

Submitted by
Dipl. Ing. Hannes
Gamper BSc

Submitted at
Institute of Robotics

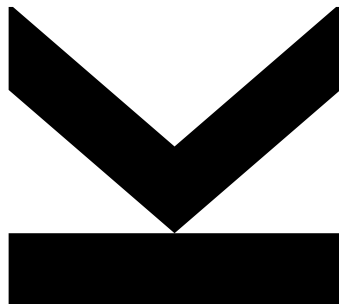
Supervisor and
First Evaluator
Prof. Dr. Andreas
Müller

Second Evaluator
Prof. Dr. Raul Marin
Prades

Co-Supervisor
Dr. Mario Di Castro

Geneva, March 2024

A Robotic System for CERN's Future Circular Collider



Doctoral Thesis
to obtain the academic degree of
Doktor der technischen Wissenschaften
in the Doctoral Program
Technische Wissenschaften



Preface

"When an astronaut appears to have turned by 90 degrees he reached the event horizon - not in one piece of course."

"The Lagrangian equation of the standard model would avoid non-holonomic robot models. Might be a overkill though."

"Is the universe discrete or continuous?"

"Do Pythagorean fifths sound well by training or is there an underlying absolute phenomena?"

"If we are in a simulation, I wouldn't change anything."

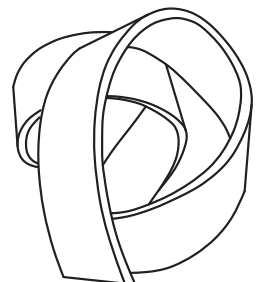
... lunch conversations at CERN. Mind bending. Incredibly fulfilling. Unforgettable. Grateful.

Science is great. Science connects.

Science is powerful. Science is humbling.

Science opens gates to new worlds. Curiosity is everything.

Science gifts purpose.



Abstract

The European Center of Nuclear Research (CERN) has over the years significantly contributed to our current understanding of the universe. The completion of the Standard Model of Particle Physics, with the discovery of the Higgs boson, marks the latest milestone in CERN's research efforts. Yet, there are still phenomena, like dark matter, the prevalence of matter over antimatter or the neutrino mass, which cannot be described by the Standard Model. This suggests, that there must be more, physics which goes beyond the Standard Model and still has to be discovered. A good chance to unveil such behavior is thought to lie in particles with a mass above 14 TeV and thus unable to be created by the current machines at CERN. To unlock observations in these high energy ranges, a new particle accelerator with a center of mass energy of 100 TeV and a circumference of 100 km was suggested: the Future Circular Collider - FCC.

One of the FCC study's research areas concerns the automation of maintenance, inspection and emergency handling along the 100 km long FCC tunnel. The automation of these tasks play a significant role for downtime, reliability and safety of the particle accelerator and decreases the radiation exposure of workers. These tasks require various adept mechanical interactions with its environment in an extremely large work space. The development of such a robotic system is the over arching topic of this thesis.

First, an in-depth study was performed to identify and clearly define the requirements and restrictions on a robotic system operating in the FCC environment. This study then concludes with high level recommendations for the so called FCC Robot, aiming to provide a solid bases for subsequent developments even beyond this thesis.

Based on the extracted requirements and restrictions, a novel design optimization algorithm has been developed to identify an optimal topology and geometry for the robotic system, which resulted in a 11 Degree of Freedom (DoF) manipulator. In order to control such a highly redundant mechanical structure in the C++ CERN Robotic Framework (CRF) a new motion controller module, containing the implementation of a novel inverse kinematics algorithm, was developed. Finally, a prototype of the FCC Robot has been designed, built and deployed in a mock-up of the LHC tunnel for proof of concept studies and comparative cost estimation.

The thesis concludes with recommendations on future improvements, a four year work package definition for the development of the final robotic system including a cost estimation and a summary on established collaborations with universities and other external institutions.

Kurzfassung

Das Europäische Zentrum für Kernforschung (CERN) hat im Laufe der Jahre maßgeblich zu unserem aktuellen Verständnis des Universums beigetragen. Die Fertigstellung des Standardmodells der Teilchenphysik durch die Entdeckung des Higgs-Bosons am CERN war ein Meilenstein für die Physik. Dennoch gibt es immer noch Phänomene wie die dunkle Materie, die Vorherrschaft von Materie über Antimaterie oder die Neutrinomasse, die nicht durch das Standardmodell beschrieben werden können. Dies legt nahe, dass es eine Physik geben muss, die über das Standardmodell hinausgeht und noch entdeckt werden muss. Eine gute Chance, ein solches Verhalten aufzudecken, wird in Partikeln mit einer Masse über 14 TeV gesehen, die von den aktuellen Maschinen am CERN nicht erzeugt werden können. Um Beobachtungen in diesen hochenergetischen Bereichen zu ermöglichen, wurde ein neuer Teilchenbeschleuniger mit einer Schwerpunktsenergie von 100 TeV und einem Umfang von 100 km vorgeschlagen: der Future Circular Collider - FCC.

Eines der Forschungsgebiete der FCC-Studie betrifft die Automatisierung von Wartung, Inspektion und Notfallinterventionen entlang des 100 km langen FCC-Tunnels. Die Automatisierung dieser Aufgaben spielt eine bedeutende Rolle für die Zuverlässigkeit und Sicherheit des Teilchenbeschleunigers und verringert zudem die Strahlenexposition der betroffenen Mitarbeiter. Diese zu automatisierenden Aufgaben erfordern geschickte mechanische Interaktionen mit der Umgebung in einem äußerst großen und komplexen Arbeitsraum. Die Entwicklung eines solchen Robotersystems ist das übergreifende Thema dieser Arbeit.

Zunächst wurde eine eingehende Studie durchgeführt, um die Anforderungen für ein Robotersystem im FCC-Umfeld zu identifizieren und klar zu definieren. Diese Studie schließt mit Empfehlungen für den sogenannten FCC-Roboter ab, die eine solide Basis für zukünftige Entwicklungen bietet.

Basierend auf den extrahierten Anforderungen wurde ein neuartiger Designoptimierungsalgorithmus entwickelt, um eine optimale Topologie und Geometrie für das Robotersystem zu identifizieren, was in einem Manipulator mit Freiheitsgrad 11 resultierte. Um eine solch hochgradig redundante, mechanische Struktur im C++ CERN Robotic Framework (CRF) zu steuern, wurde ein neues Reglermodul entwickelt, das die Implementierung eines ebenfalls neu entwickelten Inverskinematik-Algorithmus enthält. Schließlich wurde ein Prototyp des FCC-Roboters entwickelt, gebaut und in einem Modell des LHC-Tunnels - für Studien zur Konzeptbeweissführung und vergleichende Kostenschätzung eingesetzt.

Die Arbeit schließt mit Empfehlungen für zukünftige Verbesserungen und der Definition eines vierjährigen Arbeitspaketes für die Entwicklung des finalen Robotersystems ab. Ebenso angeführt werden eine Kostenschätzung sowie eine Auflistung aller etablierten Zusammenarbeiten mit Universitäten und anderen externen Institutionen.

Acknowledgment

I would like to express my heartfelt gratitude to the individuals and organizations who have contributed to the completion of this thesis. Their unwavering support, guidance, and encouragement have been instrumental in shaping this journey:

First and foremost, I extend my sincere appreciation to my supervisor and mentor, Prof. Dr. Andreas Mueller. His exceptional guidance, both on a technical and personal level, has been invaluable throughout this entire endeavor. His insights and mentorship have shaped the direction of my work and have been a constant source of inspiration.

I am equally indebted to Prof. Dr. Hubert Gattringer for his insightful advice and expertise on intricate technical matters. His contributions have been pivotal in ensuring the robustness and depth of my research.

From the CERN perspective, I am deeply grateful to Dr. Mario Di Castro, Dr. Luca Buonocore, and Dr. Eloise Matheson, as well as the entire MRO section. Their professionalism, motivation, and collaborative spirit have created an enriching and stimulating work environment. My heartfelt appreciation goes to all my colleagues at CERN, whose friendship and shared dedication have made this journey even more fulfilling.

I would like to extend a special acknowledgment to Alejandro Diaz Rosales MSc, Jorge Playan Garai MSc and Sergio Di Giovannantonio MSc. Their exceptional dedication and support, which often transcended the boundaries of their roles, have been instrumental in the realization of this thesis. Their contributions have left an indelible mark on this work.

I also want to express my profound gratitude to my family, especially my parents Gertraud and Georg and my partner Yulia. Their unwavering support, love, and encouragement have been my anchor and source of strength, enabling me to dream big and reach new heights.

I am incredibly thankful for the multi-national friendships that I have forged and the inspiring discussions that have enriched my perspective. The evenings of friendship, laughter, and insightful conversations, both at CERN and beyond, have been an integral part of this journey. I wish to acknowledge the Austrian Doctoral Student Program at CERN, as well as the "Bundesministerium für Wissenschaft, Forschung und Wirtschaft (BMWFW)" for their funding and support. The opportunity to work within CERN has been transformative, and I am grateful for their contribution to my academic and professional growth.

Lastly, my education at Johannes Kepler University, Linz in general, and especially at the Institute of Robotics, has provided me with a strong foundation that opened doors I could have never imagined and made this journey possible in first place.

In conclusion, this thesis is the culmination of the collective efforts, knowledge, and support of numerous individuals and organizations. I am deeply thankful for their contributions, and I am excited to carry forward the knowledge and experiences gained from this journey.

Contents

1	Introduction	1
1.1	The European Organization for Nuclear Research	2
1.2	CERN's Robotic Ecosystem	4
1.3	Contribution of this Thesis	5
2	High-Level Recommendations for a Robotic System for the Future Circular Collider	9
2.1	State of the Art - Task Specific Robots for big Science Facilities & Harsh Environments	10
2.2	Current State of Remote Maintenance at CERN	12
2.2.1	Environment	13
2.2.2	Robotic Fleet of the Robotics Section at CERN	13
2.2.3	Summary & Comparison	16
2.3	The Future Circular Collider	18
2.3.1	Layout of the Cross-Sections	19
2.3.2	Radiation	21
2.4	General Requirements on a Robotic System for FCC	21
2.5	Requirements Deduced from Interventions in the LHC	23
2.5.1	Measuring Radiation, Monitor Tunnel Integrity	23
2.5.2	Calibration of BLM Sensors	24
2.5.3	Measure Oxygen, Measure Alignment, Audio Inspection, Vacuum Leak De- tection and General Inspection Tasks	24
2.5.4	Coupling of Docking Stations	25
2.5.5	Transport Tools/Material	25
2.5.6	Guide People, Follow & Monitor People, Search for Life, Patrols	25
2.5.7	Locate & Approach Fire, Extinguish Fire	25
2.6	Recommendations for a Robotic Solution for FCC	25
2.6.1	Locomotion	26
2.6.2	Manipulator	28
2.6.3	End-Effector	29
2.6.4	Power Supply	29
2.6.5	Emergency System	29
2.6.6	Integration	30
2.6.7	Illustration of Concept Robot	31
2.6.8	Summary	31
2.7	Integration in the Official Base Line Design of the FCC Tunnel	32
2.8	Conclusion	33

