures According to Function, Mechanical Engineering series, CRC Press, Washington, D.C., 2001.

[11] C. Feng, T. Liu, A graph-theory approach to designing deployable mechanism of reflector antenna, Acta Astronaut. 87 (0) (2013) 40–47.

[12] M.A. Pucheta, A. Cardona, Topological and dimensional synthesis of planar linkages for multiple kinematic tasks, Multibody Syst. Dyn. 29 (2) (2013) 189–211.

[13] G. Golub, W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, J. Soc. Ind. Appl. Math. Ser. B Numer. Anal. 2 (2) (1965) 205–224.

[14] S. Chiaverini, B. Siciliano, O. Egeland, Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator, IEEE Trans. Control Syst. Technol. 2 (2) (1994) 123–134, http://dx.doi.org/10.1109/87.294335.

[15] J. Baillieul, Kinematic programming alternatives for redundant manipulators, Proceedings. 1985 IEEE International Conference on Robotics and Automation, vol. 2, 1985, pp. 722–728, , http://dx.doi.org/10.1109/ROBOT.1985.1087234.

[16] C.A. Klein, C.H. Huang, Review of pseudoinverse control for use with kinematically redundant manipulators, IEEE Trans. Syst. Man Cybern. SMC-13 (2) (1983) 245–250, http://dx.doi.org/10.1109/TSMC.1983.6313123.

[17] T.F. Chan, R.V. Dubey, A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators, IEEE Trans. Robot. Autom. 11 (2) (1995) 286–292, http://dx.doi.org/10.1109/70.370511.

[18] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, Proceedings. 1985 IEEE International Conference on Robotics and Automation, vol. 2, 1985, pp. 500–505, , http://dx.doi.org/10.1109/ROBOT.1985.1087247.

[19] C.W. Warren, Global path planning using artificial potential fields, Proceedings, 1989 International Conference on Robotics and Automation, vol. 1, 1989, pp. 316–321, , http://dx.doi.org/10.1109/ROBOT.1989.100007.

[20] S. Patel, T. Sobh, Manipulator performance measures - a comprehensive literature survey, J. Intell. Robot. Syst. 77 (3) (2015) 547–570, http://dx.doi.org/10.1007/s10846-014-0024-y.

[21] J. Angeles, C.S. López-Cajún, Kinematic isotropy and the conditioning index of serial robotic manipulators, Int. J. Robot. Res. 11 (6) (1992) 560–571, http://dx.doi.org/10.1177/027836499201100605.

[22] J. Salisbury, J. Craig, Articulated hands: Force control and kinematic issues, Int. J. Robot. Res. 1 (1) (1982) 4–17, http://dx.doi.org/10.1177/027836498200100102.

[23] S. Patel, T. Sobh, Task based synthesis of serial manipulators, J. Adv. Res. 6 (3) (2015) 479–492, http://dx.doi.org/10.1016/j.jare.2014.12.006 (editors and International Board Member collection).

[24] T.M. Sobh, D.Y. Toundykov, Optimizing the tasks at hand [robotic manipulators], IEEE Robot. Autom. Mag. 11 (2) (2004) 78–85, http://dx.doi.org/10.1109/MRA.2004.1310944.

[25] P. Shiakolas, D. Koladiya, J. Kebrle, Optimum robot design based on task specifications using evolutionary techniques and kinematic, dynamic, and structural constraints, Inverse Probl. Eng. 10 (4) (2002) 359–375, http://dx.doi.org/10.1080/1068276021000004706.

[26] J. Craig, Introduction to Robotics: Mechanics and Control, Addison-Wesley series in electrical and computer engineering: Control engineering, Addison-Wesley, 1989.

[27] J.-P. Merlet, C.M. Gosselin, N. Mouly, Workspaces of planar parallel manipulators, Mech. Mach. Theory 33 (1) (1998) 7–20, http://dx.doi.org/10.1016/S0094-114X(97)00025-6.

[28] L.J. Stocco, S.E. Salcudean, F. Sassani, On the use of scaling matrices for task-specific robot design, IEEE Trans. Robot. Autom. 15 (5) (1999) 958–965, http://dx.doi.org/10.1109/70.795800.

[29] J.H. Lee, K.S. Eom, I.I. Suh, Design of a new 6-dof parallel haptic device, Proceedings 2001 ICRA, IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), vol. 1, 2001, pp. 886–891, , http://dx.doi.org/10.1109/ROBOT.2001.932662.

[30] R. Vijaykumar, K.J. Waldron, M.J. Tsai, Geometric optimization of serial chain manipulator structures for working volume, Int. J. Robot. Res. 5 (2) (1986) 91–103, http://dx.doi.org/10.1177/027836498600500210.

[31] S. Kucuk, Z. Bingul, Comparative study of performance indices for fundamental robot manipulators, Robot. Auton. Syst. 54 (7) (2006) 567–573, http://dx.doi.org/10.1016/j.robot.2006.04.002.

Publication III

Tuomo Kivelä, Jouni Mattila, Jussi Puura, and Sirpa Launis, (2018), "Redundant Robotic Manipulator Path Planning for Real-Time Obstacle and Self-Collision Avoidance", in *Advances in Service and Industrial Robotics. RAAD 2017. Mechanisms and Machine Science*, vol 49. Springer, Cham

Redundant Robotic Manipulator Path Planning for Real-Time Obstacle and Self-Collision Avoidance

Tuomo Kivelä[1(✉)], Jouni Mattila[1], Jussi Puura[2], and Sirpa Launis[2]

[1] Laboratory of Automation and Hydraulics,
Tampere University of Technology, Korkeakoulunkatu 10, Tampere, Finland
tuomo.kivela@tut.fi
[2] Sandvik Mining and Construction Oy, Pihtisulunkatu 9, Tampere, Finland

Abstract. This paper presents a method to generate joint trajectories for a redundant manipulator. The control system of the manipulator determines the joint references so that the goal pose can be reached without any collisions, in real-time. The control system checks weather any part of the manipulator is at risk of colliding with itself or with any obstacles. If there is a risk of collision, then the collision server computes the exact points where the collision is about to happen and calculates the shortest distance between the colliding objects. The joint trajectories of the manipulator are modified so that collisions will be avoided while at the same time, the trajectory of the end-effector maintains its initial trajectory if possible. Experimental results are given for a 7 DOF redundant manipulator to demonstrate the capability of the collision avoidance control system.

Keywords: Collision avoidance · Robotic manipulator · Redundant · Real-time control

1 Introduction

In the last few decades, many papers have described collision avoidance methods for autonomous robots. This is because a reliable collision avoidance system is believed to play an important role in the field of autonomous robotics. A long-standing problem has been obtaining reliable information from the robot environment and how to react to the collision information. The general features of collision avoidance methods are well-known, and different approaches to solving the collision avoidance problem have been reported in the literature. One of the most common approaches for finding the minimum distance between two objects is based on bounding boxes [1–4]. These sample-based methods are fast and require low computation power, but they are applicable only to simple geometries. For continuous motions, several approaches have been developed to detect collisions, such as swept-volume methods [5,6], the trajectory parameterization method [7], and feature-tracking methods [8,9]. However, these methods

C. Ferraresi and G. Quaglia (eds.), *Advances in Service and Industrial Robotics*,
Mechanisms and Machine Science 49, DOI 10.1007/978-3-319-61276-8_24

suitable only for simple geometries. A method for detecting collisions between more complex geometry is presented by [10], in which graphics processing unit (GPU)-based calculations are required use the algorithm in real-time.

Despite the importance of exact collision detection, only a very few papers have studied how to accurately detect the collision points of two geometrically complex objects and use that information in a real-time control system to avoid collisions. Thus, additional studies of exact collision detection and shortest distance calculation are needed.

The aim of this paper is to present a developed collision avoidance system. In this paper, we propose a novel architecture for detecting the exact collision points of two geometrically complex objects and how to use this information for the collision avoidance system. The process to find the shortest distance between two point clouds is divided into two phases. In the first phase, we find the approximate collision location of both point clouds and then use this information to find the exact collision points. Extended oriented bounding boxes (EOBBs) are used in the first phase to detect possible collisions and to find the approximate collision location. Therefore, the proposed method calculates the shortest distance between two complex point clouds in real-time by using the central processing unit (CPU). The proposed method uses the calculated shortest distance between two point clouds to move the manipulator away from the obstacles and toward the desired goal.

An experiment validated that the proposed method can be used in real-time control. The three point clouds used in the experiment were converted from three-dimensional (3D) computer-aided design (CAD) models, where the number of points in each point cloud was restricted to 1000. With these point clouds, we created two collision pairs and monitored the shortest distance between the pairs.

We organized the rest of this paper as follows. Section 2 presents the proposed collision detection method and how to calculate the shortest distance between two objects. The proposed collision avoidance algorithm is presented in Sect. 3. Section 4 illustrates the overall collision avoidance control system and its implementation. Then the following section shows experimental results for the proposed method. Finally, Sect. 6 draws some conclusions and future works.

2 Collision Detection

To avoid the manipulator's collisions with obstacles or self-collision, a manipulator control system has to detect the obstacles, find the minimum distances between different parts of the manipulator and the obstacles, and between different parts of the manipulator itself. Finding the exact shortest distance between the manipulator and obstacles is crucial for an accurate collision avoidance system. For example, if the workspace of the manipulator is very small, it is not possible to simplify the manipulator and obstacles with bounding boxes, because the volumes of the bounding boxes are always bigger than the actual volume of the manipulator parts and the environment obstacles. Several studies used bounding

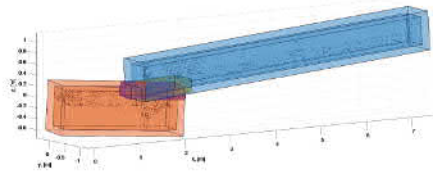
boxes to represent obstacles and the manipulator and then found the shortest distance between these bounding boxes; for example, [3] and [4] used oriented bounding boxes (OBBs). Another very popular method for finding the shortest distance between obstacles is to use vision-based techniques [11] and [12]. Although, the vision-based methods are widely used and are effective methods for calculating the shortest distance, they cannot be used in applications with a dirty and harsh environment, such as underground mining applications. In applications with a challenging environment, a laser-based measurement system that can produce point clouds is more preferable. The use of the point cloud-based distance calculation method ensures that there is no extra volume around the obstacles that could lead to an incorrect shortest distance calculation. Calculating the shortest distance between two point clouds is a challenge for a real-time control system. Therefore, the calculation procedure is divided into different stages, which ensures that the shortest distance can be found in real-time. The overall shortest distance calculation process is described only at the high level (Table 1) because the details of the shortest distance calculation are beyond the scope of this paper.

Table 1. Finding the shortest distance between two point clouds.

Initialization phase	
1	Input: Point clouds
2	Create Octrees for the point clouds
3	Create axis-aligned bounding boxes (AABBs) with δx larger than the tight-fit BBs
Calculation phase	
4	Input: Objects poses
5	Create extended oriented bounding boxes (EOBBs) form AABBs with the input poses
6	OBBs' intersection by the separating axis theorem (SAT) - No intersection $\rightarrow$ No risk of collision
7	Calculate the overlapping volume of the EOBBs - Create new OBBs of the volumes within each other
8	Transform the new OBBs to the news AABBs with the inverse of the input poses
9	Find the Octree points that are inside the new AABBs - No points found $\rightarrow$ No risk of collision
10	Transform found points with the input poses
11	Find the shortest distance within the transformed points

Figure 1a shows a manipulator used in underground tunneling rigs. Figure 1b shows the point cloud representation of Fig. 1a and the OBBs around the studied parts of the manipulator. EOBBs are used to detect possible collisions.

(a) Sandvik drilling rig with two manipulators (Photo: Sandvik)



(b) Point clouds inside corresponding OBBs and EOBBs for the single manipulator

Fig. 1. Real drilling rig and corresponding point clouds and BBs of the drilling manipulator

The EOBBs were first used by [13], where they simulated crowd motion in the 2D case. In this work, we extend the original idea of the EOBB to cover the 3D case. The use of EOBBs enables us to detect collisions enough early so that the control system is able to react to collision warnings. If the EOBBs do not intersect, then there is no risk of collision.

3 Collision Avoidance

The relation between a manipulator joint value vector $\mathbf{q} \in \mathbb{R}^n$ and the end-effector pose vector $\mathbf{x} \in \mathbb{R}^m$ is given by

$$\mathbf{x} = f(\mathbf{q}), \tag{1}$$

where $f(\cdot)$ is a nonlinear direct kinematics function of the manipulator. The end-effector linear velocity and angular velocity can be realized by mean of the following relation:

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}}, \tag{2}$$

where $\mathbf{J}$ is the geometric Jacobian matrix of the manipulator [14]. The main advantage of Eq. (2) over Eq. (1) is its linearity in the configuration velocities. This allows us to solve the differential kinematics with an inversion of the Jacobian matrix

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\,\dot{\mathbf{x}}, \tag{3}$$

which then can be integrated to give $\mathbf{q}$.

Equation (3) is valid only for manipulators that have the same dimension of the operational space and the joint space ($m = n$). When the manipulator is redundant, ($m < n$), the Jacobian matrix has more columns than rows, and an infinite number of solutions exist for Eq. (2). Therefore, a direct Jacobian

matrix inverse cannot be done. If the Jacobian matrix is a non-square matrix, then a pseudo-inverse can be adopted [14]:

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}(\mathbf{q})\,\dot{\mathbf{x}}, \tag{4}$$

and the matrix,

$$\mathbf{J}^{\dagger} = \mathbf{J}^{T}\left(\mathbf{J}\mathbf{J}^{T}\right)^{-1}, \tag{5}$$

is the right pseudo-inverse of $\mathbf{J}(\mathbf{q})$.

The pseudo-inverse have stability problems in the neighborhood of singularities. At a singularity, the Jacobian matrix no longer has a full row rank. If the configuration is close to a singularity, then the pseudo-inverse method will lead to very large changes in the joint angles, even for small movements in the target position. The damped least squares method avoids many of the pseudo-inverse method's problems with singularities and can give a numerically stable method for selecting $\dot{\mathbf{q}}$.

$$\dot{\mathbf{q}} = \mathbf{J}^{T}\left(\mathbf{J}\mathbf{J}^{T} + \lambda^{2}\mathbf{I}\right)^{-1}\dot{\mathbf{x}}, \tag{6}$$

where λ is a non-zero damping constant, and the right damped pseudo-inverse gets the form

$$\tilde{\mathbf{J}}^{\dagger} = \mathbf{J}^{T}\left(\mathbf{J}\mathbf{J}^{T} + \lambda^{2}\mathbf{I}\right)^{-1}. \tag{7}$$

The damping constant depends on the details of the multibody and the target positions and must be chosen carefully to make Eq. (6) numerically stable. If $\lambda = 0$, Eq. (4) becomes identical to Eq. (6), which is ill-conditioned near to a singularity.

The solution for Eq. (2) is shown [15] to be in the general form

$$\dot{\mathbf{q}} = \tilde{\mathbf{J}}^{\dagger}\dot{\mathbf{x}} + \left(\mathbf{I} - \tilde{\mathbf{J}}^{\dagger}\mathbf{J}\right)\mathbf{z}, \tag{8}$$

where $\mathbf{z}$ is an arbitrary vector in the $\dot{\mathbf{q}}$-space. A projection operator $\left(\mathbf{I} - \tilde{\mathbf{J}}^{\dagger}\mathbf{J}\right)$ describes the redundancy of the system and can be used to map an arbitrary $\dot{\mathbf{q}}$ into the null space of the transformation.

3.1 Path Planner

The collision-free path planner is based on Eq. (8) and the approach proposed by [16]. The primary goal is to reach the goal pose with the manipulator's end-effector, and the secondary goal is to avoid collisions. In [16], the authors presented a method for avoiding collisions with one obstacle avoidance point. In this paper, this approach is extended to cover multiple obstacle avoidance points. The primary and secondary goals are described by the equations

$$\mathbf{J}\dot{\mathbf{q}} = \dot{\mathbf{x}}, \tag{9}$$

$$\mathbf{J}_{O_i}\dot{\mathbf{q}} = \dot{\mathbf{x}}_{O_i}, \tag{10}$$

where $\mathbf{J}_{O_i}$ is the obstacle avoidance point Jacobian, and $\dot{\mathbf{x}}_{O_i}$ is the obstacle avoidance point velocity, which includes the direction and the velocity of the obstacle avoidance point.

The secondary goal solution for $\dot{\mathbf{q}}$ is used as an arbitrary vector $\mathbf{z}$ to modify the solution of Eq. (8). By combining Eqs. (8) and (10) with multiple obstacle avoidance points, the following solution can be derived:

$$\dot{\mathbf{q}} = \tilde{\mathbf{J}}^{\dagger}\dot{\mathbf{x}} + \sum_{i=1}^{n_o} \left[\left(\mathbf{I} - \tilde{\mathbf{J}}^{\dagger}\mathbf{J} \right) \tilde{\mathbf{J}}_{O_i}^{\dagger} \dot{\mathbf{x}}_{O_i} \right], \tag{11}$$

where n_o is the number of obstacle avoidance points, and $\tilde{\mathbf{J}}_{O_i}^{\dagger}$ is the damped right-pseudo inverse solution for $\mathbf{J}_{O_i}$. If the desired path cannot be maintained the Eq. 11 give up the desired velocity $\dot{\mathbf{x}}$ to ensure the collision free path.

4 Collision Avoidance Control System

Collision avoidance is based on the principle of the artificial potential field (APF) method. The APF method is widely used in collision avoidance applications and one of the first study to use this method was [17]. Thereafter, the principle of the APF method has been extended to support different collision avoidance algorithms. The very basic idea of the APF is to represent the manipulator and obstacles as a positive charge, and the goal acts as a negative charge. Therefore, the manipulator and the obstacles repel each other by generating a repulsive force, and the goal attracts the manipulator due to the opposite charge. It is shown in [17] that the desired velocity vector gets the following form

$$\dot{\mathbf{x}} = -\mathbf{k}\left(\mathbf{x}_g - \mathbf{x}\right), \tag{12}$$

where $\mathbf{k}$ is the velocity gain and $\mathbf{x}_g$ is the goal pose.

4.1 Collision Avoidance Architecture

The collision avoidance architecture proposed in this paper consists of three main subsystems: a collision server, a control system, and a manipulator. The collision server contains information about all point clouds. The point clouds are static point clouds which means that their shape does not change during the task. The position and the orientation of the static point clouds are updated based on the measurement system. The collision server calculates the shortest distance between different parts of the system and sends this information to the control system. The control system resolves new joint references based on the current configuration of the manipulator, the goal pose, and the distance information from the collision server. The new joint references are calculated so that the manipulator moves toward the goal pose, and at the same time, the manipulator avoids self-collisions and collisions with the environment and other obstacles. The manipulator moves according with the control signals from the control system. Figure 2 illustrates the overall collision avoidance architecture.

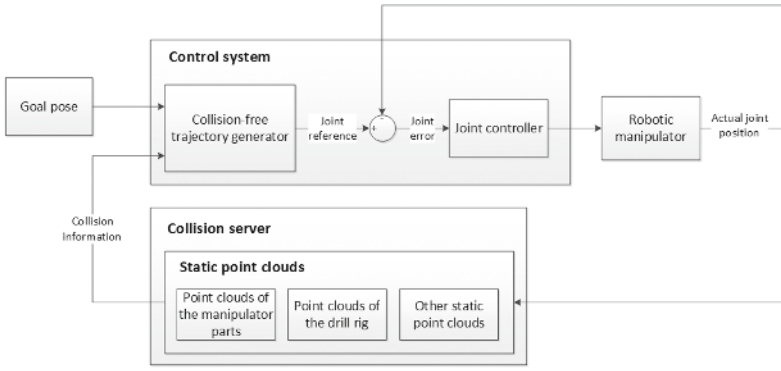


Fig. 2. Collision avoidance architecture

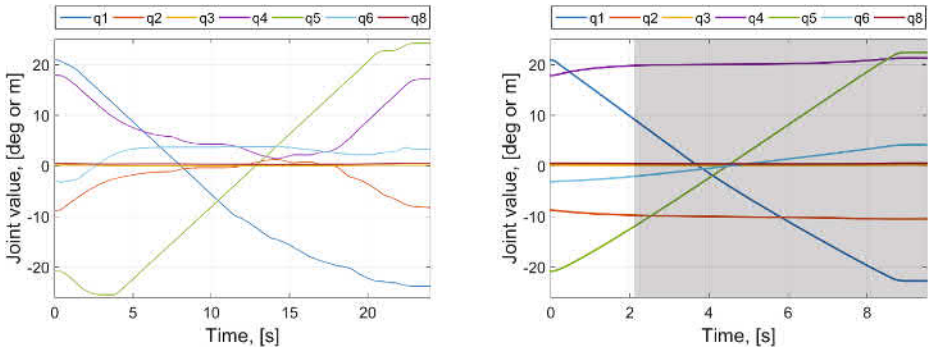
5 Experiment

The collision avoidance control system was developed to enable autonomous operation for a redundant serial manipulator. To verify this functionality, an experimental test was carried out with a heavy-duty hydraulic manipulator, which is used in underground tunneling. The manipulator itself is a redundant manipulator with seven active joints.

In order to validate the collision avoidance system, we conducted the following test. The task was to move the manipulator from a starting pose ($\mathbf{q}_s$) to a goal pose ($\mathbf{q}_g$) by avoiding self-collision. The starting pose is at the left side of the manipulator's base with a diagonally upward rotated end-effector. The goal pose is at the other side of the manipulator with the same end-effector pose except the y-coordinate is the inverse. With a conventional control system, the end-effector of the manipulator collides with the lift part of the manipulator, if it tries to move from the start pose to the goal pose. The proposed collision avoidance control system of the manipulator executes the desired task without self-collision by finding a collision-free path.

For the experiment, the 3D parts of the manipulator that could collide with other parts of the manipulator were converted to point clouds, each containing 1000 points (Fig. 1b). Thereafter, we created rules to monitor the shortest distance between possible collision pairs. If the distance between the collision pair was larger than the collision margin, then it has no influence on the control of the manipulator, but if the distance was smaller than the collision margin, then we used Eq. 11 to modify the trajectory of the manipulator based on the calculated shortest distance between two point clouds.

During the experiment, the collision server ran on a laptop with an Intel Core i7 processor. The server program to calculate the shortest distance between point clouds was programmed with C++/CLI and compiled to a standalone program. The average time to calculate the distance was 8 ms. This ensures that the proposed method can be used in a real-time control system.



(a) Joint values with collision avoidance (b) Joint values without collision avoidance

Fig. 3. The manipulator's joint values with and without the collision avoidance

Figure 3b shows the simulated joint values for the conventional control system to move the manipulator from $\mathbf{q}_s$ to $\mathbf{q}_g$. With this control approach, the manipulator will self-collide at the time $2.16s$ from the beginning. The gray area represents the manipulator's configurations that cannot be achieved without collisions. Figure 3a shows the equivalent experiment with the proposed collision avoidance control system. The joint trajectories are modified so that the desired task is achieved without collisions.

6 Conclusion

Collision avoidance systems are one of the most challenging problems for autonomous manipulators to accomplish the given task. An improved shortest distance-based collision avoidance method was proposed to obtain a collision-free path from the starting pose to the goal pose. Multiple simultaneous collision avoidance points can be used to avoid self-collisions and any part of the manipulator from colliding with obstacles or environment. The proposed method can be used with a real-time control system, and the method does not require any path planning time, because all decisions are made on the fly. This advanced shortest distance calculation method ensures that the shortest distance between point cloud type objects can be calculated in real-time. The shortest distance calculation method also minimizes the calculation power needed to find the shortest distance.

In future work, we intend to improve the collision avoidance system to cover more complex systems with multiple robotic manipulators and a dynamic environment.

References

1. Gottschalk S, Lin MC, Manocha D (1996) OBB-Tree: a hierarchical structure for rapid interference detection. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996. ACM, pp 171–180
2. van den Bergen G (1998) Efficient collision detection of complex deformable models using AABB trees. J Graph Tools 2(4):1–13
3. Puiu D, Moldoveanu F (2011) Real-time collision avoidance for redundant manipulators. In: 2011 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), pp 403–408
4. Lee I, Lee KK, Sim O, Woo KS, Buyoun C, Oh JH (2015) Collision detection system for the practical use of the humanoid robot. In: 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pp 972–976
5. Cameron S (1990) Collision detection by four-dimensional intersection testing. IEEE Trans Robot Autom 6:291–302
6. Kwon J, Khatib O (2013) Adaptive collision checking for continuous robot motions within motion constraints. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 5365–5372
7. Canny J (1986) Collision detection for moving polyhedra. IEEE Trans Pattern Anal Mach Intell 8(2):200–209
8. Mirtich B (1998) V-Clip: fast and robust polyhedral collision detection. ACM Trans Graph 17(3):177–208
9. Cohen JD, Lin MC, Manocha D, Ponamgi M (1995) I-COLLIDE: an interactive and exact collision detection system for large-scale environments. In: Proceedings of the 1995 Symposium on Interactive 3D Graphics, I3D 1995. ACM, p 189
10. Kaldestad KB, Haddadin S, Belder R, Hovland G, Anisi DA (2014) Collision avoidance with potential fields based on parallel processing of 3D-point cloud data on the GPU. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp 3250–3257
11. Asfour T, Azad P, Vahrenkamp N, Regenstein K, Bierbaum A, Welke K, Schróder J, Dillmann R (2008) Toward humanoid manipulation in human-centred environments. Robot Auton Syst 56(1):54–65 Human Technologies: Know-how
12. Grigorescu SM, Ristić-Durrant D, Gräser A (2009) ROVIS: robust machine vision for service robotic system friend. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 3574–3581
13. Mukundan R, Li B (2012) Crowd simulation: extended oriented bounding boxes for geometry and motion representation. In: Proceedings of the 27th Conference on Image and Vision Computing New Zealand, IVCNZ 2012. ACM, pp 121–125
14. Sciavicco LL, Siciliano B (2000) Modelling and control of robot manipulators. In: Advanced Textbooks in Control and Signal Processing. Springer, London
15. Greville TNE (1959) The pseudoinverse of a rectangular or singular matrix and its application to the solution of systems of linear equations. SIAM Rev 1(1):38–43
16. Maciejewski AA, Klein CA (1985) Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. Int J Robot Res 4(3):109–117
17. Khatib O (1985) Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings of 1985 IEEE International Conference on Robotics and Automation, vol 2, pp 500–505

Publication IV

Tuomo Kivelä, Jouni Mattila, Jussi Puura, and Sirpa Launis, "On-line Path Planning With Collision Avoidance for Coordinate Controlled Robotic Manipulators" in *Proceedings of the 2017 Bath/ASME Symposium on Fluid Power and Motion Control*, Sarasota, FL, USA, October 16-19, 2017.

Publication V

Tuomo Kivelä, Pauli Mustalahti, and Jouni Mattila, "Real-time Distance Query and Collision Avoidance for Point Clouds with Heavy-duty Redundant Manipulator" in *Proceedings of the 8th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Ningbo, China, November 19-21, 2017.