3	Modelling & Control of Multibody Systems	35
3.1	Position & Orientation	37
3.1.1	Orientation Representations	37
3.1.2	Error Measure	39
3.2	Translation & Rotation	39
3.3	Direct Kinematics	40
3.3.1	The Kinematic Chain	41
3.3.2	Geometric Jacobian	42
3.4	Inverse Kinematics	43
3.4.1	Nullspace & Range of the Jacobian	45
3.4.2	Minimum Norm Solution	46
3.4.3	Numeric Solution via Optimization	48
3.4.4	Augmented Jacobian	48
3.4.5	Task Priority	49
3.4.6	Representation, Kinematic & Algorithmic Singularities	49
3.4.7	Numeric Stabilization	50
3.5	Kinetics	51
3.5.1	Inverse Dynamics	52
3.6	Control	52
3.6.1	Flat Systems	53
3.6.2	Stability of the Tracking Problem	53
3.6.3	Holonomic & Non-Holonomic Systems & Coordinates	55
3.7	Mobile Platform with Manipulator	56
4	A Novel Design Optimization Technique for Simultaneous Optimization of Topology and Geometry	61
4.1	Design Optimization Algorithm	62
4.1.1	Kinematic and Dynamic Model	63
4.1.2	Collision Avoidance	63
4.1.3	Problem Formulation	64
4.1.4	Objective Function	64
4.2	Application: Cavity Inspection Robot	68
4.2.1	Model	69
4.2.2	Collision Avoidance	71
4.2.3	Kinematic and Dynamic Performance Criteria	71
4.2.4	Initial Configuration	72
4.2.5	Optimization Results	74
4.3	Application: FCC Manipulator	76
4.3.1	Manipulator & Environment	77
4.3.2	Kinematic and Dynamic Performance Criteria	78
4.3.3	Optimization Results	78
4.4	Conclusion	79
5	A Novel Inverse Kinematics Technique with Smooth Task Switching	81
5.1	Related Work	82
5.2	Smooth Task Switching Algorithm	83
5.2.1	Switching Parameter	84
5.3	Stability Analysis	85
5.3.1	Stability of the Tracking Tasks	85
5.3.2	Discrete Systems	86

5.4	Test Environment	87
5.4.1	Train Inspection Monorail - 9 DoF	87
5.4.2	Universal Robot UR10e - 6 DoF	88
5.4.3	Mobile Platform with Arm - 9 DoF	88
5.5	Results	89
5.5.1	Train Inspection Monorail - 9 DoF	89
5.5.2	Universal Robot UR10e - 6 DoF	90
5.5.3	Mobile Platform with Arm - 9 DoF	93
5.6	Conclusion	95
6	A Modular Motion Controller for the CERN Robotic Framework	97
6.1	The CERN Robotic Framework	99
6.1.1	Framework Infrastructure	101
6.1.2	Continuous Integration	102
6.1.3	Operating System	104
6.2	The Motion Controller Module	105
6.3	Development of an Open-Source Kinematics Library	107
6.3.1	Dependencies & License	111
6.4	Showcase Examples	111
6.5	Conclusion	112
7	Hardware Design for the FCC Robotic System	113
7.1	Mechanical Design	114
7.2	Motor Design	116
7.3	Electrical Design	119
7.4	Installation	123
7.5	Conclusion	125
8	Conclusion & Outlook	127
A	Appendix	139
A.1	Selected UML Diagrams of the CRF	139
A.2	Code Snippets	141
A.3	FCC Robotics Work Package	144

List of Figures

1.1	Geographical location of the PS, SPS, LHC and FCC.	2
1.2	Features of robots in structured and unstructured environments.	4
1.3	Functional diagram of a robotic system. The highlighted blocks correspond to the research focus of JKU's Institute of Robotics (blue) and CERN's MRO section (red).	5
2.1	Task specific robots at the large research facilities JET [1] (a), ITER [2] (b), SNS [3] (c) and ESS [4] (d).	11
2.2	Overview of the CERN facilities and the task specific robots TIM, KUKA and CRANEBot with fixed locations. FCC Robot indicates the yet unknown robotic solution for FCC. The three figures on top show the environment of the LHC tunnel hosting TIM, the experimental cavern CMS hosting CRANEBot and the ISOLDE facility with KUKA manipulators.	14
2.3	A subset of the robotic fleet consisting of Teodor (a), Telemax (b), CERNBot (c), TIM (d), MIRA (e), CRANEBot (f) and CHARMBot (g).	15
2.4	Intervention sites for each robot. Robots are ordered depending on their modularity and robustness. The amount of different sites the robots have been used at, correlate with increasing modularity.	16
2.5	Robots in the example environments from Figure 2.2. CRANEBot in the experimental cavern CMS (a), Kuka in the laboratory ISOLDE (b) and TIM in the LHC tunnel (c).	17
2.6	The Future Circular Collider	18
2.7	Cross-Section V-V from Figure 2.6: The FCC machine tunnel. Configuration FCC-ee (a) and FCC-hh (b). Dimension are in meters.	19
2.8	Cross-Section U-U from Figure 2.6: K - Experimental Cavern, L - Detector, M - Connection Tunnels, N - Access Cavern. Dimension are in meters.	20
2.9	Cross-Section W-W from Figure 2.6: O - Clystron Gallery, P - Electric installation for RF, Q - Main Tunnel, R - Connection Tunnels, S - Service Cavern. Dimension are in meters.	21
2.10	Rating Catalog for Locomotion Solutions	27
2.11	The visualization of the recommendation for a robotic system for FCC in the FCC-ee environment.	31
2.12	Reserved area for the robotic system in the cross-section (V-V from Figure 2.6) of the FCC tunnel. Here the FCC-ee configuration is shown. Dimensions in meter.	32

2.13	The official base line design used by the FCC study for communication of the current status. The two cross-sections on the left are showing the FCC-ee configuration and on the right the FCC-hh configuration.	33
3.1	Rigid body in space.	37
3.2	Transformation of the unit vectors of frame B to frame I	37
3.3	A 2 DoF serial mechanism.	41
3.4	Methods for Inverse Kinematic Solutions	44
3.5	Mapping of the Jacobian matrix	45
3.6	(a) The SPSBot from the CERN robotic fleet has four mecanum wheels and a six DoF manipulator attached to its base. (b) A detailed view on one of the four mecanum wheels.	57
4.1	Workflow of the design optimization algorithm	62
4.2	Optimization effects using different function types for J_1	66
4.3	Surface plot of objective function $J_1(\mathbf{p}) = \mathbf{k}^T \mathbf{h}(\mathbf{p})$	67
4.4	Contour plot of objective function $J_1(\mathbf{p}) = \mathbf{k}^T \mathbf{h}(\mathbf{p})$	68
4.5	Surface plot of objective function $J_1(\mathbf{p}) = \mathbf{k}^T \mathbf{e}(\mathbf{p})$	69
4.6	Contour plot of objective function $J_1(\mathbf{p}) = \mathbf{k}^T \mathbf{e}(\mathbf{p})$	70
4.7	Cavity Types for FCC, LINAC and LHC (left to right, all units in mm)	70
4.8	Design space	70
4.9	Collision detection for one robot link	71
4.10	Initial states for the LINAC cavity (units in m)	73
4.11	Initial states for the LHC cavity (units in m)	73
4.12	Initial states for the FCC cavity (units in m)	73
4.13	Optimized design for the LINAC cavity (units in m)	74
4.14	Optimized topology - LINAC	75
4.15	Optimized design for the LINAC cavity (units in m)	75
4.16	Optimized design for the LHC cavity (units in m)	76
4.17	Optimized design for the FCC cavity (units in m)	76
4.18	Optimized topology	76
4.19	Design space	77
4.20	FCC-ee Cross-Section Layout	78
4.21	FCC-hh Cross-Section Layout	78
4.22	Optimization results FCC-ee	79
4.23	Optimization results FCC-hh	79
4.24	Optimized topology	79
5.1	The architecture of the Kinematics Library. The robot configuration file holds the geometric description of the mechanical system in either the URDF format or hard coded mathematical expressions, that will be parsed using the urdfdom or ExprTk library. The block that implements (5.9) uses Eigen3. The configuration file also holds all other parameters that are relevant for the motion controller of the CRF.	87
5.2	The Train Inspection Monorail (TIM) during testing in the LHC tunnel.	88
5.3	Omnidirectional base with a Kinova Jaco2 with spherical wrist	89
5.4	Test data recorded with TIM in the LHC tunnel. Showing tracking error $\theta_{1,i}$ for the i -th dimension (a), tracking error θ_2 (b), the inverse kinematics solution in joint angles \mathbf{q} (c) and joint velocities $\dot{\mathbf{q}}$ (d), the gradient of the optimization task τ (e) and the switching parameters σ_θ and σ_τ (f).	91

5.5	Test data recorded with UR10e. Showing the desired task reference $\mathbf{z}_{ref,1}$ (a), tracking error $\boldsymbol{\theta}_1$ (b), the inverse kinematics solution in joint angles \mathbf{q} (c) and joint velocities $\dot{\mathbf{q}}$ (d), the gradient of the optimization task τ (e) and the switching parameters σ_θ and σ_τ (f).	92
5.6	Test data recorded with the mobile platform with arm. Showing tracking error $\boldsymbol{\theta}_1$ (a), the inverse kinematics solution in joint angles \mathbf{q} (b) and (c) and joint velocities $\dot{\mathbf{q}}$ (d) and (e) and the weighing gains σ_{Wi} for the i -th dimension (f). . .	94
6.1	The CERN Robotic Framework - CRF. An example showcasing the CRF used to control the FCC Robot.	100
6.2	The directory structure of the CRF. The expanded directory locates the implementation of the Universal Robot.	102
6.3	Visualization of the work flow for a new implementation in the CRF.	103
6.4	Latency times for the standard (left) and the preempted (right) kernel on the same embedded PC with Ubuntu 20.04.	105
6.5	Overview of the navigation module of the CRF.	106
6.6	UML diagram of the IRobot interface and its implementations.	106
6.7	Modules and libraries of the Kinematics Library	109
6.8	Implementation of the inverse kinematics algorithm visualized with the modules from Figure 6.7.	110
6.9	CamBot and the derived kinematic chain from the URDF file.	112
7.1	CAD model of the full robotic system.	115
7.2	Topology optimization of link 5. The weight of the link was reduced from 26.4 kg (a) to 15.3 kg (b) and finally to 9.9 kg (c).	115
7.3	Stress analysis of links 4 and 5.	116
7.4	Natural frequency analysis of links 4 and 5.	116
7.5	Simulation data for motor design.	118
7.6	Electrical Schematic of the FCC Prototype	121
7.7	The CAD design of the robot base including the cabling (top) and the cabling work ongoing on the prototype (bottom).	124
7.8	The designed circuit monitors the voltage in the main circuit. In case of an voltage increase, energy will be dissipated via the power resistors.	124
7.9	Installation phases of the FCC robot	125
A.1	UML diagram of the ITaskTrajectoryGenerator interface.	139
A.2	UML diagram of the IJointTrajectoryGenerator interface.	140
A.3	UML diagram of the IController interface.	141

List of Tables

2.1	General requirements on a robotic system for FCC.	22
2.2	Past and ongoing interventions in the LHC and the derived requirements for a robotic system for the FCC.	23
2.3	Relative dexterity and relative payload for different manipulator topologies. . . .	29
2.4	Summary of the recommendations for a robotic system for FCC. Bold letters refer to corresponding labels in Figure 2.11.	32
3.1	Secondary tasks and their corresponding artificial potential.	47
4.1	Initial geometry	70
4.2	Optimized geometry - LINAC	75
4.3	Optimized geometry	76
4.4	Initial geometry	77
4.5	Optimized geometry	79
6.1	List of external libraries and corresponding licenses.	111
7.1	List of the components of the FCC robotic system.	120

List of Acronyms and Symbols

The following tables will provide a comprehensive list of the most frequently used symbols and abbreviations to provide a quick reference for the reader, to avoid searching for definitions in the entire document. However, all symbols and abbreviations will also be introduced at their first appearance.