Real-time Distance Query and Collision Avoidance for Point Clouds with Heavy-duty Redundant Manipulator

Tuomo Kivelä, Pauli Mustalahti, and Jouni Mattila
Laboratory of Automation and Hydraulics
Tampere University of Technology
33720 Tampere, Finland
Emails: tuomo.kivela@tut.fi,
pauli.mustalahti@tut.fi, jouni.mattila@tut.fi

***Abstract*—This paper presents a real-time method for generating joint trajectories for redundant manipulators with collision avoidance capability. The coordinated motion control system of the heavy-duty hydraulic manipulator resolves joint references so that a goal position can be reached in real-time without any collisions. The proposed method is able to detect and prevent different types of possible collisions, including self-collisions and collisions with obstacles. When the control system detects the risk of collision, the collision server searches the points where the collision is about to occur and calculates the shortest distance between the colliding objects. The collision server is used to retain static point clouds and to calculate the shortest distance between objects that are too close to each other. The point clouds on the server are kept up to date with the manipulators' joint sensors and laser scanner-based measurements. During coordinated motion control, the joint trajectories of the redundant manipulator are modified so that the collisions can be avoided, while at the same time, the trajectory of the end-effector maintains its initial trajectory if possible. Results are given for a 4-DOF redundant heavy-duty hydraulic manipulator to demonstrate the capability of this collision avoidance control system.**

I. Introduction

During the last several decades, many researchers have described collision avoidance methods for manipulators in the field of autonomous robotics. The main reason for this is the increased need for automated operations where the ability to avoid collisions plays an important role. Long-standing problems include obtaining accurate information from the manipulator environment and how to actually prevent the collision if a potential collision is detected. The general features of collision avoidance methods are well-known, and different approaches to solving the collision avoidance problem have been reported in the literature. Two main strategy types have been proposed for solving collision avoidance problems: global strategies and local strategies.

Global collision avoidance methods [1], [2] find a collision-free path from the starting position to the goal position if such a path exists. These methods are typical for mobile robots, where the task is to find a collision-free path to the desired location. Existing knowledge of the robot environment is used to plan the route. Therefore, these methods are sometimes called planning algorithms. However, global methods are computationally heavy and may not be applicable for real-time control applications.

Local strategy solutions are very popular real-time collision avoidance methods [3]. These methods treat obstacle avoidance as a control problem instead of a path planning problem. This enables modification of the manipulator's path if obstacles are detected or if the obstacles move too close to the manipulator. Therefore, these methods can be used in an unknown environment where there is no information about the location of the obstacles during the desired manipulator operations.

Local methods [4], [5], [6], [7] are based on potential fields in the manipulator's operational space. These methods are further divided into two categories based on how the methods use potential fields. When the collision avoidance method uses potential fields to generate forces on the manipulator [4], [5], the solution requires position-based impedance control (PBIC). Another approach is to use potential fields to generate joint velocities directly [6], [7]. This approach does not require a dynamic model of the manipulator and therefore is easier to implement.

Successful collision avoidance requires accurate information where a potential collision is about to happen. Despite the importance of detecting exact locations of potential collisions, only a few papers have examined how to detect these collision points of two geometrically complex objects. Often, the complex geometry of the objects is ignored by simplifying the geometry or by representing the objects as bounding boxes (BB). One of the most common approaches for finding the minimum distance between two objects is based on BBs [8], [9], [10], [11]. These sample-based methods are fast and require low computational power, but they are applicable only to simple geometries. A method for detecting collisions between more complex geometry is presented by [12], in which graphics processing unit (GPU)-based calculations for real-time implementation are required. Solutions with real-time capability, accurate enough collision detection, and a shortest distance query method seem to be lacking in the

literature. Therefore, additional studies on detecting collisions exactly and calculating the shortest distance are needed.

The aim of this paper is to present a collision avoidance system. We propose a method for detecting the collision points between two objects, presented as point clouds, and describe how to use this information in a collision avoidance system. The process for finding the shortest distance between two point clouds is divided into two phases. In the first phase, we find the approximate collision location of both point clouds and then use this information to find the exact collision points. Oriented bounding boxes (OBBs) are used in the first phase to detect possible collisions and to find the approximate collision location. After this, the actual shortest distance is calculated by using Euclidean distance. If there is a risk that the objects are about to collide, the calculated shortest distance is used with the proposed collision avoidance method to move the manipulator away from the obstacles and toward the desired goal.

Experiments validated that the proposed method can be used in real-time control. Multiple point clouds, representing the manipulator, were converted from three-dimensional (3D) computer-aided design models, where the number of points in each point cloud was restricted to 1000. Point clouds representing the environment and obstacles were created with a 3D laser scanner. With these point clouds, we created several collision pairs and monitored the shortest distance between the pairs. The results show that the proposed method is suitable for real-time applications to avoid collisions in which there can be one or more simultaneous collision risks.

The remainder of this paper is organized as follows: The next section presents the proposed collision detection method and how to calculate the shortest distance between two point cloud-type objects. The proposed collision avoidance algorithm is presented in the following section. Next, we describe the obstacle and environment detection system based on a 3D laser scanner. Finally, we show the experimental results for the proposed method, draw some conclusions, and describe future work.

II. Distance Query

To prevent the manipulator from colliding with obstacles or with itself, a manipulator control system must detect obstacles, as well as find the minimum distance between different parts of the manipulator and the obstacles. In the case of possible self-collisions, the distance between different parts of the manipulator has to be calculated. Finding the exact shortest distances between the manipulator and obstacles is crucial for an accurate collision avoidance system. For example, if the manipulator's workspace is small, it might not be possible to simplify the manipulator and the obstacles with BBs, because the volumes of the BBs are always larger than the actual volumes of the manipulator parts and the obstacles. This might lead to a situation where there is no more free space for the manipulator to move. Commonly, two types of BBs are used to simplify the distance query: an axis-aligned bounding box (AABB) and an oriented bounding box (OBB). AABBs are aligned with global coordinate axes, and the size of the AABB might change if the object inside is rotated. OBBs, instead, are aligned with the local coordinate axes of the object; therefore, the size of the OBB is not changed when the object inside is rotated. Choosing between the different BBs depends on the application.

TABLE I
Finding the Shortest Distance Between Two Point Clouds.

Initialization phase
1. Input: Point clouds
2. Create octrees for the point clouds
3. Create axis-aligned bounding boxes (AABBs) with δx larger than the tight-fit AABBs
Calculation phase
4. Input: Object poses
5. Create extended oriented bounding boxes (EOBBs) from AABBs with the input poses
6. Intersection of the EOBBs using the separating axis theorem (SAT)
• No intersection $\rightarrow$ No risk of collision
7. Calculate the overlapping volume of the EOBBs
• Create new OBBs of the volumes within each other
8. Transform the new OBBs into the new AABBs with the inverse of the input poses
9. Find the octree points that are inside the new AABBs
• No points found $\rightarrow$ No risk of collision
10. Transform the found points with the input poses
11. Find the shortest distance within the transformed points

Several studies have used BBs to represent obstacles and the manipulator and then found the shortest distance between BBs; for example, [10] and [11] used OBBs. Another popular method for finding the shortest distance between obstacles is to use vision-based techniques [13], [14]. Vision-based methods are widely used and are effective for calculating the shortest distance if the conditions are correct. For example, lighting should be appropriate so that the camera is able to see the objects. If the conditions are not good enough, vision-based methods cannot be used, for example, in applications with dirty and harsh environments, such as in underground mining applications. In applications with challenging environments, a laser-based measurement system that can produce point clouds is preferable.

Implementation of the point cloud-based distance query method is straightforward. However, calculating the shortest distance between two point clouds is a challenge for a real-time control system, because the calculation time depends on the size of the point clouds. Therefore, it is necessary to reduce the number of point cloud points among which the shortest distance can be found. This can be achieved by using a two-state calculation process. In the first state, an approximate location of the shortest distance is discovered. The second state is then used to calculate the actual shortest distance based on the approximate location. The overall process for calculating the shortest distance is described in Table I. In short, different types of BBs are used to minimize the number of point cloud points among which the shortest distance can be found. BBs are used to find an overlapping volume of two BBs. Then, an

Fig. 1. Hiab crane with an obstacle in its workspace.

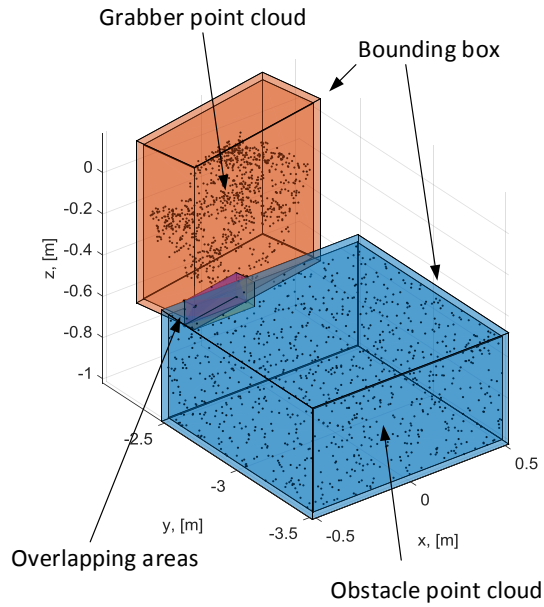


Fig. 2. The grabber of the manipulator and an artificial obstacle. The point clouds inside bounding boxes represent the grabber and the obstacle.

octree data structure [15] is used to extract subsets of points from the original point clouds according to the overlapping volume. Finally, the shortest distance between these subsets can be calculated, for example, using the Euclidean distance calculation.

Fig. 1 shows a typical hydraulic 4 degrees of freedom (4-DOF) manipulator used, for example, to lift logs. This figure also shows an obstacle inside the manipulator's workspace that needs to be avoided while operating the manipulator. To detect collisions and calculate the shortest distances between the manipulator and the obstacle, both are modeled with point clouds. The critical part of the manipulator, as a point cloud, is then used to detect potential collisions with obstacles. Fig. 2 shows these critical point clouds of the manipulator and the obstacle and the OBBs around these point clouds. EOBBs are used to detect potential collisions. The use of EOBBs enables us to detect collisions early enough so that the control system is able to react to the collision warnings. If the EOBBs do not intersect, then there is no risk of collision.

III. Collision Avoidance

The collision avoidance method used in this paper is based on the principle of the artificial potential field (APF) method. The APF method was pioneered by Khatib [5]. Thereafter, the principle of the APF method was extended to support different collision avoidance algorithms. Within the APF method, the manipulator and obstacles can be thought to have the same charge, and the goal position of the manipulator acts as a different charge. Therefore, the manipulator and the obstacles repel each other by generating a repulsive force between each other, and the goal position attracts the manipulator due to the opposite charge.

The final force of the manipulator is achieved by combining the attractive force and the repulsive forces caused by obstacles:

$$\mathbf{F}(\mathbf{p}) = -\nabla \mathbf{U}(\mathbf{p}), \tag{1}$$

$$\mathbf{U}(\mathbf{p}) = \mathbf{U}_{att}(\mathbf{p}) + \mathbf{U}_{rep}(\mathbf{p}), \tag{2}$$

where $\mathbf{F}$ is the final force, $\mathbf{U}$ is the sum of the potentials, and $\mathbf{p}$ is the position of the manipulator. The potential functions $\mathbf{U}_{att}$ and $\mathbf{U}_{rep}$ must be selected properly to achieve suitable results. In this paper, we rely on the potential functions proposed by [5].

A. Collision Avoidance for an End-Effector

A path for the end-effector of the manipulator to follow is created using a proportional position controller. Let $\mathbf{p}_d$ represent the goal position of the manipulator's end-effector. Then the position controller can be described as follows:

$$\dot{\mathbf{p}}_G = k_v \left(\mathbf{p}_d - \mathbf{p} \right), \tag{3}$$

where $\mathbf{p}_d$ is the goal position of the end-effector, and k_v is the position control coefficient.

The shortest distance between the manipulator parts and the obstacles is calculated by using the method proposed in section II. When the manipulator is too close to the obstacles, it is necessary to produce a repulsive force that prevents the manipulator from colliding with the obstacles. This can be

achieved by producing an obstacle avoidance point velocity. The obstacle avoidance point velocity is calculated by using the shortest distance calculation method described in section II and by using the artificial potential field proposed by [5]:

$$\mathbf{U}_{O_i} = \begin{cases} \frac{1}{2}\mu\left(\frac{1}{\rho_i} - \frac{1}{\rho_O}\right), & \text{if } \rho_i \leq \rho_O \\ 0, & \text{if } \rho_i > \rho_O, \end{cases} \tag{4}$$

where ρ_O is the limit for collision avoidance, ρ_i is the distance between two obstacles, and μ is a scalar coefficient. Now $\dot{\mathbf{p}}_{O_i}$ can be formulated with (4),

$$\dot{\mathbf{p}}_{O_i} = \begin{cases} \mu\left(\frac{1}{\rho_i} - \frac{1}{\rho_O}\right)\frac{1}{\rho_i^2}\frac{\delta\rho_i}{\delta\dot{\mathbf{p}}_{O_i}}, & \text{if } \rho_i \leq \rho_O \\ 0, & \text{if } \rho_i > \rho_O, \end{cases} \tag{5}$$

where $\frac{\delta\rho_i}{\delta\dot{\mathbf{p}}_{O_i}}$ is the direction of the collision line. The path for the manipulator's end-effector can be calculated to be the sum of the attractive velocity and the repulsive velocities as follows:

$$\dot{\mathbf{p}} = \dot{\mathbf{p}}_G + \sum_{i=1}^{n_e} \dot{\mathbf{p}}_{O_i}, \tag{6}$$

where n_e is the number of end-effector obstacle avoidance points.

The linear velocity of the manipulator's end-effector must be bounded. This allows the use of a larger controller coefficient for the velocity, which ensures that the demanded end-effector velocity is also suitable near the goal position. The end-effector velocity is bounded as follows:

$$\dot{\mathbf{p}} = \begin{cases} \frac{\dot{\mathbf{p}}\dot{\mathbf{p}}_{max}}{max(|\dot{\mathbf{p}}|)}, & \text{if } max(|\dot{\mathbf{p}}|) > \dot{\mathbf{p}}_{max} \\ \dot{\mathbf{p}}, & \text{otherwise,} \end{cases} \tag{7}$$

where $\dot{\mathbf{p}}_{max}$ is the maximum linear velocity of the end-effector.

B. Collision Avoidance for a Manipulator's Body

If the manipulator's body parts are about to collide, these collisions have to handled differently from end-effector collisions. Repulsive velocities can be calculated as described in (5), but these velocities cannot be added directly to (6). Instead, a collision-free path planner based on null-space projection and the approach proposed by [16] have to be used. In [17], the general form of the solution for a null-space projection is shown to be

$$\dot{\mathbf{q}} = \tilde{\mathbf{J}}_G^{\dagger}\dot{\mathbf{p}} + \left(\mathbf{I} - \tilde{\mathbf{J}}_G^{\dagger}\mathbf{J}_G\right)\mathbf{z}, \tag{8}$$

where $\tilde{\mathbf{J}}_G^{\dagger}$ is a right damped pseudo-inverse of the manipulator's Jacobian matrix ($\mathbf{J}_G$), and $\mathbf{z}$ is an arbitrary vector in the $\dot{\mathbf{q}}$ space. A projection operator $\left(\mathbf{I} - \tilde{\mathbf{J}}_G^{\dagger}\mathbf{J}_G\right)$ describes the redundancy of the system and can be used to map an arbitrary $\dot{\mathbf{q}}$ into the null space of the transformation.

The primary goal is to reach the goal position with the manipulator's end-effector; the secondary goal is to avoid collisions with the manipulator's body. In [16], the authors presented a method for avoiding collisions with one obstacle avoidance point. In the present paper, this approach is extended to cover multiple obstacle avoidance points. The primary and secondary goals are described by the equations

$$\mathbf{J}_G\dot{\mathbf{q}} = \dot{\mathbf{p}}, \tag{9}$$

$$\mathbf{J}_{O_i}\dot{\mathbf{q}} = \dot{\mathbf{p}}_{O_i}, \tag{10}$$

where $\mathbf{J}_{O_i}$ is the obstacle avoidance point Jacobian, and $\dot{\mathbf{p}}_{O_i}$ is the obstacle avoidance point velocity.

The secondary goal solution for $\dot{\mathbf{q}}$ is used as an arbitrary vector $\mathbf{z}$ to modify the solution of (8). By combining (8) and (10) with multiple obstacle avoidance points, the following solution can be derived:

$$\dot{\mathbf{q}} = \tilde{\mathbf{J}}_G^{\dagger}\dot{\mathbf{p}} + \sum_{i=1}^{n_o}\left[\left(\mathbf{I} - \tilde{\mathbf{J}}_G^{\dagger}\mathbf{J}_G\right)\tilde{\mathbf{J}}_{O_i}^{\dagger}\dot{\mathbf{p}}_{O_i}\right], \tag{11}$$

where n_o is the number of obstacle avoidance points, and $\tilde{\mathbf{J}}_{O_i}^{\dagger}$ is the damped right pseudo-inverse solution for $\mathbf{J}_{O_i}$. If the desired path cannot be maintained, (11) gives up the desired velocity $\dot{\mathbf{p}}$ to ensure a collision-free path.

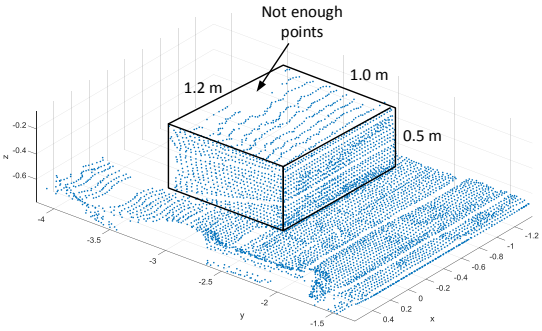


Fig. 3. The point cloud produced by the laser scanning system. Due to the position of the laser scanner, there is an area in the point cloud where the number of points is too low.

IV. Obstacle Detection System

In order to avoid manipulator collisions with obstacles, they need to be detected and identified. Identification of obstacle dimensions is difficult and especially, in the case of unforeseen obstacles, might be impossible. Therefore, in this paper we used point cloud-based approach where it is not required to identify the obstacles. Instead, in the point cloud-based methods, it is possible to use the point cloud directly. The point cloud of the obstacles was produced with a laser scanning system.

The laser scanning system consisted of a SICK LMS511 laser scanner and a mechanism that rotated the scanner along one horizontal axis. The Scanner itself provided scans in only two dimensions, and a three-dimensional point cloud was acquired by rotating the scanner. A Maxon DC motor equipped with a planetary gear rotated the scanner. There was also an absolute encoder at the end of the shaft, providing accurate measurements of the scanner position.

The angular resolution of the laser scanner (the difference between two consecutive points in one scan) was chosen to be 0.25 degrees. As the scanner was mounted in an upright position, this defined the horizontal resolution of the point cloud. The scanner was configured to send measurements constantly at a frequency of 25 Hz, and at the same time, the scanner was rotated at a speed of roughly 5 degrees/s, which resulted in a vertical resolution of 0.2 degrees. Knowing the exact rotation speed was not necessary, because the angle measurements were assumed to be accurate at all times.

The laser scanner output was the distance value for each point in a scan. Combining these values with the angle information from the laser scanner and the encoder yielded the Cartesian coordinates of each point. The collected scanner data and relevant kinematic calculations were implemented using a dSpace Microautobox. Measurements from multiple scans were then combined into a point cloud using point cloud-handling methods provided by matlab (Computer Vision System Toolbox). Next, points located outside the working area were removed from the data set, and the point cloud was downsampled so that there were only points within 3 cm of each other. The number of points in the final cloud was approximately 5500, which proved to be sufficient for this application.

Fig. 3 shows the scanned and processed point cloud. Due to the position of the laser scanner, the point cloud points are not distributed evenly across the obstacle and the floor. The areas which are closer to the laser scanner have more points than the areas which are farther. There is also a small area at the back of the obstacle where there are no points (Fig. 3).

V. Experimental Case Study

The collision avoidance control system was developed to enable autonomous operation of a redundant serial manipulator. To verify this functionality, an experimental test was carried out with a heavy-duty 4-DOF hydraulic serial manipulator, the Hiab XS 033. The Hiab XS 033 manipulator is a hydraulic multi-purpose manipulator for almost a limitless range of applications. It has three revolute joints and one prismatic joint. In the following case study, the manipulator was equipped with a grabber that included two free non-actuated joints. During the experiment, the tool center point (TCP) of the manipulator was controlled, and the grabber hung from the TCP parallel to the gravity.

During the experiment, the collision server for distance queries ran on a laptop with an Intel Core i7 processor. The server program to calculate the shortest distance between point clouds was programmed with C++/CLI and compiled to a standalone program. The average time needed to calculate the shortest distance between all collision pairs was 1.0 ms with a standard deviation of 1.7 ms. This proved that the proposed method could be used in a real-time control system where the sample time was 10 ms. However, the time required to calculate all potential collisions depends on the current configuration of the manipulator. If a different starting position and goal position are used, then the calculation time might also be different.

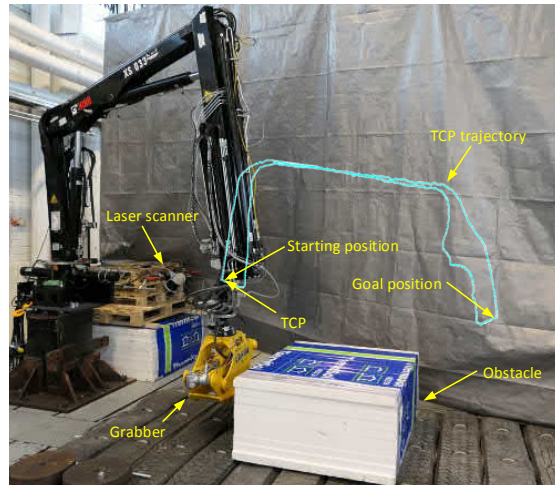


Fig. 4. The collision-free trajectory for the manipulator from the starting position to the goal position and back.

In order to validate the collision avoidance system, we conducted an experimental test. The task for the experiment was to move the grabber of the manipulator from a starting position ($\mathbf{q}_s$) to a goal position ($\mathbf{q}_g$) by avoiding collisions with obstacles. The starting position was at the left of the obstacle, and the goal position was at the other side of the obstacle (Fig. 4). With a conventional control system without collision avoidance, the grabber of the manipulator would collide with the obstacle if the grabber tried to move from the starting position to the goal position. The proposed collision avoidance control system of the manipulator together with the proposed distance query method execute the desired task without collisions by finding a collision-free path.

For the experiment, the 3D part of the grabber that could collide with the obstacle was converted into a point cloud, containing 1000 points (Fig. 2). The environment and the obstacle were scanned with the laser scanning system, containing 5500 points (Fig. 3). Thereafter, we created rules to monitor the shortest distance between possible collision pairs. If the distance between the collision pair was larger than the collision margin, then the distance had no influence on the control of the manipulator, but if the distance was smaller than the collision margin, we used (11) to modify the trajectory of the manipulator based on the calculated shortest distance between two point clouds.

Fig. 4 shows the overall experimental case study setup and how the manipulator, laser scanning system, and obstacle were located compared to each other. The trajectory of the TCP of the manipulator is also shown is this figure. This trajectory was automatically modified so that the grabber does not collide with the obstacle.

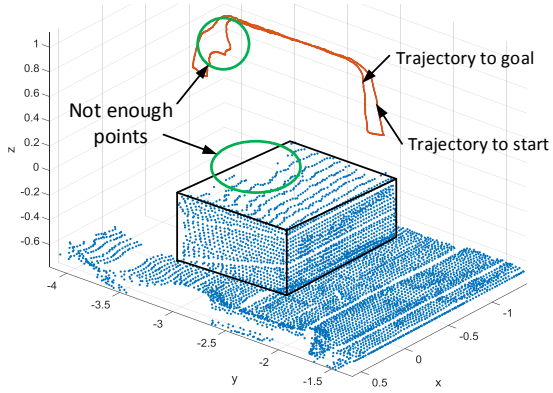


Fig. 5. The collision-free trajectory for the TCP of the manipulator from the starting position to the goal position and back with the scanned point cloud.

Fig. 5 shows the TCP trajectories of the manipulator from the starting position to the goal position and back. This figure reveals that the trajectory from the goal position to the starting position is not ideal and contains unwanted movements. These movements are caused by the lack of points at the back of the obstacle's point cloud. This area is shown in Fig. 5. If the points of the obstacle's point cloud are distributed evenly, it can be assumed that the trajectories on both sides of the obstacles would be similar.

VI. Conclusion

For autonomous manipulators to accomplish the given task without human supervision, one of the most challenging problems is avoiding collisions. A shortest distance-based collision avoidance method was used to obtain a collision-free path from the starting position to the goal position and back with a novel shortest distance query algorithm. Multiple simultaneous collision avoidance points can be used to prevent self-collisions, as well as collisions with other manipulators, and any part of the manipulator from colliding with obstacles or the environment. The proposed method can be used with a real-time control system, and the method does not require any path planning time because all decisions are made spontaneously.

The proposed method for calculating the shortest distance ensures that the shortest distance between two point cloud-type objects can be calculated in real-time. This method minimizes the calculation power needed to find the shortest distance by using a method that reduces the number of points among which the shortest distance can be found. The use of point cloud data extracted from CAD design models and from an external laser scanning system ensures that the possible collision locations are found with high accuracy. This enables the use of only small margins around obstacles to avoid collisions.

In future work, we intend to improve the collision avoidance system to cover more complex systems with a fully automatized sequence and position control with moving obstacles in the workspace of the manipulator.

Acknowledgment

This work was funded by the Doctoral School of Industry Innovations (DSII) of Tampere University of Technology. This funding is greatly appreciated. The DSII supports effective progress toward doctoral degrees, offers companies novel avenues for addressing research questions that are relevant in terms of their future competiveness, and generates new scientific knowledge.

References

[1] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[2] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Transactions on Computers*, vol. C-32, no. 2, pp. 108–120, Feb 1983.

[3] L. Zlajpah and T. Petric, *Obstacle Avoidance for Redundant Manipulators as Control Problem*. InTech, 2012. [Online]. Available: https://www.intechopen.com/books/serial-and-parallel-robot-manipulators-kinematics-dynamics-control-and-optimization/obstacle-avoidance-for-redundant-manipulators-as-a-control-problem

[4] O. Khatib, "Dynamic control of manipulator in operational space," in *Proc. 6th IFToMM World Congress on Theory of Machines and Mechanisms*, 1983, pp. 1128–1131.

[5] ——, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, Mar 1985, pp. 500–505.

[6] J. O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, Jun 1992.

[7] F. Padula and V. Perdereau, "A new pseudoinverse for manipulator collision avoidance," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 14 687 – 14 692, 2011.

[8] S. Gottschalk, M. C. Lin, and D. Manocha, "Obbtree: A hierarchical structure for rapid interference detection," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 171–180.

[9] G. van den Bergen, "Efficient collision detection of complex deformable models using aabb trees," *J. Graph. Tools*, vol. 2, no. 4, pp. 1–13, Jan 1998.

[10] D. Puiu and F. Moldoveanu, "Real-time collision avoidance for redundant manipulators," in *2011 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, May 2011, pp. 403–408.

[11] I. Lee, K. K. Lee, O. Sim, K. S. Woo, C. Buyoun, and J. H. Oh, "Collision detection system for the practical use of the humanoid robot," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov 2015, pp. 972–976.

[12] K. B. Kaldestad, S. Haddadin, R. Belder, G. Hovland, and D. A. Anisi, "Collision avoidance with potential fields based on parallel processing of 3d-point cloud data on the gpu," in *2014 IEEE Int. Conf. Robotics Autom. (ICRA)*, May 2014, pp. 3250–3257.

[13] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schrder, and R. Dillmann, "Toward humanoid manipulation in human-centred environments," *Robotics and Autonomous Systems*, vol. 56, no. 1, pp. 54 – 65, 2008, human Technologies: Know-how.

[14] S. M. Grigorescu, D. Ristić-Durrant, and A. Gräser, "Rovis: Robust machine vision for service robotic system friend," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 3574–3581.

[15] H. Samet, *An Overview of Quadtrees, Octrees, and Related Hierarchical Data Structures*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 51–68. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-83539-1-2

[16] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.

[17] T. N. E. Greville, "The pseudoinverse of a rectangular or singular matrix and its application to the solution of systems of linear equations," *SIAM Review*, vol. 1, no. 1, pp. 38–43, 1959. [Online]. Available: http://www.jstor.org/stable/2028031

Unpublished Manuscript II

Tuomo Kivelä, Jouni Mattila, Jussi Puura, and Sirpa Launis, "A Method for Task-based Optimization of a Mining Rig's Serial Manipulators with Arbitrary Topology", June 2017.