Acronym	Definition
API	Application Programming Interface
AR	Augmented Reality
BLM	Beam Loss Monitor
CCC	CERN Control Center
CDR	Conceptual Design Report
CERN	Conseil Européen pour la Recherche Nucléaire or European Organization for Nuclear Research
CI	Continuous Integration
CLIK	Closed-Loop Inverse Kinematics
CMS	Compact Muon Solenoid
CRF	CERN Robotic Framework
DoF	Degree of Freedom
ESS	European Spallation Source
FCC	Future Circular Collider
FCC-ee	Future Circular Collider for lepton collisions
FCC-hh	Future Circular Collider for hadron collisions
GUI	Graphical User Interface
ISOLDE	Isotope Separator on Line Device
ITER	International Thermonuclear Experimental Reactor
JET	Joint European Torus
LHC	Large Hadron Collider
LINAC	Linear Accelerator
MoU	Memorandum of Understanding
MPC	Model Predictive Control
n-ToF	Neutrino Time of Flight
OLIK	Open Loop Inverse Kinematics
OS	Operating System
PMSM	Permanent Magnet Excited Synchronous Machine
PS	Proton Synchrotron
PSO	Particle Swarm Optimization

Acronym	Definition
RT	Real-Time
RF	Radio Frequency
SNS	Spallation Neutron Source
SPS	Super Proton Synchrotron
SUT	Subject Under Testing
TDR	Technical Design Report
TIM	Train Inspection Monorail
UML	Unified Modeling Language
URDF	Unified Robotics Description Format
VR	Virtual Reality

Symbol	Definition
$\dot{()}$	Derivative by time
$ () $	L2 Norm
$()^T$	Transpose
$()^\dagger$	Damped Pseudo Inverse
$()^\perp$	Orthogonal Complement
$\tilde{()}$	Cross product matrix of vector $()$
$()^e$	External Forces
$()_d$	Desired values
$()_{ref}$	Reference values
$()_{act}$	Actual values
∇	Nabla Operator
A	Bold capital letters represent matrices
a	Bold small letters represent vectors
c	Collision inequality constraints
E	End effector frame
\hat{E}	End effector position
e	Error, Unit vectors defining the span of a coordinate systems base
\mathbf{f}^e	External forces
$\mathbf{f}(\mathbf{q})$	Forward Kinematics
G1 ... G10	General Requirements
g	Pruning function, Non-linear terms in equation of movement
I	Identity matrix
I	Inertial frame
J	Cost function in optimization problem
$\mathbf{J}(\mathbf{q})$	Jacobian Matrix
J	Inertia Matrix
K	Constant matrix of gains
k	Constant vector of gains
L	Rotational impuls
lb	Lower bounds
M1 ... M14	Maintenance Requirements
M	Mass matrix
\mathbf{M}^e	External torques
N	Null space
$\mathcal{N}()$	Kernel of mapping
p	Translational impuls
Q	External forces in equation of movement
q	Minimal coordinates
\mathbf{R}_{AB}	Rotation matrix from coordinate system B into A
\mathbf{r}_{BC}^A	Vector pointing from B to C represented in coordinate system A
$\mathcal{R}()$	Image of mapping
$\dot{\mathbf{s}}$	Minimal velocities
ub	Upper bounds
${}_I\mathbf{v}_S$	Absolute translational velocities of point S represented in the inertial frame I

Symbol	Definition
\mathbf{W}	Weighing matrix
w_{kin}	Kinematic Manipulability
\mathbf{z}	Task position
$\mathbf{0}$	Zero matrix
Δ	Difference
σ	Singular values, Switching gains
${}^I\boldsymbol{\omega}_{IE}$	Absolute angular velocities between the inertial I and end effector frame E represented in the inertial frame

Introduction

ABSTRACT. This chapter provides a comprehensive overview on the European Organization for Nuclear Research (CERN), the motivation for the investigation of a 100 km long Future Circular Collider (FCC) and implications for a robotic system in the main tunnel of the FCC. Subsequently an overview of CERN's robotic ecosystem is presented and a comparison on the methodologies and approaches to robotics at JKU's Institute of Robotics and CERN's Mechatronics, Robotics and Operations (MRO) section will be outlined. Furthermore, the contribution of this thesis on an academic level as well as the responsibilities of the author within the wider scope of the FCC Study will be highlighted.

Chapter Content

1.1	The European Organization for Nuclear Research	2
1.2	CERN's Robotic Ecosystem	4
1.3	Contribution of this Thesis	5

1.1 The European Organization for Nuclear Research

Since being founded in 1954, the European Organization for Nuclear Research (CERN) [5] has been a continuously successful institution for particle physics, contributing major steps to complete the Standard Model of Particle Physics. The consecutive particle accelerators PS (Proton Synchrotron) [6], SPS (Super Proton Synchrotron) [7] and LHC (Large Hadron Collider) [8] with their increasing center of mass collision energy of 50 GeV, 900 GeV and 14 TeV and corresponding increase in circumference of 628 m, 6.9 km and 27 km, were directly related to the award of the Nobel Prize in 1979 (PS), 1984 (SPS) and 2013 (LHC). The latter Nobel prize was related to the discovery of the Higgs-Boson, which was predicted almost 50 years before its detection at CERN by Peter Higgs and thus completed the most precise model of our universe so far, the Standard Model of Particle Physics.

Yet, there are still phenomena, like dark matter, the prevalence of matter over antimatter or the neutrino mass, which cannot be described by the Standard Model. This suggests, that there must be more, physics which goes beyond the Standard Model and still has to be discovered. A good chance to unveil such behavior is thought to lie in particles with a mass above 14 TeV and thus unable to be created by the current machines at CERN. To unlock observations in these high energy ranges, a new particle accelerator with a center of mass energy of 100 TeV and a circumference of 100 km was suggested: the Future Circular Collider - FCC.

Figure 1.1 shows the geographical location of the PS, SPS, LHC and FCC. The FCC tunnel lies around 100 m underground at its deepest point and will be built across the borders of France and Switzerland, cross the Lake Geneva in the north and will enclose the entire Mont Salève.

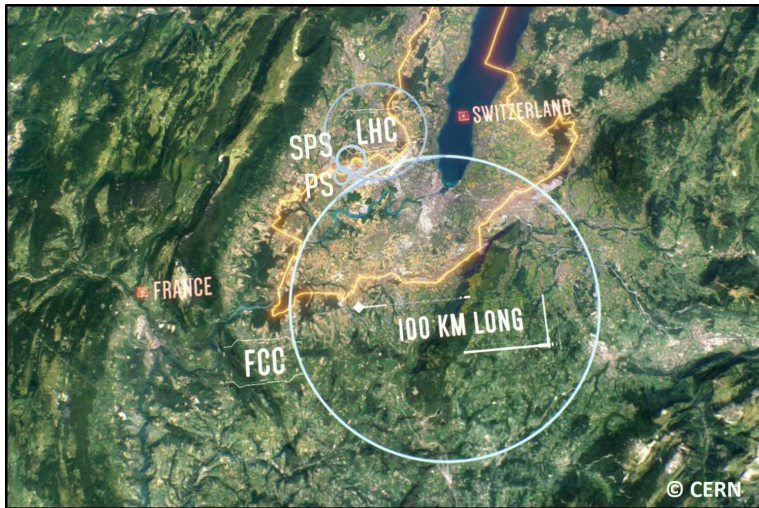


Figure 1.1: Geographical location of the PS, SPS, LHC and FCC.

A key parameter to demonstrate the efficiency of a particle accelerator is its luminosity, a metric that can be approximated as the volume of physics data it generates within a given time frame. To reach the luminosity targets [9] for the FCC, the proportion of days that the machine can deliver data, called machine availability, must reach a minimum of 80 % [10]. Recent studies, based on expert interviews and historical LHC data, have shown that this target can only be achieved by a 15-fold increase of the mean time between failures of certain critical systems.

Introducing a robotic system and thus reducing intervention times, allows to relax this constraint by a factor of three [10]. Thus, to ensure the required availability of a machine at this scale, it is necessary to integrate a holistic robotic concept for automated inspection and maintenance interventions.

To effectively reduce the intervention times, the robotic system must be capable of handling various procedures, ranging from simple inspection tasks to complex maintenance tasks that require adept mechanical interactions with the environment. Examples for required procedures with lower complexity are the monitoring of radiation and oxygen levels along the machine tunnel [11], tunnel health monitoring [12] or performing patrols to ensure that no personnel is present in the tunnel before starting the machine. More complex tasks contain the calibration of Beam Loss Monitors [13], replacement of equipment or alignment of the dipole magnets [14]. Even though the robotic system should be fully autonomous for all foreseeable and repetitive interventions, we still expect the necessity of human cognition for intervention planning in some scenarios and therefor novel teleoperation interfaces will be required [15].

The robotic system is also an integral part of the safety concept for the FCC. Ensuring the safety of the personnel has highest priority and requires substantial efforts considering a 100 km long tunnel filled with high power electronics, cryogenic installations, and radiation sources 100 m underground with access shafts only every 12 km. The robotic system must reach any point in the machine tunnel within a maximum of 10 min and must be able to detect hazards like fire or gas leaks, detect and guide personnel or support the CERN fire brigade with material transport or data from the incident location.

In 2013 the European Strategy for Particle Physics initiated a Conceptual Design Report (CDR) and feasibility study on the FCC. Since then numerous physics studies on future discoveries, the most promising energy ranges and the underlying theories, as well as engineering studies on possible construction sites, logistics, material research, automation techniques and socio-economic aspects investigating the impact of a GCHF science project in the middle of Europe, were conducted in collaborations of over 150 universities. The results were published in 2019 within the Conceptual Design Report (CDR) for a Lepton (FCC-ee) and Hadron (FCC-hh) Collider [16][9]. The 2020 Update of the European Strategy for Particle Physics [17] listed the further investigation of the FCC as one of three main priorities and subsequently launched a Technical Design Report (TDR).

Within the TDR in-depth studies like geodetic measurements to investigate possible placements of the tunnel, structural simulations, material handling during construction, power supply, emergency scenarios and many more are conducted to estimate feasibility, cost and construction time [18].

One of these studies focused on the automation of maintenance, inspection and emergency handling along the 100 km long FCC tunnel and resembles the first part of this PhD thesis. The automation of these tasks plays a significant role for downtime, reliability and safety of particle accelerators and decreases the radiation exposure of workers. The overarching objectives of automation are to reduce maintenance expenses, achieving the required machine availability, safeguard workers from hazardous interventions, and effectively respond to emergency situations.