A Method for Task-based Optimization of Mining Rig's Serial Manipulators With Arbitrary Topology

Tuomo Kivelä* and Jouni Mattila

Laboratory of Automation and Hydraulics,
Tampere University of Technology, Tampere, Finland
E-mail: tuomo.kivela@tut.fi
E-mail: jouni.mattila@tut.fi
*Corresponding author

Jussi Puura and Sirpa Launis

Sandvik Mining and Construction, Tampere, Finland
E-mail: jussi.puura@sandvik.com
E-mail: sirpa.launis@sandvik.com

Abstract: Increased automation, advanced robotics, and computer control systems allow for completely new structural designs for mining rig's manipulators that are driven by the required task specifications. Compared to conventional human-operated systems, the new-generation robotic manipulators do not need an intuitive user-friendly structure. Instead, the structure for the serial robotic manipulator can be designed to be kinematically optimal. This paper introduces the kinematic synthesis method for designing an new type of robotic manipulator for future autonomous mining operations. The synthesis is treated at the kinematic level using nonlinear Levenberg-Marquardt algorithm for multivariate optimization. As a case study, a real-world underground drilling task is used as a requirement to find the optimal structure for a robotic manipulator used in tunnel construction. The obtained results encourage kinematic synthesis in applications, such as underground tunnel drilling, where the confined environment and the required task add challenges to design the most optimal robotic manipulator structure.

Keywords: Serial manipulator; Robotics; Optimization; Synthesis; Computer-aided design; Task-based; Mining; Design; Mining Rig; Underground mining

1 Introduction

A construction project is always a battle against time and costs. One has to get the project right and get it right from the start. This requires not only very deep knowledge of construction but also a range of equipment that can handle the required tasks, whenever needed. For example, tunneling construction project is very demanding task with a confined environment. Therefore, it is very important that a robotic manipulator for tunnel drilling is designed so that it is able to do all work assigned to it.

Usually most common industrial 6 degree of freedom (DOF) robotic manipulators with spherical wrist are designed for desired rated payload while maximizing robot workspace envelope. In addition, robot folded position for transportation and storage is an important design parameter to keep the design cost-effective. In more advanced design scenarios, robot total orientation workspace (TOW) can be specified as a set of range of rotation angles of the end-effector in a position inside of the bounded workspace Merlet et al. (1998).

Clearly, e.g. for tunneling construction this is not an adequate design approach. Instead, in more specific robotic manipulator design scenarios, the reachable workspace should be divided into several bounded subspaces that each have different requirements for the required range of wrist orientation angles. For example in tunnel drilling task, the drilling pattern near the center of the tunnel is more dense and requires wide range of orientation angles to be reached. On the other hand, near the boundaries of the tunnel (walls, ceiling, and ground) the drilling pattern execution requires lower range of orientation angles to be reached. Therefore, this design problem consists of well-defined robotic manipulator design requirements with a large amount of defined task positions to be reached in workspace position depended range of wrist orientation angles. Due to the complexity of the given design optimization problem, it is not obvious which robotic topology should be chosen for this given robotic manipulator design problem.

Current hydraulic manipulator mechanisms for mining machines are a mature technology, and their kinematics have been developed with a focus on the human operator maneuvering a hydraulically controlled system without any numerical control input. As the trend in mining is increased automation, computer control systems are being more and more widely used, and the requirements for robotic manipulator kinematics are different. Numeric control enables a different kind of robotic manipulator kinematics, which are not optimum for direct control by a human operator, because the joint motions related to the different trajectories are not native for the human mind. Numerically controlled robotic manipulators can accept kinematics that are more efficient in doing the job expected by the drilling customer.

New computation methods and computational power enable the use of optimization methods in manipulator kinematic synthesis. Comprehensive studies have been conducted on the kinematic synthesis for serial type robotic manipulators, in Singla et al. (2010), Tarek & Daniel (2003), Barissi & Taghirad (2008), Patel & Sobh (2015*b*), Sobh & Toundykov (2004), Vijaykumar et al. (1986), Kucuk & Bingul (2006), Shiakolas et al. (2002), Sun et al. (2007), Ouezdou et al. (2007). A classic approach to task-based dimensional synthesis is to create objective function and constraint functions and then sum them to form a cost function, which can be minimized to find the optimal solution, Singla et al. (2010), Tarek & Daniel (2003), Barissi & Taghirad (2008). Constraint functions can be weighted to address the solution in the specified direction. Although previous studies have shown how different optimization approaches can be used for task-based dimensional synthesis problems, they are limited to solve only part of the given dimensional synthesis problem. For example, Patel & Sobh (2015*b*), Sobh & Toundykov (2004), Vijaykumar et al. (1986) focus on non-redundant manipulators and some of the works concentrate to optimize serial manipulators with only three joints Barissi & Taghirad (2008), Sobh & Toundykov (2004), Kucuk & Bingul (2006). Furthermore, some of the works are restricted to optimize the manipulator structure based on end-effector position, not with position and orientation Barissi & Taghirad (2008), Shiakolas et al. (2002). Usually there is no explanation where the task points come from and the task point are selected for the optimization. In addition, the number of task points is very often limited to only few points Singla et al. (2010), Patel & Sobh (2015*b*),

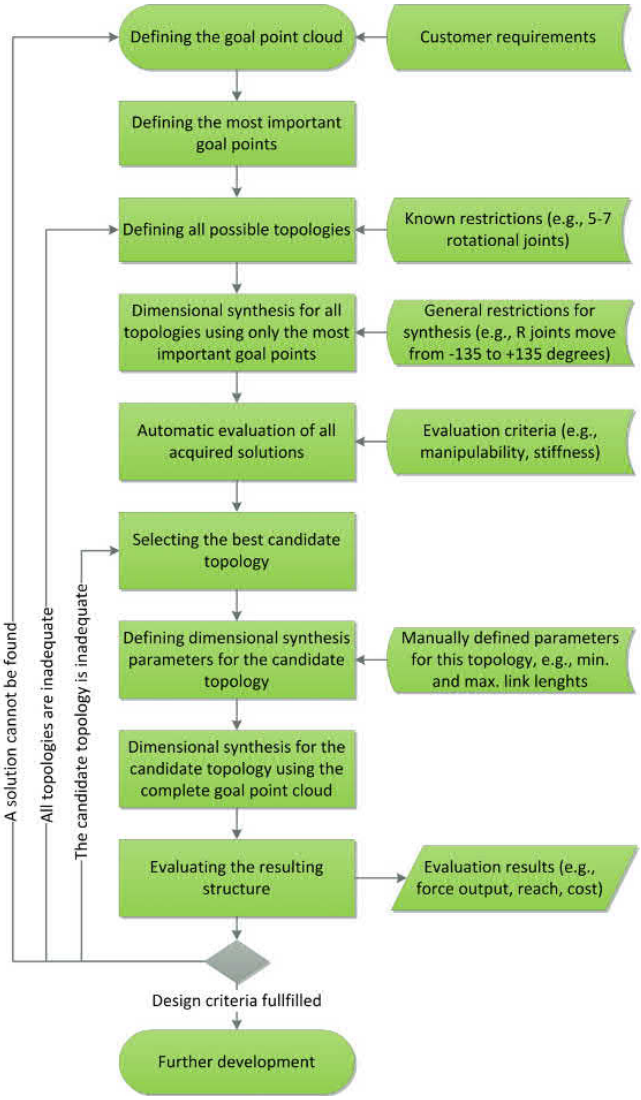


Figure 1: Simplified synthesis process of a robotic manipulator

Shiakolas et al. (2002). As a result, no comprehensive real-world scale study appears to exist.

In spite of these previous studies, the use of task-based optimization for an arbitrary manipulator structure with practical relevance has remained unclear. Hence, additional

studies of the task-based optimization for an arbitrary manipulator to find the optimal values for the design parameters, the base frame, and the joint values are needed. This means that there are several hundreds of parameters in total that need to be optimized. Therefore, the aim of this paper is to extend the above mentioned works to cover a more general serial manipulator optimization problem with practical relevance. The proposed optimization method can handle non-redundant and redundant manipulators. Furthermore, it can handle both revolute and prismatic joints and n-dimensional task space including end-effector position and orientation. The proposed method can also handle large number of task points.

In this paper, we propose an optimization method that takes into account not only the desired task points but also the practical design constrains for the given task and most important constraints such as available confined working envelope inside tunnel and joint ranges. In the generic approach proposed in this paper, due to practical reasons such as resulting robot design cost-effectiveness, set of robot topologies is limited to serial-type manipulators with a maximum number of seven actuators. Studied tunnel drilling pattern consist of multiple task points where several real drill plans are combined into a single set of task points. This real customer application requires long reach manipulator (10-15 meters) due to the desired tunnel size. Therefore, manipulator topologies that have at least one prismatic joint are interesting, however, it is not clear if six joints are better than seven joints. The proposed method optimizes the set of robot topologies and ranks them according to the selected performance measure. Our results indicate that this kind of method is effective in finding the optimal structure and dimensions for the serial type manipulator from the task specifications based on the selected performance measure.

We have organized the rest of this paper in the following way. In Section 2, the kinematic synthesis process is described from the task specification perspective. In Section 3, the manipulator kinematics are derived. In Section 4, the dimensional synthesis method based on selectively damped least squares is described with optimization constraints and initial design parameter values. The proposed kinematic synthesis method is applied to the case study with a discussion of the results in Section 5, and conclusions are presented in Section 6.

2 Kinematic Synthesis

In general, kinematic synthesis can be divided into two separate parts: topology synthesis and dimensional synthesis. Topology synthesis involves determining the structure of a device, whereas dimensional synthesis is devoted to determining the dimensions of a device. This section presents an overview of the complete process flow from the user requirements to an acceptable mechanical structure with a defined number of links of known lengths and connected by joints of known types.

The flow of the proposed synthesis process is depicted in Fig. 1. The synthesis process is started by defining the task that the manipulator structure should be able to fulfill. This step is preferably performed with the customer to maximize the business benefit the machine will bring him or her. For example, if the machine must be carefully repositioned before starting a new work cycle, it may be desirable to reach a certain number of goal points before repositioning is needed. In general, the task consists of several postures that need to be reached by the manipulator with an attached tool. In addition to locations and orientations given with the required degrees of freedom, tasks may include other required attributes, such

as required force output and structural stiffness. Although the complete task may contain a large number of points, a subset of the most important or descriptive task points should also be defined. This subset of goal points will be used during the initial evaluations to speed up the process.

Once a suitable subset of the task has been defined, the next step is to define all possible topologies. This is referred to as topology synthesis. Topology in this context means the number of links and the number, types, and ordering of the joints that connect the links to each other. To limit the infinite number of topologies imaginable, known restrictions should be taken into account before or during topology synthesis. For example, it may be desirable to consider only open chain kinematic structures, as this simplifies the evaluation considerably. It may also be desirable to limit the number of types of joints that should be used to reach the desired outcome, the type and axis direction of the first joint, or any other parameter that would otherwise be free to vary.

Once the list of possible topologies has been defined, it is time to perform a dimensional synthesis for all topologies using a subset of the task (most important task points that define the limits of the required reach, as well as some typical operating points). Dimensional synthesis is used to calculate the required link lengths, given the goal points with respect to the machine's base coordinates and the topology of the structure. At this point in the process, constraints that are common to all topologies should be considered in the synthesis algorithm. For example, it may be desirable to limit the motion range of all rotational joints to some value and assign minimum lengths for all links. If environmental entities, such as obstacles or walls, are to be included in the evaluation, they should be included in the synthesis algorithm at this stage of the process.

After the first round of dimensional synthesis has been performed, all solutions (topologies) that were able to reach all of the desired goal points should be organized into an ordered list for further processing. To this end, the solutions (topology and link lengths) should be evaluated against predefined criteria in each of the goal points. The list could be ordered, for example according to a weighted combination of measures of manipulability and stiffness.

After dimensional synthesis has been performed for all topologies and feasible solutions (those able to reach all task points) have been evaluated and ranked, the designer should select a group of the best candidate solutions from the ranked list for further synthesis. The next steps in the process are to assign dimensional synthesis parameters specific to the given topologies (e.g., length limits for specific links) and to execute the dimensional synthesis routine for those topologies by using either the complete task or only the most important points. An evaluation of the resulting structure should be automatically generated by the design software used. The designer can then choose to change some of the specific synthesis parameters and run the evaluation again.

If the current solution cannot adequately fulfill the design criteria, the designer should move on to evaluate the next best solution. If all of the solutions found in the automatically ranked list fail to provide feasible structures, the designer can either return to modify the restrictions for generating the original list of topologies or evaluate the need to revisit customer requirements based on the results of the synthesis process. Once a structure that fulfills the design criteria has been found, kinematic synthesis can be considered complete, and the solution(s) may be subjected to further modeling and design.

This process assumes that the initially generated number of topologies is high, that the task contains a large number of points, and that the kinematic synthesis algorithm is resource-intensive. If these assumptions can be relaxed, the complete task may be used at

all times, and a preliminary round of kinematic synthesis will not be required (re-runs with different optimization parameters may still be desirable). This somewhat simplified process is shown in Fig. 1.

3 Manipulator Kinematics

The Denavit-Hartenberg (DH) convention is a well known way to describe the kinematic and geometric structure of a robotic manipulator. This method defines the relative position and orientation of two consecutive links. There are two different ways to represent DH parameters: classic DH parameters Sciavicco & Siciliano (2000) and modified DH parameters Craig (1989). The difference between these two representations is the location of the coordinate system attachment to the links and the order of the performed transformations. In this paper we use the modified DH notation.