In 2023 the studies for the TDR are in full progress and in 2025 a 50 m long string (segment of the FCC main tunnel) will be built on the CERN Preveessin site including all installation (dipols, cables, FCC Robot, ventilation, cryogenics, etc.) for final tests of the cross-section layout.

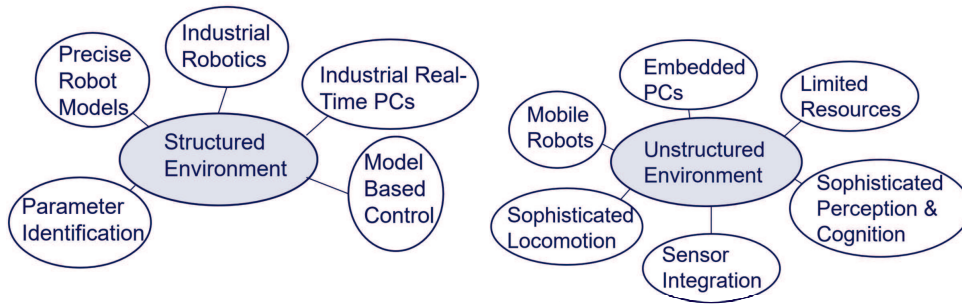


Figure 1.2: Features of robots in structured and unstructured environments.

1.2 CERN's Robotic Ecosystem

The research activities at CERN offered invaluable insights into the diverse interpretations of robotics across various institutions, exemplified by the contrast between the Institute of Robotics at Johannes Kepler University, Linz (JKU) and the Mechatronics, Robotics, and Operations (MRO) section at CERN. The differences in their approaches have unquestionably influenced this thesis and will therefore be elaborated upon in the following discussion.

JKU's Institute of Robotics has many collaborations with industrial partners, which allows to fund expensive, high-end hardware like industrial robots and powerful industrial real-time PC's. Furthermore the institute approaches research and education with a focus on the physical and mathematical foundation of robotics and control. Thus, JKU's Institute of Robotics finds itself well positioned to in research areas that require a deep understanding of the underlying mathematical or physical principles in robotics and test the developed algorithms in a structured environment, see Figure 1.2, with industry standards in terms of robots, sensors and PC's. The institute thus focuses on the development of novel algorithms in robotics and is able to validate them in a controlled environment using industrial real-time PC's.

CERN's MRO section mostly operates their robots in unstructured environments, see Figure 1.2, and has to quickly provide solutions for critical interventions in case of emergencies. Moreover, these interventions frequently necessitate the incorporation of highly specialized sensors crafted at CERN for particular applications, such as the assessment of Beam Loss Monitor (BLM) sensors along the tunnel. The MRO section, therefore, has a focus on fast problem-solving for interventions and rapid prototyping. This leads to a focus on software and hardware integration in a modular framework that controls the entire robotic fleet at CERN.

Figure 1.3 represents a functional diagram of a robot. The main focus of JKU's Institute of Robotics is on algorithmic development in the fields marked in blue and uses industrial tool chains for deployment. The CERN's MRO section has a strong focus on field marked in red and focuses on the integration of readily available solutions for other fields.

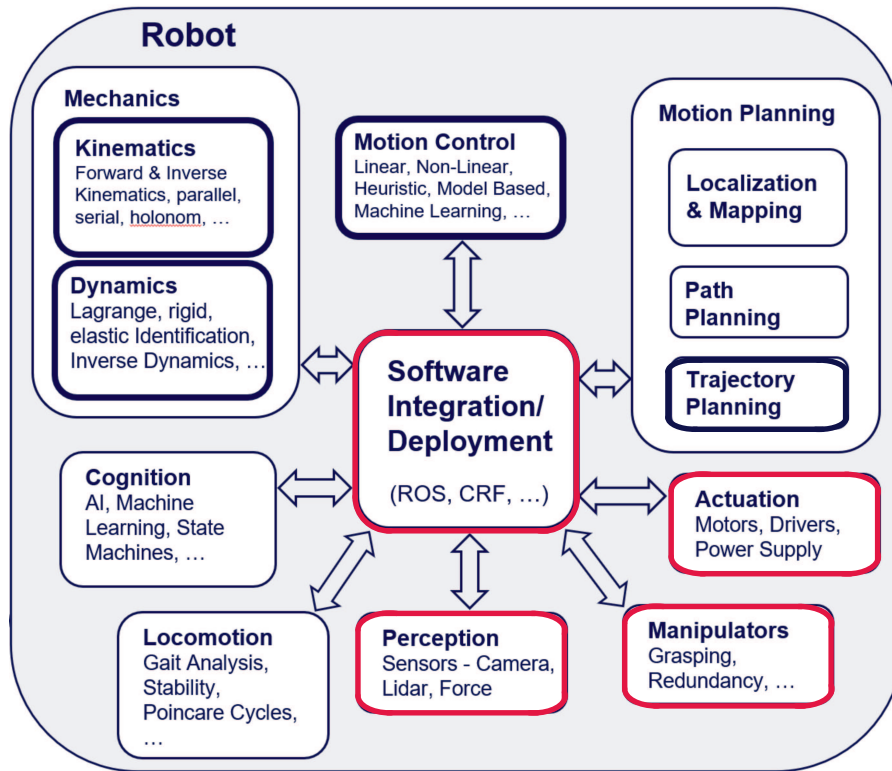


Figure 1.3: Functional diagram of a robotic system. The highlighted blocks correspond to the research focus of JKU's Institute of Robotics (blue) and CERN's MRO section (red).

1.3 Contribution of this Thesis

This section provides an overview of the contents of the thesis as well as tasks and responsibilities that were carried out within this project, but are not of a strict academic nature and hence not further discussed or presented in this document.

Chapter 2 presents the initial study on the feasibility, requirements and restrictions of a robotic system in the FCC and concludes with high-level recommendations for such a robotic system. Chapter 3 will introduce the theory and conventions referred to throughout the entire thesis for the modelling and control of multi-body system. Chapter 4 is dedicated to identify an optimal design of the FCC robot considering the environment and interventions that the system has to perform. The result of this design optimization was a 11 DoF system, which could not be controlled with existing modules in the CRF and thus, the development of a novel inverse kinematics algorithm and a new motion controller module for the CRF will be presented in Chapters 5 and 6, respectively. Chapter 7 is dedicated to the electro-mechanical design and construction of the FCC Robot prototype. Finally, Chapter 8 concludes this thesis with recommendations and some future improvements. The content of each chapter is based on and cites parts of the publications mentioned under *Related Publications* on every title page.

The author was enrolled at Johannes Kepler University, Linz during the entire PhD studies and was employed full time at CERN via the special program *Austrian Doctoral Student Program* [19]. The thesis was placed in the *Beams* (BE) department, under the *Control, Electronics and Mechatronics* (CEM) group in the *Mechatronics, Robotics and Operations* (MRO) section

with the full code BE-CEM-MRO and was requested by FCC Study Board [20] to investigate a possible robotic system for the FCC.

This PhD thesis aimed to meet the academic goals of his home institution while also addressing CERN's main interest to investigate a robotic system within the FCC from a project management perspective that required interdisciplinary coordination, project management, and strategic alignment.

The author was positioned as the scientific secretary for the FCC-ee Technology R&D work group, assuming a pivotal role in shaping the trajectory of research in this work group. The FCC-ee Technology R&D work group hosted meetings with leaders from different work groups to align efforts and identify the most critical open design questions of FCC-ee related to technology.

The author supervised two master theses, one related to the development of a inspection system for Radio Frequency (RF) cavities at École Polytechnique Fédérale de Lausanne (EPFL) [21] and one related to the motion controller development for the CERN Robotic Framework (CRF) at Universidad Politécnica de Madrid (UPM) [22]. Both theses were completed with highest grades at their universities and the students were offered a continuation of their contracts at CERN.

The research activities related to this thesis were kicked off during the development of the CDR by the FCC Study, when no automated system was considered yet to operate in the FCC underground facilities. However, the need for automation of inspection, installation and emergency interventions already became increasingly obvious, triggering FCC Study lead to initiate a PhD thesis to investigate the feasibility and impact of a robotic system in the 100 km long FCC tunnel. After the first feasibility report, presented to the FCC Study by the author, several milestones were defined for the study on a robotic system for FCC, resulting in the following points:

1. Report on feasibility. (complete)
2. Reserve space for the robotic system in the cross section of the FCC main tunnel. (complete)
3. Define a work package for the development of the robotic system. (complete)
4. Establish collaborations with external institutions. (complete)
5. Build a prototype to be integrated in the existing mockup of the LHC tunnel. (complete)
6. Perform a cost estimation to be presented to the FCC board and included in the TDR for the European Strategy for Nuclear Physics. (complete)
7. Install the prototype in the mockup of the FCC tunnel. (mock up to be completed in 2025)

The second milestone centered on integrating a robotic system into the existing layout of the cross-section of the FCC main tunnel for both lepton and hadron collider configurations. The objective was to quickly allocate space for the robotic system, given that developments and integration processes were already well underway in other sections.

The third milestone was dedicated to outlining a comprehensive work package (see Appendix A.3) for the development of such a robotic system. This package aimed to identify the primary challenges, engaging institutions, estimating workforce and material requirements, and proposing potential collaborations with external organizations and universities.

For the fourth milestone, the emphasis was on establishing Memorandums of Understanding (MoUs) with the designated institutions listed within the work package, among them the Johannes Kepler University, Linz.

The fifth milestone was dedicated to the electro-mechanical design of a prototype for the robotic system, intended for initial testing within the existing mock-up of the LHC tunnel. This step was crucial in assessing the capabilities of the CRF and the operator interfaces.

The sixth milestone centered on formulating a comparative cost estimation for presentation to the FCC board. This estimation would subsequently become part of the TDR, a crucial document that will be submitted to the European Strategy for Nuclear Physics. This body holds the authority to determine the feasibility of the entire FCC project.

The final and seventh milestone will revolve around the integration of the robotic system into a mock-up of a 50-meter segment of the FCC main tunnel. This mock-up will encompass the installation of all materials, sensors, and machinery, facilitating the validation of the cross-section layout of the FCC tunnel in 2025.

High-Level Recommendations for a Robotic System for the Future Circular Collider

ABSTRACT. This chapter presents a foundation for future developments related to the robotic system for the Future Circular Collider, including requirements, constraints, and a proposed optimal robotic system. First, parameters and requirements will be defined to which the recommended solution is optimal. These parameters and requirements are then quantified to evaluate and compare various robotic concepts, ultimately leading to the identification of the optimal solution. This work serves as a solid foundation for future robotic system developments, clearly outlining the facts and assumptions on which the recommendations are based. Even if future developments necessitate adjustments, this work will remain a valuable starting point for new developments.