Combining the position and the orientation of an end-effector describes a manipulator posture

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \phi \end{bmatrix}, \tag{1}$$

where $\mathbf{p}$ describes the end-effector position and ϕ its orientation. Now, the relation between a joint value vector $\mathbf{q} \in \mathbb{R}^n$ and the end-effector posture vector $\mathbf{x} \in \mathbb{R}^m$ is given by

$$\mathbf{x} = f(\mathbf{q}), \tag{2}$$

where $f(\cdot)$ is a nonlinear direct kinematics function of the manipulator. The geometric Jacobian matrix $\mathbf{J}$ relates the joint velocity vector $\dot{\mathbf{q}}$ to end-effector velocity vector $\dot{\mathbf{x}}$ through the mapping

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}} \tag{3}$$

where $\dot{\mathbf{p}}$ and $\boldsymbol{\omega}$ represent end-effector linear and angular velocities, respectively. This allows us to solve the inverse kinematics by an inversion of the Jacobian matrix

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\,\dot{\mathbf{x}}. \tag{4}$$

Equation (4) is valid only for manipulators that have the same number of dom (n) and number of operational variables necessary to specify a given task (r), i.e., manipulators with $n = r$. When the manipulator is redundant, $r < n$, the Jacobian matrix has more columns than rows, and an infinite number of solutions exist for Eq. (3). Therefore, a direct Jacobian matrix inverse cannot be done. If the Jacobian matrix is a non-square matrix, then a suitable pseudo-inverse can be adopted Sciavicco & Siciliano (2000):

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}(\mathbf{q})\,\dot{\mathbf{x}}, \tag{5}$$

and the matrix,

$$\mathbf{J}^{\dagger} = \mathbf{J}^{T}\left(\mathbf{J}\mathbf{J}^{T}\right)^{-1}, \tag{6}$$

is the right pseudo-inverse of $\mathbf{J}(\mathbf{q})$.

The pseudo-inverse method sets the value of $\dot{\mathbf{q}}$ equal to $\mathbf{J}^{\dagger}$ (Eq. (5)). This pseudo-inverse method gives the best possible solution to Eq. (3) in the sense of least squares. Unfortunately, the pseudo-inverse tends to have stability problems in the neighborhood of singularities. At a singularity, the Jacobian matrix no longer has a full row rank. If the configuration is close to a singularity, then the pseudo-inverse method will lead to very large changes in the joint angles, even for small movements in the target position.

The damped least squares method avoids many of the pseudo-inverse method's problems with singularities and can give a numerically stable method of selecting $\dot{\mathbf{q}}$. Rather than just finding the minimum vector $\dot{\mathbf{q}}$ that gives the best solution, the damped least squares method finds the value of $\dot{\mathbf{q}}$ that minimizes the quantity

$$\|\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}\| + \lambda^2 \|\dot{\mathbf{q}}\|^2, \tag{7}$$

where λ is a non-zero damping constant. Now, the damped least squares solution is equal to

$$\dot{\mathbf{q}} = \mathbf{J}^T \left(\mathbf{J}\mathbf{J}^T + \lambda^2 \mathbf{I}\right)^{-1} \dot{\mathbf{x}}. \tag{8}$$

The damping constant depends on the details of the multibody and the target positions and must be chosen carefully to make Eq. (8) numerically stable. If $\lambda = 0$, Eq. (8) becomes identical to Eq. (5), which is ill-conditioned near a singularity. The damping constant should be large enough so that the solutions for $\dot{\mathbf{q}}$ are well-behaved near singularities, but if it is too large, then the convergence rate is slow. A number of methods have been proposed for selecting damping constant dynamically, and these methods are called selectively damped least squares (SDLS) methods Buss & Kim (2004). The damping constant of SDLS depends on the current configuration of the manipulator and the relative posture of the end-effector and the task posture.

Singular value decomposition (SVD) Golub & Kahan (1965) is a useful method for analyzing the damped least squares method. This method can also be used to select a damping coefficient for the SDLS method. According to SVD theory, with some matrix $\mathbf{J}(m \times n)$ there exist orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ of dimensions $m \times n$ and $n \times n$, respectively, so that

$$\mathbf{J} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{9}$$

where $\mathbf{D}$ is the $n \times n$ diagonal matrix formed by the singular values of $\mathbf{J}$, which are arranged in descending order, i.e., $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_m \geq 0$.

Dampening the pseudo-inverse method is necessary only near a singular region. Therefore, the damping factor can be defined so that damping is applied only when entering a singular region. The singular region can be defined based on of the estimate of the smallest singular value of a Jacobian matrix via SVD. The damping factor can be defined based on the smallest singular value, as in Chiaverini et al. (1994):

$$\lambda^2 = \begin{cases} 0, & \text{if } \sigma_m \geq \epsilon \\ \left(1 - \left(\frac{\sigma_m}{\epsilon}\right)^2\right) \lambda_{max}^2, & \text{if } \sigma_m < \epsilon, \end{cases} \tag{10}$$

where σ_m is the smallest singular value of the Jacobian matrix, ϵ defines the width of the singular region in which the damping factor gets a non-zero value, and λ_{max} is the maximum damping factor value.

The pseudo-inverse has a self-motion property that can be used to modify the inverse kinematics solution. The matrix $\left(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}\right)$ performs a projection onto the null-space of $\mathbf{J}$. The pseudo-inverse solution in Eq. (5) can be rewritten as Baillieul (1985)

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + \left(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}\right)\dot{\mathbf{q}}_0, \tag{11}$$

where $\dot{\mathbf{q}}_0$ is an arbitrary joint-space velocity. The null-space projection ensures that for all vectors $\dot{\mathbf{q}}_0$

$$\mathbf{J}\left(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}\right)\dot{\mathbf{q}}_0 = \mathbf{0}. \tag{12}$$

By selecting suitable joint-space velocity, one can force internal motions of the manipulator and therefore avoid constraints violations.

4 Dimensional Synthesis

Dimensional synthesis use nonlinear Levenberg-Marquardt (LM) method to solve the sythesis problem. The LM algorithm is an iterative technique that finds a minimum of a multivariate function. The LM interpolates between the Gauss-Newton method and the steepest descent gradient method. When considering the direct kinematics equation in Eq. (2), it can be rewritten in the form where all DH parameters are emphasized. Then, Eq. (2) becomes

$$\mathbf{x} = f\left(\mathbf{a}, \mathbf{d}, \boldsymbol{\alpha}, \boldsymbol{\theta}\right), \tag{13}$$

where $\mathbf{a} = \left[a_1 \dots a_n\right]^T$, $\mathbf{d} = \left[d_1 \dots d_n\right]^T$, $\boldsymbol{\alpha} = \left[\alpha_1 \dots \alpha_n\right]^T$, and $\boldsymbol{\theta} = \left[\theta_1 \dots \theta_n\right]^T$ are DH parameters for the whole structure.

Let $\mathbf{x}_d$ be the posture of the end-effector defined by a task, and let $\mathbf{x}$ be the current posture that can be computed with the DH parameters. The deviation $\Delta\mathbf{x} = \mathbf{x}_d - \mathbf{x}$ gives a measure of accuracy at the given posture. On the assumption of small deviations, at the first approximation, it is possible to derive the relationship

$$\Delta\mathbf{x} = \frac{\partial f}{\partial \mathbf{a}}\Delta\mathbf{a} + \frac{\partial f}{\partial \mathbf{d}}\Delta\mathbf{d} + \frac{\partial f}{\partial \boldsymbol{\alpha}}\Delta\boldsymbol{\alpha} + \frac{\partial f}{\partial \boldsymbol{\theta}}\Delta\boldsymbol{\theta}. \tag{14}$$

The deviation $\Delta\mathbf{x}$ for the end-effector of the manipulator is created using two proportional controllers, one for the position and the second for the orientation. The position controller is described as follows:

$$\dot{\mathbf{p}} = k_v\left(\mathbf{p}_d - \mathbf{p}\right), \tag{15}$$

where $\mathbf{p}_d$ is the goal position of the end-effector and k_v is the position control coefficient.

In order to devise the deviation for the end-effector orientation, a suitable orientation error must be defined. The end-effector orientation error depends on the choice of the orientation description. Direct computation of the orientation error requires the extraction

of the Euler angles from the end-effector rotation matrix, which suffers from representation singularities. Therefore, the orientation error is calculated based on unit quaternion representation. The drawbacks of the Euler parameters can be overcome by the unit quaternion Sciavicco & Siciliano (2000). Let $\boldsymbol{\sigma}_d = \{\eta_d, \boldsymbol{\epsilon}_d\}$ and $\boldsymbol{\sigma} = \{\eta, \boldsymbol{\epsilon}\}$ represent the quaternions associated with $\mathbf{R}_d$ and $\mathbf{R}$, respectively. The orientation error is defined as

$$\mathbf{e}_O = \eta\boldsymbol{\epsilon}_d - \eta_d\boldsymbol{\epsilon} - \mathbf{S}(\boldsymbol{\epsilon}_d)\boldsymbol{\epsilon}, \tag{16}$$

where $\mathbf{e}_O$ is the orientation error and $\mathbf{S}(\boldsymbol{\epsilon}_d)$ is the skew-symmetric matrix for the $\boldsymbol{\epsilon}_d$. The end-effector's orientation controller can be described as

$$\boldsymbol{\omega} = k_w\mathbf{e}_O, \tag{17}$$

where k_w is the angular velocity control coefficient. Using Eqs. (3), (15), and (17), the manipulator's end-effector deviation takes the following form:

$$\Delta\mathbf{x} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} k_v\left(\mathbf{p}_d - \mathbf{p}\right) \\ k_w\mathbf{e}_O \end{bmatrix}. \tag{18}$$

Equation (14) can be expressed in a more compact form as

$$\Delta\mathbf{x} = \boldsymbol{\Phi}\Delta\boldsymbol{\zeta}, \tag{19}$$

where

$$\boldsymbol{\Phi} = \begin{bmatrix} \dfrac{\partial\mathbf{f}}{\partial\mathbf{a}} & \dfrac{\partial\mathbf{f}}{\partial\mathbf{d}} & \dfrac{\partial\mathbf{f}}{\partial\boldsymbol{\alpha}} & \dfrac{\partial\mathbf{f}}{\partial\boldsymbol{\theta}} \end{bmatrix}, \tag{20}$$

and $\boldsymbol{\Phi} \in \mathbb{R}^{m\times 4n}$, and m is the number of operational space variables. The DH parameters can be grouped into the vector

$$\boldsymbol{\zeta} = \begin{bmatrix} \mathbf{a}^T & \mathbf{d}^T & \boldsymbol{\alpha}^T & \boldsymbol{\theta}^T \end{bmatrix}, \tag{21}$$

where $\boldsymbol{\zeta} \in \mathbb{R}^{4n\times 1}$. Now the parameter variations with respect to the current parameter values can be expressed as

$$\Delta\boldsymbol{\zeta} = \boldsymbol{\zeta}_d - \boldsymbol{\zeta}, \tag{22}$$

where $\boldsymbol{\zeta}$ denotes the current parameters, and $\boldsymbol{\zeta}_d$ denotes the parameters required by the task.

When the number of task locations is k, it yields the following

$$\Delta\bar{\mathbf{x}} = \begin{bmatrix} \Delta\mathbf{x}_1 \\ \vdots \\ \Delta\mathbf{x}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_k \end{bmatrix}\Delta\boldsymbol{\zeta} = \bar{\boldsymbol{\Phi}}\Delta\boldsymbol{\zeta}. \tag{23}$$

The solution for $\Delta\boldsymbol{\zeta}$ can be found by using Eq. (8):

$$\Delta\boldsymbol{\zeta} = \bar{\boldsymbol{\Phi}}^T\left(\bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\Phi}}^T + \lambda^2\mathbf{I}\right)^{-1}\Delta\bar{\mathbf{x}}. \tag{24}$$

Since this method is an iterative method, the procedure is iterated until the change in parameter values converges at a given threshold. At each iteration, the pseudo-inverse matrix is updated with the parameter estimates ($\boldsymbol{\zeta} = \boldsymbol{\zeta} + \Delta\boldsymbol{\zeta}$), as well as the posture errors via direct kinematics.

4.1 Design Parameter Constraints

Several design constraints limit the manipulator structure, and these constraints must be taken into account in the optimization process. One of the most important constraints is design parameter or DH parameter constraints. In practice, all DH parameters are bounded within their limits. The reason for the limits can be anything from mechanical design limitations to cost-reduction issues. Several studies have investigated DH parameter constraints, for example Klein & Huang (1983), Chan & Dubey (1995). These researchers used performance criteria to avoid the joint limit. In their work, they proposed a performance criterion to limit parameter movement outside the parameter's center position. The performance criterion gives higher weight to the parameters near their limits and goes to infinity at the parameter bounds. This performance criterion is not appropriate for our case because the whole range of each parameter and joint must be freely usable. Therefore, a dynamic weighting matrices method Schinstock et al. (1994) is used to limit the joint and parameter values. The weighting matrix remains one within the range of the parameters and rises rapidly just before the limits. This ensures that almost the whole range of the parameter range is always usable.

Considering the following reformulation of Eq. (23),

$$\mathbf{W}_x \Delta\bar{\mathbf{x}} = \mathbf{W}_x \bar{\mathbf{\Phi}} \mathbf{W}_q \mathbf{W}_q^{-1} \Delta\boldsymbol{\zeta}, \tag{25}$$

where the weighting matrices are defined by

$$\begin{aligned} \mathbf{W}_x &\equiv diag\left[w_{x_1} \ldots w_{x_m}\right], w_{x_i} > 0 \\ \mathbf{W}_q &\equiv diag\left[w_{q_1} \ldots w_{q_n}\right], w_{q_i} > 0. \end{aligned} \tag{26}$$

Solving Eq. (25) using damped least squares results in a solution, given by

$$\Delta\boldsymbol{\zeta} = \mathbf{W}_q \bar{\mathbf{\Phi}}_w^{\dagger} \Delta\bar{\mathbf{x}}_w, \tag{27}$$

where

$$\begin{aligned} \Delta\bar{\mathbf{x}}_w &\equiv \mathbf{W}_x \Delta\bar{\mathbf{x}}, \\ \bar{\mathbf{\Phi}}_w^{\dagger} &\equiv \bar{\mathbf{\Phi}}_w^T \left(\bar{\mathbf{\Phi}}_w \bar{\mathbf{\Phi}}_w^T + \lambda^2 \mathbf{I}\right)^{-1}, \\ \bar{\mathbf{\Phi}}_w &\equiv \mathbf{W}_x \bar{\mathbf{\Phi}} \mathbf{W}_q. \end{aligned} \tag{28}$$

By choosing the weighting matrices correctly, it is possible to modify the solution of the damped least squares method. This feature is used to implement the joint and parameter bounds.