Chapter Content

2.1	State of the Art - Task Specific Robots for big Science Facilities & Harsh Environments .	10
2.2	Current State of Remote Maintenance at CERN	12
2.3	The Future Circular Collider	18
2.4	General Requirements on a Robotic System for FCC	21
2.5	Requirements Deduced from Interventions in the LHC	23
2.6	Recommendations for a Robotic Solution for FCC	25
2.7	Integration in the Official Base Line Design of the FCC Tunnel	32
2.8	Conclusion	33

Related Publications

[23]	Gamper H., Gattringer H., et al.	Design Optimization of a Manipulator for CERN's Future Circular Collider (FCC) - ICINCO
[24]	Gamper H., Mueller A., et al.	A Robotic Concept for Automated Tunnel Maintenance in the Future Circular Collider - RA-M (under review)
[25]	Gamper H., Gattringer H., et al.	A Robotic System for Remote Interventions in the FCC Complex - FCC Week
[26]	Gamper H., Mueller A., Di Castro M.	A Robotic Concept for Automated Maintenance of the Future Circular Collider - Science Robotics (under review)

The chapter provides quantified results and recommendations for an optimal robotic system for FCC. In order to do so, first of all parameters and requirements to which this recommendation represents an optimal solution, will be defined. Then these parameters and requirements will be quantified in order to rate and compare different robotic concepts and finally identify the optimal solution. This work should be a solid basis on which future developments for robotic systems can build on. It demonstrates clearly on which facts, assumptions and requirements the final recommendations are given, such that even if future developments will have to be amended, this work will still be a valuable and solid base to start new developments.

In Section 2.1 a state of the art overview of existing task specific robotic systems in big research facilities is given. Section 2.2 presents the current state of remote maintenance of the whole CERN complex and in the following Section 2.3 the environment of the FCC is described. In Sections 2.4 and 2.5 the requirements for a robotic system will be identified, once based on general requirements and once based on experience from past and ongoing interventions in the LHC. Finally, in Section 2.6 the recommendation for a robotic solution will be presented. In the last Section 2.8 a conclusion and open points for future work will be discussed.

2.1 State of the Art - Task Specific Robots for big Science Facilities & Harsh Environments

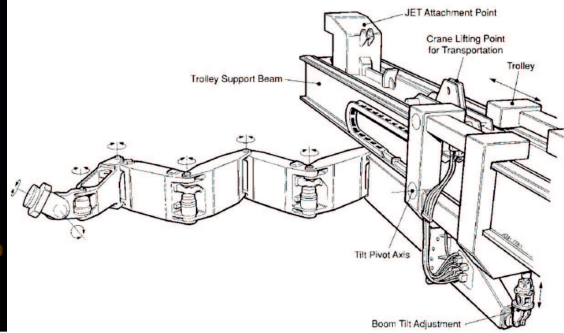
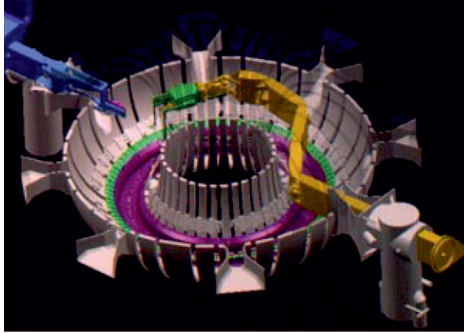
Initially, a comparative evaluation between universal systems and task-specific solutions was conducted. Universal systems find their niche in unstructured environments, where adaptability to diverse terrains and obstacle avoidance are paramount requirements. Achieving these capabilities demands sophisticated locomotion, perception, and cognition, thus necessitating significant investments in development and maintenance. These systems inherently rely on a multitude of software and hardware components. Every additional element introduces potential points of failure and as a consequence impact system availability and robustness.

In contrast, task-specific systems come into play within structured environments, leveraging a priori knowledge of the surroundings to minimize system complexity and optimize efficiency for designated tasks. While such systems may exhibit reduced agility, they shine in terms of efficiency, availability, and robustness.

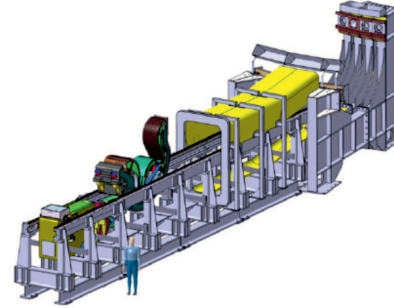
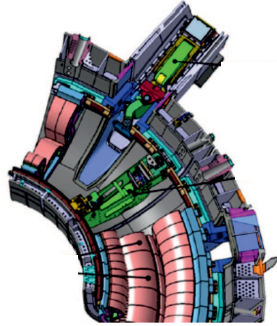
The findings, achieved by insights from market research on automation systems, clearly underscored these principles. Given that the FCC's environment can be categorized at worst as semi-structured, it makes practical sense to design a task-specific system tailored to the FCC's unique needs. Based on this strategic decision, a comprehensive state-of-the-art study was initiated, which focuses on task-specific robotic systems specifically designed for structured and semi-structured environments. Some relevant examples of this study are presented in the following, suggesting directions to a tailored approach to meet the FCC's requirements effectively.

Figure 2.1 shows relevant examples of task specific robotic systems deployed at big research facilities in harsh environments with respect to radiation or confined and complex work spaces that they have to navigate in.

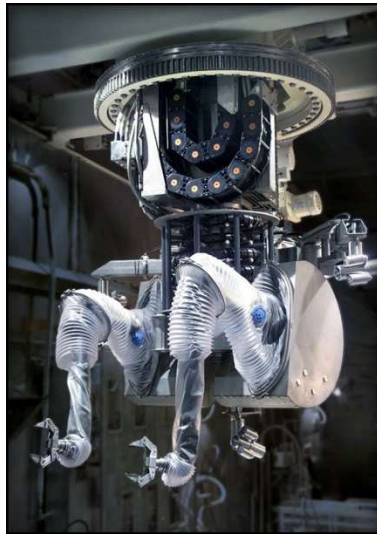
Figure 2.1(a) shows the robotic system at the Joint European Torus (JET) which resembles a task specific robotic system [1]. The JET's robotic system is designed and programmed to



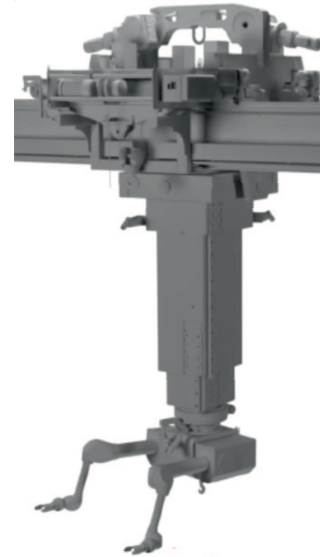
(a)



(b)



(c)



(d)

Figure 2.1: Task specific robots at the large research facilities JET [1] (a), ITER [2] (b), SNS [3] (c) and ESS [4] (d).

navigate and operate in the challenging environment of the tokamak doughnut-shaped chamber. The robot undertakes critical tasks like maintaining and repairing plasma-facing components or inspecting the integrity of the heat shield tiles inside the torus, thus ensuring the functionality of vital fusion components. The robotic system at JET significantly enhances operational efficiency and safety. Looking at the design of the robotic system it becomes obvious that it is heavily tailored to the specific operation in the JET.

The robotic system at ITER (shown in Figure 2.1(b)), the International Thermonuclear Experimental Reactor, again resembles a task specific robotic system [2][27][28][29]. The robotic system plays a pivotal role in facilitating safe and efficient operations in the fusion reactor. ITER's robotic ensemble encompasses a diverse range of specialized machines, each tailored to perform intricate tasks in the challenging environment of the fusion reactor. From maintenance and inspection of critical components within the ultra-high temperature plasma chamber to handling hazardous materials and assisting in the assembly of complex reactor components. The robotic system at ITER not only enhances operational safety but also contributes to the realization of limitless, clean energy through controlled nuclear fusion, ushering in a new era in the quest for sustainable energy solutions.

Figure 2.1(c) shows the robotic system at the Spallation Neutron Source (SNS) represents a cutting-edge fusion of automation and scientific research. Within this world-renowned neutron science facility, this specialized robot [3] operates in an environment where neutron beams are generated and radiation levels are extremely high in the hot cell. These robots play an important role in remotely handling and positioning experimental samples in the SNS instruments, a task that demands high precision and control. This task specific robotic solution enables researchers to conduct experiments with increased safety and efficiency, while collecting data for scientific experiments like materials science, chemistry, biology, and beyond.

Also the robotic system at the European Spallation Source (ESS), see Figure 2.1(d) represents a task specific robotic system [4]. Within this state-of-the-art neutron research facility, these advanced robots serve as critical assets in facilitating complex experiments and sample handling. Their ability to work in high-radiation and extreme temperature conditions is crucial to the experiments. The robotic system at ESS not only ensures the safety of researchers but also allows them to conduct experiments with high precision and efficiency.

Based on the presented examples and state-of-the-art research it soon became clear that a task specific solution needs to be implemented in an efficient and intuitive way, that shows high robustness against the harsh conditions in some of the facilities.

2.2 Current State of Remote Maintenance at CERN

Over the last decade remote maintenance interventions in the CERN facilities have significantly increased. This trend can be explained, by the major advances that have been made in several core disciplines of robotics like perception, cognition and the software frameworks that are being used to integrate hardware and control robots. Another important factor was the development of customized robotic solutions that were tailored to specific needs, which started in 2015. Subsequently, also the awareness for the advantages and capabilities of robotic systems among potential stakeholders at CERN has increased since the MRO section was established in 2014.

The two main reasons for the remote maintenance efforts are safety and efficiency. Many of the research facilities at CERN hold potential dangers for workers like gas tanks causing oxygen deficiency, high power electrics, chemicals or radiation. This is especially true in the underground tunnels with long evacuation ways. Thus, sending robots instead of people to hazardous locations increases the safety and decreases the radiation exposure of workers.

In the following Section 2.2.1 an overview of the whole CERN complex with its main research facilities is given and some examples of the diverse environments are shown. In Section 2.2.2 the CERN robotic fleet and their applications are presented.

2.2.1 Environment

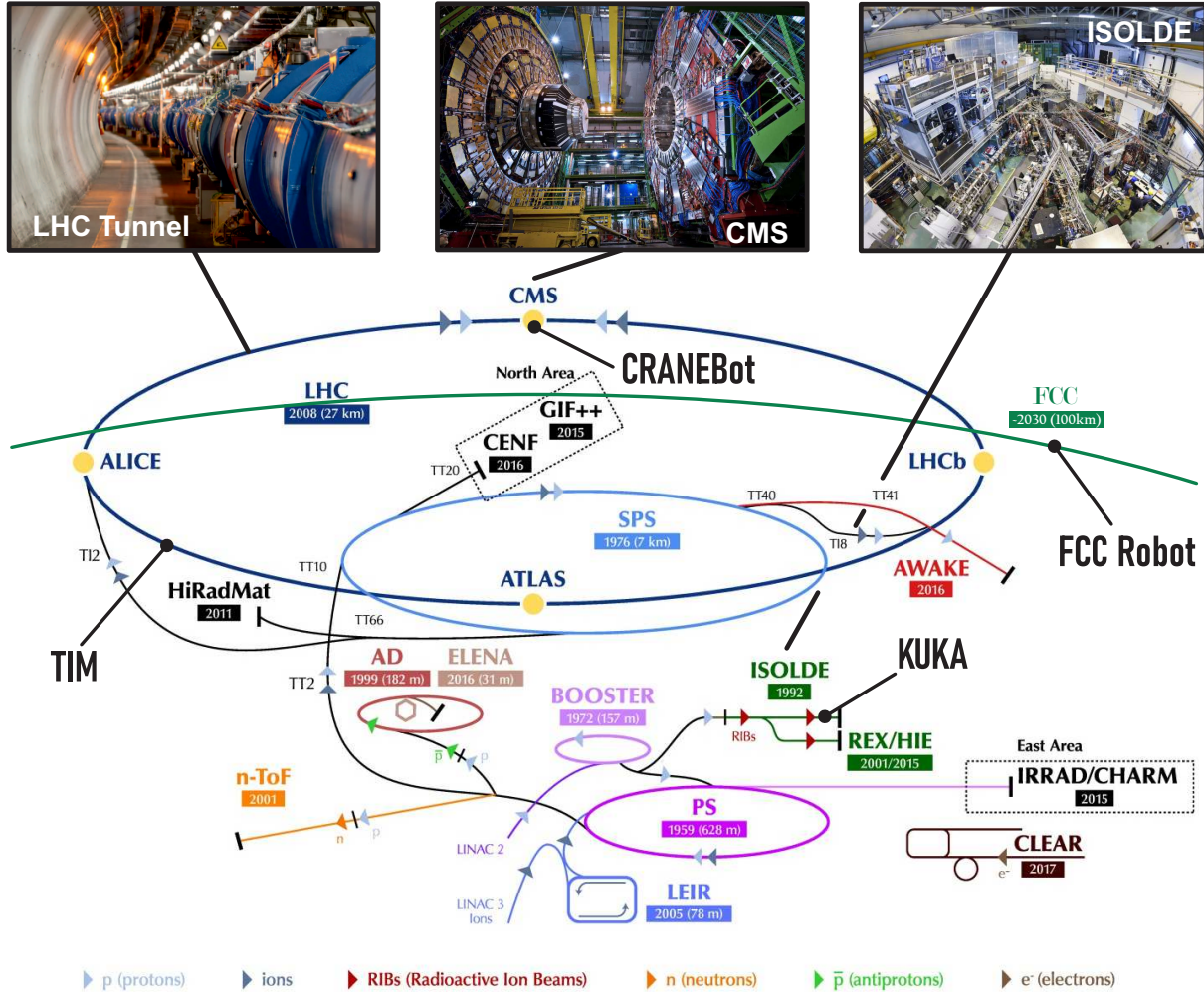
The CERN complex is split up into two main sites which are situated in Switzerland and France and multiple smaller buildings along the LHC ring, serving as control centers for experiments or access points to the tunnel. Overall there are more than 700 buildings. In Figure 2.2 the main experiments and laboratories are shown. Every particle that will be accelerated in the LHC ring has to be pre-accelerated in smaller accelerators and thus started either in LINAC2 or LINAC3, then will be injected into PS, SPS and finally (so far) LHC. The green arc represents the not yet built FCC which will use LHC as a pre-accelerator. The indicated facilities next to this cascade of circular colliders like n-ToF, ISOLDE or North Area are experiments that can request a beam, which will be extracted from the circular colliders and injected in the requester laboratory for other tests.

As already mentioned in the introduction the CERN complex was continuously growing for over 60 years. In the 1950's when CERN was founded, facilities were not designed for robotic interventions. This means that there are no conventions for interfaces and thus facilities use very different installation material, machines, connectors, etc. which heavily increases the complexity of remote interventions. On top of Figure 2.2 some examples of the diverse environments the robots have to operate in are shown. In the experimental cavern at the Compact Muon Solenoid - CMS experiment vacuum equipment had to be installed at the height of the beam line, around 10 m above the ground and 90 m below the surface. In the ISOLDE laboratory radioactive isotope targets had to be exchanged after the experiments and in the LHC tunnel several maintenance or inspections tasks have to be performed along the 27 km long underground tunnel.

2.2.2 Robotic Fleet of the Robotics Section at CERN

In the previous section the many distinct environments at the CERN complex were discussed. Obviously it is impossible, considering current technology, to build one robotic system that would be able to handle all tasks in all environments. Hence, over the last years the robotics section operated many different solutions of which most of them are fully developed at CERN. A subset of the robotic fleet with the most distinct robots is shown in Figure 2.3. The first robots in the section were the commercial solutions Teodor and Telex, see Figure 2.3(a) and (b). The two systems are reliable and robust, but did not allow any adaptation in hard- or software to adjust them to the specific needs of some interventions.

To overcome these limitations, the robotics group started the in-house development of customized solutions, with the goal to facilitate fast prototyping of new universal platforms, see



LHC - Large Hadron Collider // SPS - Super Proton Synchrotron // PS - Proton Synchrotron // AD - Antiproton Decelerator // CLEAR - CERN Linear Electron Accelerator for Research // AWAKE - Advanced WAKEfield Experiment // ISOLDE - Isotope Separator OnLine // REX/HIE - Radioactive Experiment/High Intensity and Energy ISOLDE // LEIR - Low Energy Ion Ring // LINAC - LINEar ACcelerator // n-ToF - Neutrons Time Of Flight // HiRadMat - High-Radiation to Materials // CHARM - Cern High energy AcceleraTOR Mixed field facility // IRRAD - proton IRRADIation facility // GIF++ - Gamma Irradiation Facility // CENF - CErn Neutrino platForm

Figure 2.2: Overview of the CERN facilities and the task specific robots TIM, KUKA and CRANEBot with fixed locations. FCC Robot indicates the yet unknown robotic solution for FCC. The three figures on top show the environment of the LHC tunnel hosting TIM, the experimental cavern CMS hosting CRANEBot and the ISOLDE facility with KUKA manipulators.

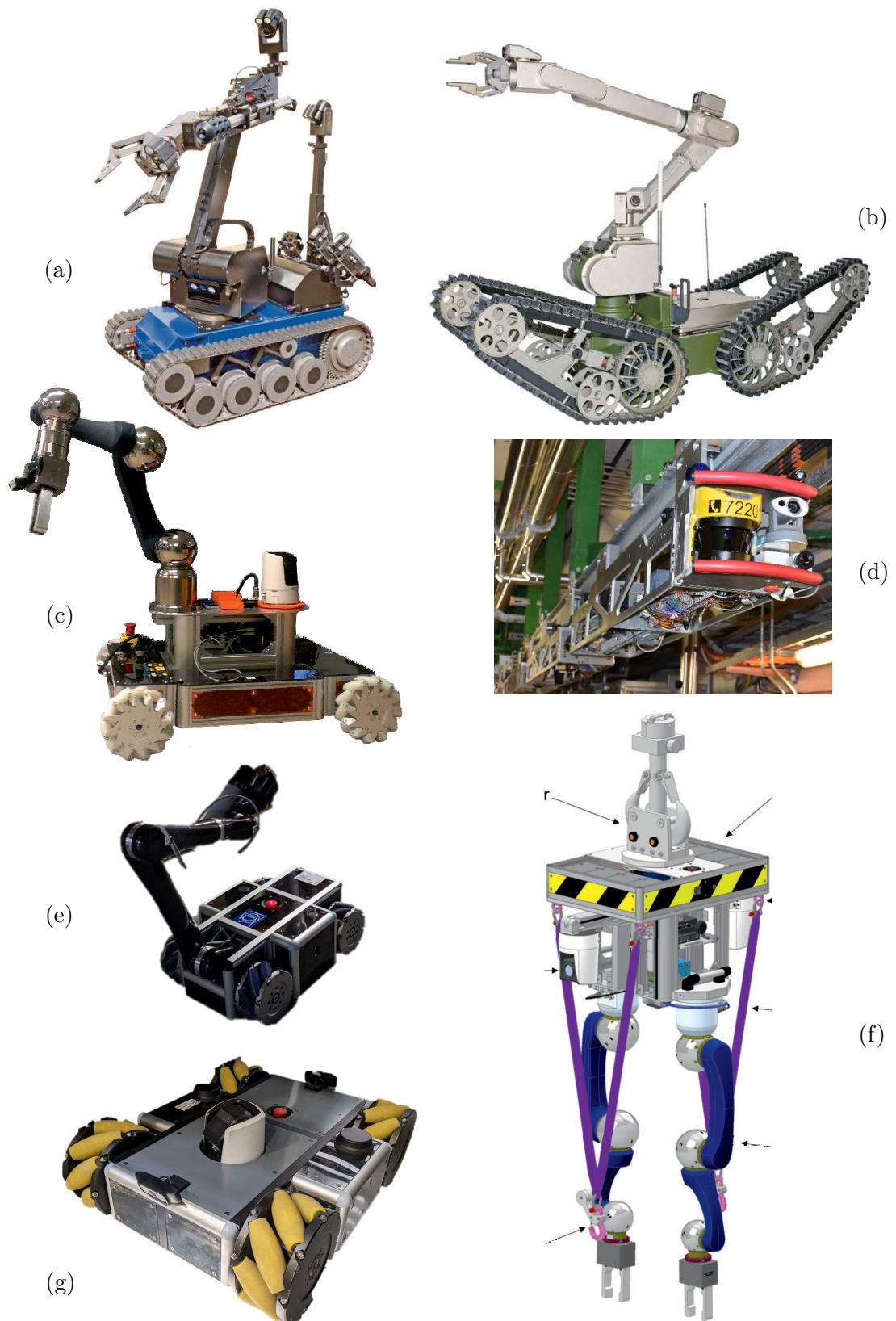


Figure 2.3: A subset of the robotic fleet consisting of Teodor (a), Telexmax (b), CERNBot (c), TIM (d), MIRA (e), CRANEBot (f) and CHARMBot (g).

Figure 2.3(c), (e), (f) and (g). For the 30 km long LHC tunnel a more efficient solution tailored to the environment of the LHC tunnel was required in order to decrease intervention times. As already mentioned before, robotic systems were not considered during the design phase of the LHC, but during the installation phase I-profile beams were installed on the ceiling. These beams were then used as rails for a train like robotic system called Train Inspection Monorail - TIM [14], see Figure 2.3(h).

All robots were designed in a modular way in order to facilitate fast prototyping of robotic systems. Mechanical interfaces of tools, motor and sensor placement and power supply parameters were standardized. The software architecture was designed in a similar fashion. The code is split into modules that can be enabled or disabled depending on the current configuration of a robot. Modules can be hardware integration like robotic arms, grippers, cameras or sensors as well as algorithms like control strategies, state estimators or trajectory planners. The software framework CRF and is usually deployed on Linux systems with preempted real-time kernels.

The increased modularity of a robot often comes with the price of reduced robustness. The question if an universal or task specific system should be used for a certain combination of tasks and environment is not a trivial one. The correlation of modularity and robustness as well as its impact on the usage of robots is well visible in the following Figure 2.4. In Figure 2.5 customized

	<div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 10px;">← modularity</div> <div style="margin-right: 10px;">—</div> <div style="margin-left: 10px;">robustness →</div> </div>							
	<i>CERNBot</i>	<i>CHARMBot</i>	<i>Teodor</i>	<i>Telenax</i>	<i>MIRA</i>	<i>CRANEBot</i>	<i>TIM</i>	<i>Kuka</i>
LHC	✓	✓	-	-	-	✓	✓	-
SPS	✓	✓	-	-	✓	-	-	-
SPS north	✓	✓	✓	-	✓	✓	-	-
n-ToF	✓	-	✓	✓	-	-	-	-
ELENA	✓	-	-	-	-	-	-	-
LINAC3	✓	-	-	✓	-	-	-	-
LINAC4	-	-	-	✓	-	-	-	-
ISOLDE	-	-	✓	✓	-	-	-	✓
CHARM	✓	✓	✓	-	-	-	-	-

Figure 2.4: Intervention sites for each robot. Robots are ordered depending on their modularity and robustness. The amount of different sites the robots have been used at, correlate with increasing modularity.

robotics solution are shown during interventions in the environments shown in Figure 2.2.