4.2 Workspace Constraints

Other constraints that affect manipulator optimization are workspace constraints. For example, in the case study, the manipulator is located in an underground tunnel, which means that there are walls, a floor, and a ceiling around the manipulator. Therefore, it is necessary to take a workspace limitations into account. The null-space motion concept (Section 3) is used to avoid manipulator collisions with a workspace. The workspace can be handled as an external object, and collisions of the manipulator's internal structure with external

objects are represented. An artificial potential field (APF) is used to create a repulsive force between the manipulator and obstacles that pushes the manipulator away from the obstacles. The APF method as a collision avoidance method is widely used in several studies Khatib (1985), Warren (1989).

The collision-free path planner is based on Eq. (11) and the approach proposed by Maciejewski & Klein (1985). The primary goal is to reach the task pose with the manipulator's end-effector; the secondary goal is to avoid collisions. In Maciejewski & Klein (1985), the authors presented a method for avoiding collisions with one obstacle avoidance point. In the present paper, this approach is extended to cover multiple obstacle avoidance points. The primary goal is described by Eq. (3) and secondary goals is described by the equation

$$\mathbf{J}_O \dot{\mathbf{q}} = \dot{\mathbf{x}}_O, \tag{29}$$

where $\mathbf{J}_O$ is the obstacle avoidance point Jacobian, and $\dot{\mathbf{x}}_O$ is the obstacle avoidance point velocity.

The obstacle avoidance point velocity can be calculated by using the shortest distance between the manipulator and the workspace limits and by using the artificial potential field proposed by Khatib (1985):

$$\mathbf{U}_O = \begin{cases} \frac{1}{2}\mu\left(\frac{1}{\rho} - \frac{1}{\rho_O}\right), & \text{if } \rho \leq \rho_O \\ 0, & \text{if } \rho > \rho_O, \end{cases} \tag{30}$$

where ρ_O is the limit for the collision avoidance, ρ is the distance between two obstacles, and μ is a scalar coefficient. Now the $\dot{\mathbf{x}}_O$ can be formulated with Eq. (30),

$$\dot{\mathbf{x}}_O = \begin{cases} \mu\left(\frac{1}{\rho} - \frac{1}{\rho_O}\right)\frac{1}{\rho^2}\frac{\delta\rho}{\delta\dot{\mathbf{x}}_O}, & \text{if } \rho \leq \rho_O \\ 0, & \text{if } \rho > \rho_O, \end{cases} \tag{31}$$

where $\frac{\delta\rho}{\delta\dot{\mathbf{x}}_O}$ is the direction of the collision line.

The secondary goal solution for $\dot{\mathbf{q}}$ is used as an arbitrary vector $\dot{\mathbf{q}}_0$ to modify the solution of Eq. (11). By combining Eq. (11) and Eq. (29) with multiple workspace limits avoidance points, the following solution can be derived:

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}} + \left(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}\right) \mathbf{J}_O^\dagger \dot{\mathbf{x}}_O, \tag{32}$$

where n_o is the number of workspace limits avoidance points, and $\mathbf{J}_O^\dagger$ is the damped right-pseudo inverse solution for $\mathbf{J}_O$.

4.3 Parameter Update Function

In the previous sections, the optimality criterion and constraints were formulated separately. In order to optimize a given manipulator structure, the final parameter update function must be created. This can be done by adding the optimality criterion and different constraints together. Therefore, combining equations (27) and (32), which is extended to cover whole problem, the final parameter update function is

$$\Delta\boldsymbol{\zeta} = \mathbf{W}_q \bar{\boldsymbol{\Phi}}_w^\dagger \Delta\bar{\mathbf{x}}_w + \left(\mathbf{I} - \mathbf{W}_q \bar{\boldsymbol{\Phi}}_w^\dagger \bar{\boldsymbol{\Phi}}_w\right) \tilde{\mathbf{J}}_O^\dagger \tilde{\dot{x}}_O. \tag{33}$$

4.4 Initial Joints and Design Parameters Values

Finding a solution to the LM-based optimization method depends in part on the initial values given. Therefore, it is very important to choose as accurate initial values of the problem as possible. Poorly selected initial values may mean the problem does not converge and is therefore placed in the locale minimum where the problem cannot be resolved. This work uses a method based on random number selection of the initial values. First, we specify a certain number of design parameter sets, based on beta distribution. Thereafter, the joint values for each set of parameters are solved with inverse kinematics for each task point. The parameter set that gets closest to each task point is selected as the initial value.

4.5 Performance Measure

Performance measures are required to describe a behavior of a manipulator. A large number of different performance measures have been studied since the early days of robotics Patel & Sobh (2015*a*). All performance indexes can be categorized based on their scope, performance characteristic, or application. Which performance measure should be used to compare manipulators depends on the application. For example, for the optimized manipulator in this paper, it is not worth using a performance index that measures the workspace of the manipulator, because the manipulator has been optimized to fulfill a specific task.

The condition number is a local kinematic conditioning index, which is a measure of the degree of ill-conditioning of the manipulator or a measure of kinematic isotropy of the Jacobian Angeles & López-Cajún (1992). The condition number is defined as the ratio of the maximum and minimum singular values of the Jacobian Salisbury & Craig (1982).

$$\kappa = \frac{\sigma_{max}}{\sigma_{min}}. \tag{34}$$

The condition number does not have an upper bound, $\kappa \in [1, \infty]$. Therefore, the condition number's inverse, known as the local conditioning index (LCI), is more commonly used. The LCI is bounded, $LCI = [0, 1]$.

$$LCI = \frac{1}{\kappa}. \tag{35}$$

When the Jacobian is non-homogeneity due to different units used to represent the link lengths and joint angles, the condition number does not accurately represent ill-conditioning of the manipulator's Jacobian Patel & Sobh (2015*a*). There are many scaling techniques that have been used to treat the non-homogeneity of the Jacobian Stocco et al. (1999). In this paper, the method proposed by Lee et al. (2001) is used to scale the Jacobian matrix. The scaling method is based on the nominal link ($_{NL}$) whose length is defined as the distance from the base frame to tool center point. The scaled form of the Jacobian that is used to calculate the LCI gets form

$$\tilde{\mathbf{J}}_A = \mathbf{S}\mathbf{J}_A, \tag{36}$$

where

$$\mathbf{S} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{l}_{NL} \end{bmatrix}. \tag{37}$$

Local performance indices are performance metrics that depend on the posture of the manipulator. Therefore, local indices are not usually used to compare the structures of the manipulators. Instead, global performance indices are posture-independent indices, and therefore, they can be used to compare the structure and behavior of the manipulators that perform the same task.

4.5.1 Global Conditioning Index

The global conditioning index (GCI) is based on the condition number, and the LCI can be extended to the GCI. The GCI proposed by Gosselin & Angeles (1991) calculates the distribution of the condition number over the entire workspace. Later, Puglisi et al. (2012) proposed a simpler discrete formulation of the GCI that is used in this paper:

$$GCI = \frac{1}{n_{tp}} \sum_{i=1}^{n_{tp}} LCI(i), \tag{38}$$

where n_{tp} are the manipulator workspace nodes or task points. If the GCI approaches one, the manipulator is said to have, for example, better possibilities of generating output forces from input torques and Cartesian velocity from joint velocity.

4.5.2 Kinematic Conditioning Index

The kinematic conditioning index (KCI) is another performance measure based on the condition number proposed by Angeles & López-Cajún (1992). The KCI is defined as:

$$KCI = \frac{1}{\kappa_{min}}, \tag{39}$$

where κ_{min} is the minimum condition number. Therefore, the KCI is a global index that shows the worst performance of the manipulator in the task space. The $KCI \in [0, 1]$, where 1 means that the manipulator is an isotropic manipulator since all the singular values of the Jacobian are identical. The KCI with zero means that there is at least one singular posture in the task space.

5 Case Study of Manipulator Design

Current boom mechanisms are out-of-date technology, when talking about mining machine booms, and their kinematics have been developed with a focus on a human operator maneuvering a hydraulically controlled system. As the trend is increasing automation, the requirements for manipulator kinematics are different. Numeric control enables a different kind of boom kinematics, which were not optimum for direct control by a human operator, because the joint motions related to the different trajectories are not native to the human mind. Therefore, it is necessary to design a completely new manipulator for autonomous control.

It is extremely difficult to optimize the number and type of joints for the manipulator. Instead of trying to automatically optimize the most suitable topology, we will optimize several different topologies in terms of link lengths. Thereafter, suitable performance measures can be used to rank the optimized structures and select the most attractive one.

Figure 2: A typical mining rig with a drilling manipulator attached (Photo: Sandvik).

Table 1 Different serial manipulator topologies to be optimized. The fixed column states whether the twist angles are locked or free to change.

Configuration	Manipulator	Topology	Fixed
1	Man1	RRPRRRP	Yes
2	Man2	RRPRRRP	Yes
3	Man3	RRPRRR	Yes
4	Man4	RRPRRR	Yes
5	Man5	PRRRRR	Yes
6	Man6	RRRRRRR	Yes

As a case study, the real-world underground tunneling drilling pattern of holes is used to find the optimal structure for a manipulator to drill these holes. The objective is to optimize the manipulator structure by modifying the manipulator link lengths, as well as the position of the coordinate system of the manipulator base in order to satisfy a number of design constraints. The drilling pattern consists of 87 task points (n_{tp}) (Fig. 3). The aim should be to reach all these positions with the correct orientation of the tool. There are six ($n_r = 6$) operational space variables, three variables for the positions of the drilling holes and three variables for the orientations of the drilling holes. In general, the drilling operation requires five n_r parameters to accomplish the operation. Drilling tool has a spindle that is axially symmetric, thus the orientation of the end-effector around this axis is irrelevant for the task at hand. However, the drilling tool itself is not symmetric around the drilling axis. Therefore, is some end-effector poses it is necessary to be able to rotate the tool also around the drilling axis. For example, near tunnel walls it might be necessary for the end-effector to get as close as possible to the wall to be able to achieve the required drilling pose. Therefore, sin n_r parameters are considered in this case study.

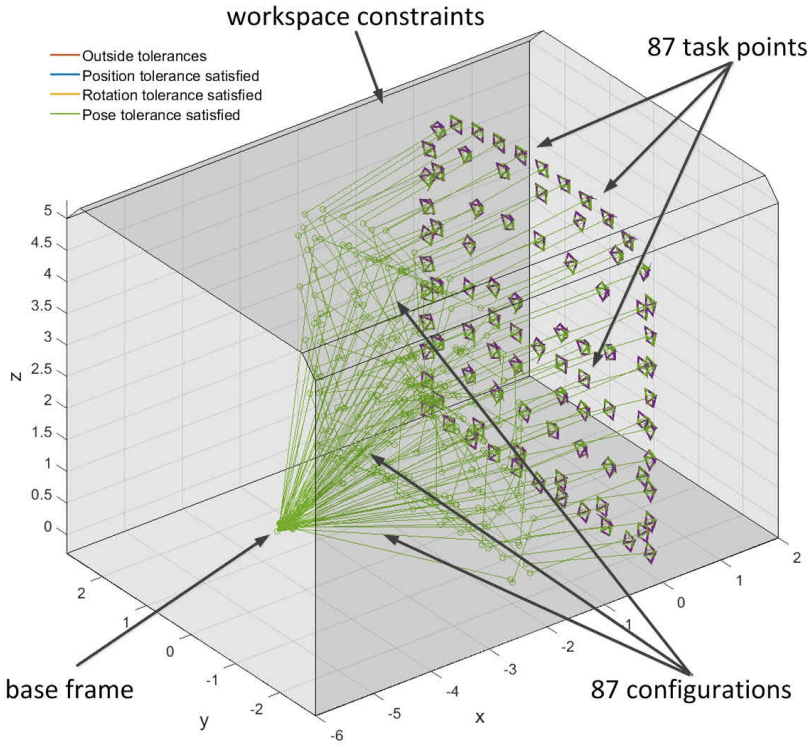
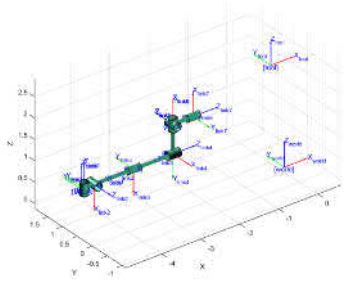


Figure 3: Final manipulator configurations for each task point

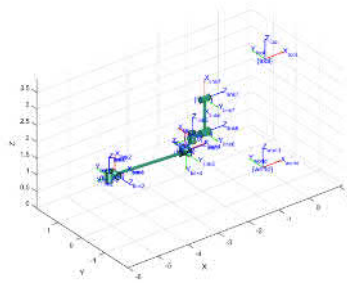
Figure 2 shows one mining rig with a drilling manipulator attached. This manipulator has seven actuators for tunnel face drilling purposes. The topology of this manipulator is one of the studied topologies (configuration 2). In this case study, we optimize the manipulator shown in Fig. 2 together with several other randomly selected topologies and find the most attractive manipulator structure and dimensions for the case task.

There are six different topologies with tool to be optimized for the drilling task, Table 1 and Fig. 4. In general, it is possible to automate the kinematic synthesis process so that all possible topologies are generated automatically and then optimized. For this case study, six topologies were selected randomly to show how the proposed optimization method works. Each topology has several design parameters (n_{dp}) to be optimized and that can be changed during optimization. In addition, the manipulator base frame (n_{bf}) is freely movable. The optimization task is to find the appropriate values for the design parameters, the base frame, and the joint values (n_{fj}). This means that there are several hundreds of ($n_{np} = n_{dp} + n_{bf} + n_{fj} * n_{tp}$) parameters ($\boldsymbol{\zeta}_o \in \mathbb{R}^{n_{np} \times 1}$) in total to be optimized.