2.2.3 Summary & Comparison

Modularity is important for fast prototyping for upcoming interventions but increases the complexity of the robots and thus decreases robustness. Due to the fact, that robot friendly design was not considered at an early design phase of the CERN facilities, many distinct robotic solution are necessary to handle the required tasks at the CERN complex. This comes with higher development efforts and less robustness compared to an integrated task specific robotic system in a well defined environment. Thus, a key point for the development of the FCC tunnel from a

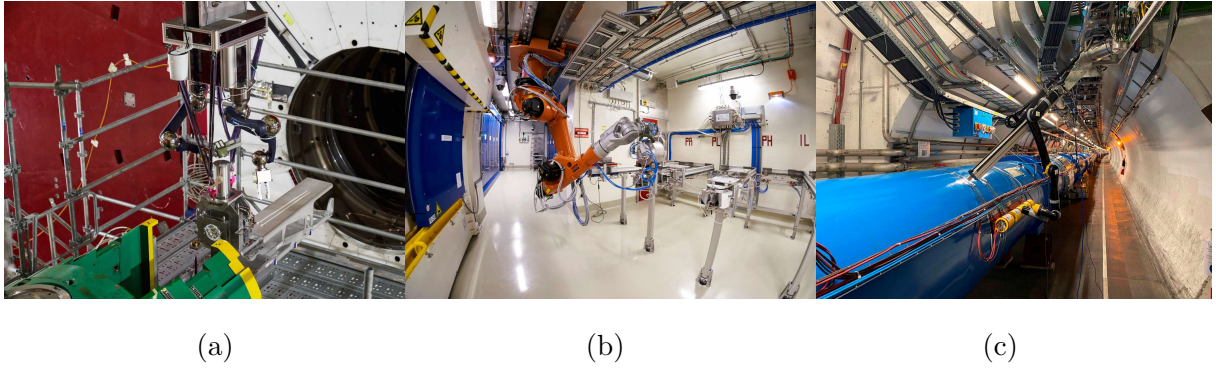


Figure 2.5: Robots in the example environments from Figure 2.2. CRANEBot in the experimental cavern CMS (a), Kuka in the laboratory ISOLDE (b) and TIM in the LHC tunnel (c).

robotics perspective is to consider robotic friendly design at an early design phase. This would subsequently allow the design of a task specific system with increased robustness, higher efficiency and less development costs. To support this, the MRO section started a document called "Remote Maintenance - Code of Practice", which provides guidelines for the design of interfaces and procedures for robotic or remote interventions. Similar efforts have been made by other big research facilities and were discussed in the previous section.

2.3 The Future Circular Collider

Many of the requirements in Section 2.4 on the robotic system originate in the specific environment of the Future Circular Collider. Thus, it is important to clearly understand the possibilities and restrictions in this underground facility, shown in Figure 2.6. For better visibility and due to the long distances between the access points this drawing is not to scale.

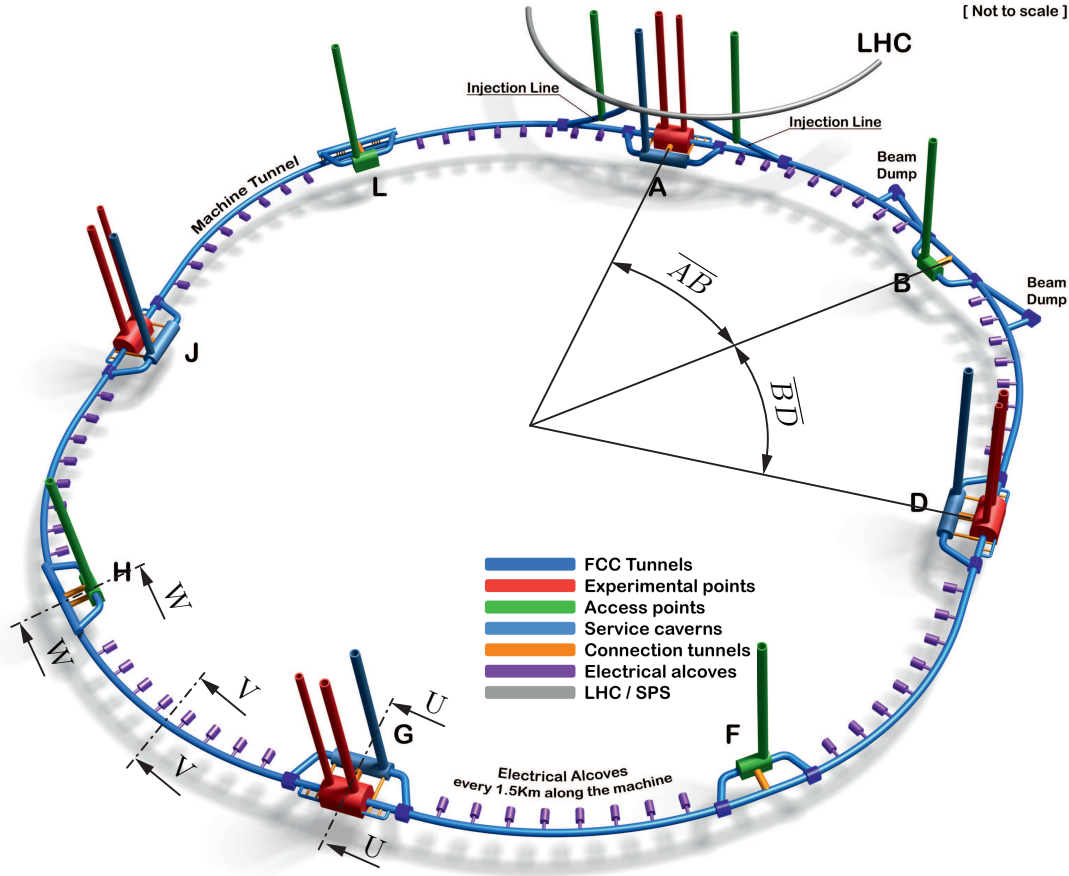


Figure 2.6: The Future Circular Collider

The blue machine tunnel has a circumference of around 100 km (different studies are currently working with slightly different numbers). The tunnel lies around 100 m underground at its deepest point and the ring plane is not perpendicular to the gravitation vector but slightly tilted, such that the first solid rock layer can be used as a solid foundation. Unlike the name Future Circular Collider would suggest, the blue machine tunnel does not perfectly resemble a circle. The circular segments of the tunnel exclusively host dipol magnets, whose only purpose is to provide a magnetic field, that creates the necessary radial acceleration, which will keep the particles and beam bunches on a circular path. Since the dipol magnets physically completely surround and cover the beam line, it is technically and logistically not feasible to control other properties of a particle beam at the same time or at the same position of the tunnel. Therefore, the FCC contains several linear segments in which no radial acceleration and thus no dipol magnets are needed. These straight sections are e.g. used for beam optics like focusing or cleaning particle bunches, forward acceleration of the particles in order to increase their energy using RF cavities or the particle collisions in the experimental caverns. The forward acceleration is applied at only one point of the machine, which is either point H or L depending on the cross-

section configuration (see Section 2.3.1). The particle collisions occur at the four experimental caverns in points A, D, G and J, visualized in red in Figure 2.6. The green and blue caverns are used during installation or maintenance tasks for material transport and serve as access shafts for workers and are called access or service caverns, respectively. The electrical alcoves, shown in purple, are shielded from radiation and host radiation sensitive electronics. The distances between the points A, B, D, ..., J, L namely \overline{AB} , \overline{BD} , ..., \overline{JL} = 11.4 km, as shown in Figure 2.6 and sum up to a total circumference of 91.2 km. Fire or section doors divide the tunnel into around 400 m long segments in order to contain a hazard like fire or gas in a single section.

Over its life time the FCC will be operating as a lepton and a hadron collider, called FCC-ee and FCC-hh. This will require two different layouts of the FCC machine tunnel cross-section. The tunnel and cavern cross-sections will be discussed in more detail in Section 2.3.1. The radiation situation in different areas of the FCC complex will be discussed in Section 2.3.2.

2.3.1 Layout of the Cross-Sections

In Figure 2.7(a) and (b) the two different layouts for the machine tunnel FCC-ee and FCC-hh are visualized. The FCC machine tunnel has a diameter of 5.5 m and the distance from floor to

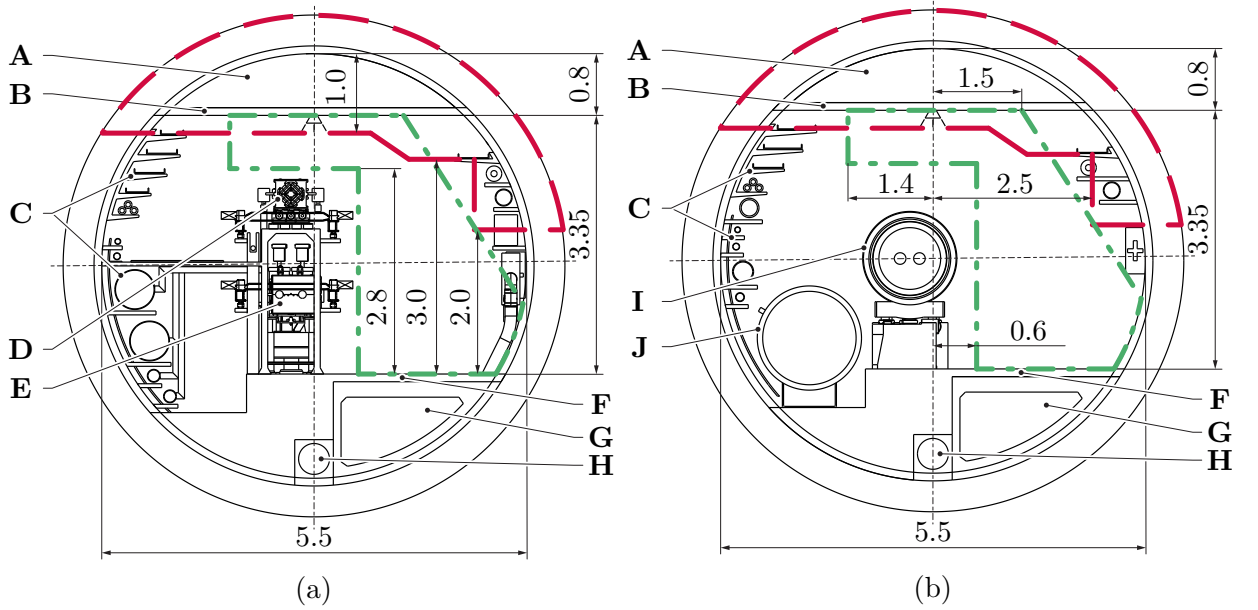


Figure 2.7: Cross-Section V-V from Figure 2.6: The FCC machine tunnel. Configuration FCC-ee (a) and FCC-hh (b). Dimension are in meters.

ceiling is 3.35 m. The ceiling **B** creates a separate compartment **A** on top of the tunnel which is used as smoke and helium extraction duct in case of an emergency. **C** represents cables trays and pipes for various kinds of material. **F**, **G** and **H** are the floor, the fresh air and ventilation duct, respectively.