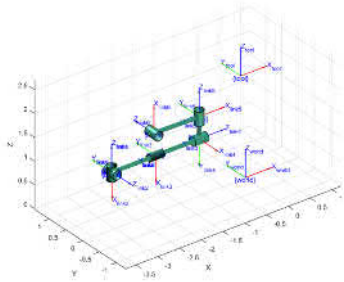
Optimization can be performed in such a way that only the link lengths between the joints are optimized or so that the twist angle between two joints can change. If the twist



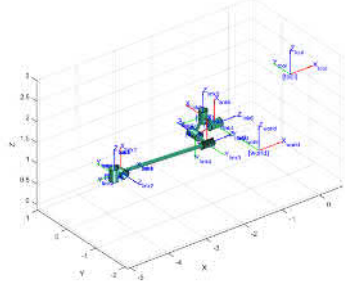
(a) Manipulator configuration 1



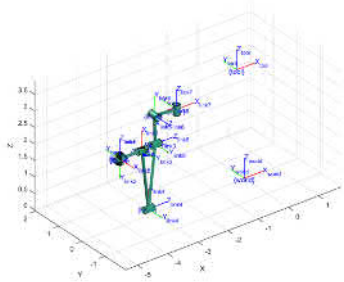
(b) Manipulator configuration 2



(c) Manipulator configuration 3



(d) Manipulator configuration 4



(e) Manipulator configuration 5

Figure 4: Different manipulator configurations with optimized parameters

angles are free to change, it may result in a very complex structure, which is mathematically optimal but can be difficult to put into practice. In the case of a heavy machine, the joints are often implemented in hydraulics, which limits the placement of the joints. Therefore, it is very important to keep in mind whether to allow a change in the twist angle between two joints or just to allow a change in the link lengths between joints. In the case study,

Table 2 Manipulator performance indices for optimized structures. - means that there is no solution for the problem.

Configuration	GCI [0..1]	KCI [0..1]	Mean [0..1]
1	0.1043	0.0744	0.0894
2	0.1158	0.0858	0.1008
3	0.1059	0.0652	0.0856
4	0.1006	0.0840	0.0923
5	0.0741	0.0058	0.0400
6	-	-	-

Table 3 The optimized DH parameters for the manipulator 1. Joint variables are bolded

frame	a_{i-1}	d_i	α_{i-1}	θ_i
1	a_0	0	0	$\boldsymbol{\theta}_1$
2	a_1	0	$\pi/2$	$\boldsymbol{\theta}_2 - \pi/2$
3	0	$d_{3p} + \mathbf{d}_{3j}$	$-\pi/2$	0
4	0	d_4	0	$\boldsymbol{\theta}_4 - \pi/2$
5	0	d_5	$\pi/2$	$\boldsymbol{\theta}_5 + \pi/2$
6	0	0	$-\pi/2$	$\boldsymbol{\theta}_6 - \pi/2$
7	0	$d_{7p} + \mathbf{d}_{7j}$	$-\pi/2$	0

Table 4 The optimized DH parameters for the manipulator 2. Joint variables are bolded

frame	a_{i-1}	d_i	α_{i-1}	θ_i
1	0	0	0	$\boldsymbol{\theta}_1$
2	a_1	d_2	$\pi/2$	$\boldsymbol{\theta}_2 + \pi/2$
3	a_2	$d_{3p} + \mathbf{d}_{3j}$	$\pi/2$	0
4	a_3	d_4	$\pi/2$	$\boldsymbol{\theta}_4 + \pi/2$
5	a_4	d_5	$\pi/2$	$\boldsymbol{\theta}_5 + \pi/2$
6	0	d_6	$\pi/2$	$\boldsymbol{\theta}_6 + \pi/2$
7	a_6	$\mathbf{d}_7$	0	0

this paper is focused on optimizing the manipulator in such a way that the twist angles are locked.

5.1 Results

Figure 4 shows studied manipulators configurations after the optimization process. Joint value for each joint for all configurations are set to zero. Tables 3 - 7 show the optimized

Table 5 The optimized DH parameters for the manipulator 3. Joint variables are bolded

frame	a_{i-1}	d_i	α_{i-1}	θ_i
1	0	0	0	$\boldsymbol{\theta}_1$
2	a_1	0	$\pi/2$	$\boldsymbol{\theta}_2 - \pi/2$
3	0	$d_{3p} + \mathbf{d}_{3j}$	$-\pi/2$	0
4	0	d_4	0	$\boldsymbol{\theta}_4 - \pi/2$
5	0	d_5	$\pi/2$	$\boldsymbol{\theta}_5 + \pi/2$
6	a_5	d_6	$-\pi/2$	$\boldsymbol{\theta}_6 - \pi/2$

Table 6 The optimized DH parameters for the manipulator 4. Joint variables are bolded

frame	a_{i-1}	d_i	α_{i-1}	θ_i
1	0	0	0	$\boldsymbol{\theta}_1$
2	a_1	0	$\pi/2$	$\boldsymbol{\theta}_2 + \pi/2$
3	a_2	$d_{3p} + \mathbf{d}_{3j}$	$\pi/2$	0
4	a_3	d_4	$\pi/2$	$\boldsymbol{\theta}_4 + \pi/2$
5	a_4	d_5	$\pi/2$	$\boldsymbol{\theta}_5 + \pi/2$
6	a_5	d_6	$\pi/2$	$\boldsymbol{\theta}_6 + \pi/2$

Table 7 The optimized DH parameters for the manipulator 5. Joint variables are bolded

frame	a_{i-1}	d_i	α_{i-1}	θ_i
1	0	0	0	$\boldsymbol{\theta}_1 - \pi/2$
2	0	$\mathbf{d}_2$	$-\pi/2$	0
3	0	d_3	0	$-\pi/2$
4	a_3	d_4	0	$\boldsymbol{\theta}_4$
5	a_4	d_5	0	$\boldsymbol{\theta}_5$
6	a_5	d_6	$-\pi/2$	$\boldsymbol{\theta}_6 - \pi/2$
7	a_6	d_7	$-\pi/2$	$\boldsymbol{\theta}_7$

DH parameters for each configuration. Results are given for five topologies, because one topology was not able to solve the given task (configuration 6). An example of one configuration to reach all task points is shown in Fig. 3. Green color indicates that the task pose is reached. This figure shows also the workspace constraint, which was same for all configurations. Joint limits for a prismatic joint were $d_i \in [0, 3]$ and for a revolute joint $\theta_i \in [-\pi, \pi]$. Link length limits for a and d parameters were $a_1 \in [-2, 2]$ and $d_i \in [-2, 2]$.

Table 2 shows the GCI and KCI for all the manipulator configurations that were optimized. These indices indicate that the best serial manipulator structure for the case study is alternative 2, in the sense of the GCI and KCI. These performance metrics are needed to rank the optimized topologies. Five of the six topologies were able to fulfill the task requirements and thereafter these optimized structures need to be sorted according

to some metrics. The GCI is an attractive choice for this case study because it describes manipulator's possibilities of generating output forces from input torques and Cartesian velocity from joint velocity. And the KCI describes manipulator's isotropy.

6 Conclusion

In this paper, a generic method for selecting and optimizing a redundant serial manipulator structure is presented. The method finds an optimal solution in the sense of the LM for a constrained optimization problem. The case study shows that the proposed method can be used as a design tool to find the most attractive structure for a manipulator with specific constraints.

This method gives a kinematic structure for building the manipulator. Thereafter, the link arms must be sized with the proper material strengths, and the joints need to find suitable actuators. If this is not possible for the structure produced by the optimization method, one have to add such constraints that the link lengths and joint actuators can be implemented and do the optimization again with proper constraints.

The result shows that the drilling pattern of holes of the case study can be achieved with many topologies. Which topology is the best depends on the application. In this case, the global conditioning index and kinematic conditioning index were used to rank the optimized topologies to find the best manipulator structure. If the application is different from this case study, different performance measures can be used to rank the manipulators and to find the best manipulator structure.

In this paper, six topologies were optimized to show how the proposed method can be used to find the most attractive manipulator structure. However, in general, it is possible to automate the kinematic synthesis process so that all possible topologies are generated automatically and then optimized. Then the performance indices are calculated for every topology, and the topologies are ranked according to the selected performance method. This procedure ensures that the suitable manipulator structure is found.

The effect of uncertainty in task points for the optimization results will be studied in the future studies. The proposed optimization method will also be extended to find the best possible structure for the serial type manipulator from all possible topologies.

References

Angeles, J. & López-Cajún, C. S. (1992), 'Kinematic isotropy and the conditioning index of serial robotic manipulators', *Int. J. Rob. Res.* **11**(6), 560–571.

Baillieul, J. (1985), Kinematic programming alternatives for redundant manipulators, *in* 'IEEE International Conference on Robotics and Automation. Proceedings. 1985', Vol. 2, pp. 722–728.

Barissi, S. & Taghirad, H. (2008), Task based optimal geometric design and positioning of serial robotic manipulators, *in* 'IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications, 2008. MESA 2008.', pp. 158–163.

Buss, S. R. & Kim, J.-S. (2004), 'Selectively damped least squares for inverse kinematics', *Journal of Graphics Tools* **10**, 37–49.

Chan, T. F. & Dubey, R. (1995), 'A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators', *IEEE Transactions on Robotics and Automation,* **11**(2), 286–292.

Chiaverini, S., Siciliano, B. & Egeland, O. (1994), 'Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator', *IEEE Transactions on Control Systems Technology* **2**(2), 123–134.

Craig, J. (1989), *Introduction to Robotics: Mechanics and Control*, Addison-Wesley series in electrical and computer engineering: Control engineering, Addison-Wesley.

Golub, G. & Kahan, W. (1965), 'Calculating the singular values and pseudo-inverse of a matrix', *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis* **2**(2), 205–224.

Gosselin, C. & Angeles, J. (1991), 'A global performance index for the kinematic optimization of robotic manipulators', *Journal of Mechanical Design* **113**(3), 220–226.

Khatib, O. (1985), Real-time obstacle avoidance for manipulators and mobile robots, *in* 'IEEE International Conference on Robotics and Automation,', Vol. 2, pp. 500–505.

Klein, C. & Huang, C.-H. (1983), 'Review of pseudoinverse control for use with kinematically redundant manipulators', *IEEE Transactions on Systems, Man and Cybernetics,* **SMC-13**(2), 245–250.

Kucuk, S. & Bingul, Z. (2006), 'Comparative study of performance indices for fundamental robot manipulators', *Robotics and Autonomous Systems* **54**(7), 567 – 573.

Lee, J. H., Eom, K. S. & Suh, I. I. (2001), Design of a new 6-dof parallel haptic device, *in* 'Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation', Vol. 1, pp. 886–891 vol.1.

Maciejewski, A. A. & Klein, C. A. (1985), 'Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments', *The International Journal of Robotics Research* **4**(3), 109–117.

Merlet, J.-P., Gosselin, C. M. & Mouly, N. (1998), 'Workspaces of planar parallel manipulators', *Mechanism and Machine Theory* **33**(1), 7 – 20.

Ouezdou, F., Régnier, S. & Mavroidis, C. (2007), 'Kinematic synthesis of manipulators using a distributed optimization method', *Journal of Mechanical Design* **121**, 492–501.

Patel, S. & Sobh, T. (2015*a*), 'Manipulator performance measures - a comprehensive literature survey', *Journal of Intelligent Robotic Systems* **77**(3), 547–570.

Patel, S. & Sobh, T. (2015*b*), 'Task based synthesis of serial manipulators', *Journal of Advanced Research* **6**(3), 479 – 492. Editors and International Board Member collection.

Puglisi, L. J., Saltaren, R. J., Moreno, H. A., CÃ¡rdenas, P. F., Garcia, C. & Aracil, R. (2012), 'Dimensional synthesis of a spherical parallel manipulator based on the evaluation of global performance indexes', *Robotics and Autonomous Systems* **60**(8), 1037 – 1045.

Salisbury, J. & Craig, J. (1982), 'Articulated hands: Force control and kinematic issues', *The International Journal of Robotics Research* **1**(1), 4–17.

Schinstock, D., Faddis, T. & Greenway, R. (1994), 'Robust inverse kinematics using damped least squares with dynamic weighting', *AIAA PAPER 94-0889-CP*.

Sciavicco, L. L. & Siciliano, B. (2000), *Modelling and control of robot manipulators*, Advanced textbooks in control and signal processing, Springer, London, New York.

Shiakolas, P., Koladiya, D. & Kebrle, J. (2002), 'Optimum robot design based on task specifications using evolutionary techniques and kinematic, dynamic, and structural constraints', *Inverse Problems in Engineering* **10**(4), 359–375.

Singla, E., Tripathi, S., Rakesh, V. & Dasgupta, B. (2010), 'Dimensional synthesis of kinematically redundant serial manipulators for cluttered environments', *Robotics and Autonomous Systems* **58**(5), 585–595.

Sobh, T. M. & Toundykov, D. Y. (2004), 'Optimizing the tasks at hand [robotic manipulators]', *IEEE Robotics Automation Magazine* **11**(2), 78–85.

Stocco, L. J., Salcudean, S. E. & Sassani, F. (1999), 'On the use of scaling matrices for task-specific robot design', *IEEE Transactions on Robotics and Automation* **15**(5), 958–965.

Sun, Y., Liu, H., Luo, Z. & Wang, F. (2007), Robot mechanical structure optimization design, *in* 'IEEE International Conference on Robotics and Biomimetics, 2007. ROBIO 2007.', pp. 1919–1923.

Tarek, S. & Daniel, T. (2003), Kinematic synthesis of robotic manipulators from task descriptions, *in* 'Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003.', Vol. 2, pp. 1018–1023 vol.2.

Vijaykumar, R., Waldron, K. J. & Tsai, M. J. (1986), 'Geometric optimization of serial chain manipulator structures for working volume', *Int. J. Rob. Res.* **5**(2), 91–103.

Warren, C. (1989), Global path planning using artificial potential fields, *in* 'IEEE International Conference on Robotics and Automation,', pp. 316–321 vol.1.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-4043-1
ISSN 1459-2045