For the FCC-ee configuration the booster **D** is situated on top of the magnet **E** and for better accessibility for maintenance and inspection tasks, an additional floor on the left side will be installed.

For the FCC-hh configuration all magnets have to be cooled down to supraconductivity at 1.9 K in order to apply currents of 12.5 kA to provide the necessary magnetic fields. To limit the cost during operation, the cooled down region was design as compact as possible and thus all the beam intercepting devices are installed in the pipe **I** and **J** represent the installations for cooling.

The area marked with a red (- -) line represents the area in which it is allowed to permanently install material that could somehow be used for a robotic system. This area is restricted by necessary installations on both tunnel walls, the floors that need to be free for e.g. maintenance and obviously the space occupied by the machine in both configurations. The area marked with the green (-.-) line shows the area in which it is possible for a robotic system to move along the tunnel axis, but material can not be installed permanently. This area is restricted by certain areas that need be free for workers during maintenance operations, the ability to pass the fire and section doors and a certain volumetric through put in **A** that can not be decreased by the footprint of a robotic system.

Figure 2.8 shows the cross-section U-U as indicated in Figure 2.6. At this point the FCC machine tunnel is disrupted by the experimental cavern **K** and the cylindrical detector **L**, whos axis coincides with the FCC machine tunnel axis. The connection tunnels **M** lead to the service cavern **N**. This cross-section is identical for the points A, D, G and J of the FCC complex in Figure 2.6.

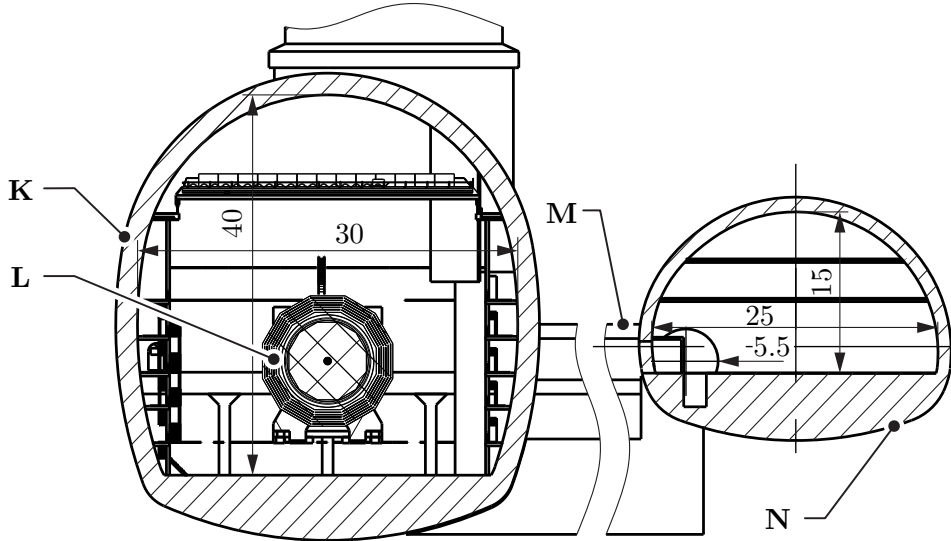


Figure 2.8: Cross-Section U-U from Figure 2.6: K - Experimental Cavern, L - Detector, M - Connection Tunnels, N - Access Cavern. Dimension are in meters.

Figure 2.9 shows the cross-section W-W as indicated in Figure 2.6. This cross-section lies in one of the straight sections of the FCC tunnel in which the particles experience a forward acceleration. This acceleration is achieved by RF cavities, which create an oscillating electric field that pushes charged particles forwards. The RF cavities are installed in the FCC machine tunnel **Q** and the required electrical installations are housed in the radiation safe clystron gallery **O**, which are connected via the tunnel **P**. The clystron gallery as well as the machine tunnel are connected via the connection tunnels **R** to the access caverns **S** which show a diameter of 24 m in order to manipulate the up to 12 m long dipol magnets during the installation phase.

This cross-section is identical at the points H and L in Figure 2.6 and the access caverns **S** can also be found at points B and F of the FCC complex.

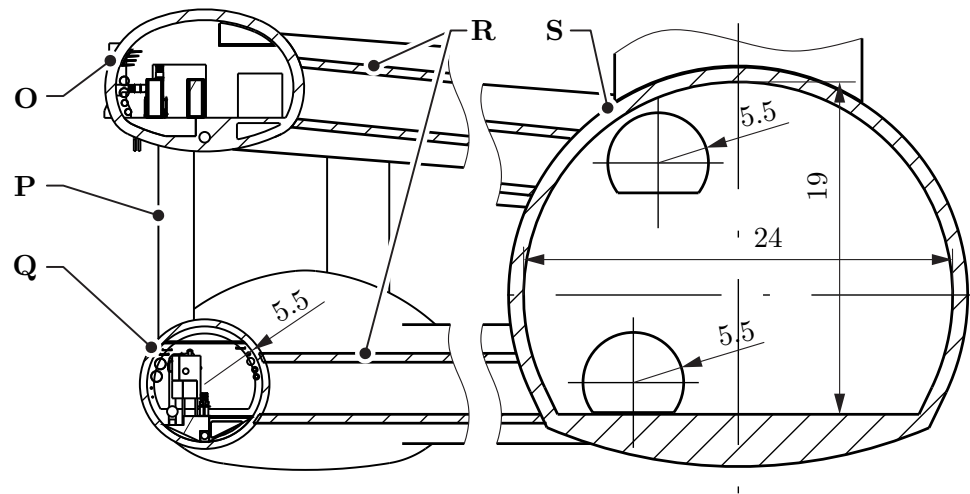


Figure 2.9: Cross-Section W-W from Figure 2.6: O - Clystron Gallery, P - Electric installation for RF, Q - Main Tunnel, R - Connection Tunnels, S - Service Cavern. Dimension are in meters.

2.3.2 Radiation

During the operation of the FCC-ee collider, synchrotron radiation (SR) is the main contribution for radiation load effects in the collider environment. SR is electro-magnetic radiation, emitted tangentially by a charged particle following a curved trajectory.

The SR power, which is at 50 MW/beam in FCC-ee, impacts the collider components and the tunnel environment. The average ionising dose in the FCC machine tunnel is expected to reach values of a few hundred kGy/year, above and below the collider ring and up to 1 MGy/year in close vicinity at beam level. As these levels are unsustainably high, mitigation measures, like additional shielding or well protected alcoves and caverns (see Figure 2.6), to reduce the radiation impact on sensitive components are required. Due to the spectrum of the SR, the activation of material is expected to be fairly low and access should be granted quickly after shut down of the machine [30].

To mitigate the effect of radiation on sensitive installations several areas that are shielded from radiation, like the service and access caverns, alcoves (see Figure 2.6) or clystron gallery (see Figure 2.9 - O) have been designed. These shielded access points also allow workers to go closer to the tunnel for maintenance work without being exposed to radiation.

2.4 General Requirements on a Robotic System for FCC

This section summarizes the general requirements on a fully integrated and task specific robotic system. The presented wishes and requirements on a novel robotic system in the FCC are mainly required by different engineering groups at CERN that are responsible for certain installations

Table 2.1: General requirements on a robotic system for FCC.

Index	General Requirements
G1	Fully Autonomous Interventions
G2	Detect Hazards
G3	Robust Control
G4	Low Maintenance
G5	Reliable/Redundant Power Supply
G6	Robust Communication
G7	Intuitive HR Interface
G8	Dexterity in Maneuverability
G9	No Further Contamination by Robot
G10	Low Cost

in the tunnel and thus might need to perform some maintenance and repair work or testing of their systems. Due to the sheer size of the FCC (~ 100 km) simple tasks, that have been conducted manually in the existing facilities, can become complex problems in terms of logistics or are simply not feasible in terms of time or costs.

As it was already discussed in Section 2.1 the most efficient and robust robotic systems are task specific systems, which were considered during the design phase of the facility they will be operating in. Therefor the vision for the FCC is, to integrate one system that can reduce the necessary amount of different robotic solutions to a minimum. The requirements on such a system are listed and enumerated from G1 to G10 in Table 2.1 and discussed in the following.

The robotic system should be able to conduct all planned and repetitive interventions fully autonomously. For unplanned interventions an operator should be able to send the robot fully autonomously to a point of interest at which the operator can take over in tele-operation mode, whenever human cognition is required. The required workspace of the robotic system is dictated by the cross-section layout of the FCC main tunnel as presented in Section 2.3.1 and comes down to $3.35 \times 5.5 \times 10^5$ m. This forms the general requirement G1. The system should be able to detect hazards to workers like power cuts, oxygen deficiency or fire (G2). The availability of the robotic system is key to achieve the high level goals for FCC as mentioned in the introduction for maintenance works and especially for emergency interventions, leading to G3, G4, G5 and G6. Obviously the financial impact of such a system should be low, corresponding to G4 and G10. An intuitive human-robot interface (G7) will be important when the robot is being controlled in tele-operation mode. Even though the goal is a high level of automation, some unplanned interventions will still (at least within the next decades) require a human operator. Furthermore it is crucial for the operator to clearly understand the current situation and configuration of the robot to facilitate efficient interventions e.g. via Augmented Reality (AR) interfaces and for mechanical interactions with the environment during teleoperation it is important to provide an intuitive haptic feedback to the operator. Since the FCC tunnel resembles a cluttered and semi-structured, hence complex environment the robot needs high dexterity in maneuverability (G8). Finally, for requirement G9, the robot should at no point increase the radiation contamination by its interaction with the environment.

The presented requirements will be quantified and used in Section 2.6 to rate and compare different robotic solutions. The next Section 2.5 will discuss specific tasks that a robotic system will have to perform, and the derived requirements corresponding to these tasks.

Table 2.2: Past and ongoing interventions in the LHC and the derived requirements for a robotic system for the FCC.

	Sec.	Task Group	Derived Requirements
Maintenance	2.5.1	Measuring Radiation; Monitor Tunnel; Integrity	M1: Cover full work space; M2: Stable movement along tunnel axis; M3: Pass Fire Doors; M4: Robust Collision Avoidance;
	2.5.2	Test BLM Sensors	M5: High Dexterity Manipulator; M6: Autonomous operation; M7: Operate in cluttered work space;
	2.5.3	Measure Oxygen; Measure Alignment; Audio Inspection; Vacuum Leak Detection; General Interventions	M8: Specific Tools; M9: Tool Changer; M10: Fast Interventions; M11: Modularity; M12: Precise Teleoperation with Haptic Feedback;
	2.5.4	Docking Station;	M13: Payload $\sim 15\text{kg}$
	2.5.5	Transport Tools/Material;	M14: Payload $\sim > 50\text{ kg}$
Emergency	2.5.6	Guide People; Follow Monitor People; Search for Life;	E1: Not Blocking Emergency Ways; E2: Specific Tools (Infrared Camera, Radar); E3: Move in Harsh Cluttered Environment
	2.5.7	Locate Approach Fire; Extinguish;	E4: Robot Speed $\sim 34.2\text{ km/h}$; E5: Specific Tool;

2.5 Requirements Deduced from Interventions in the LHC

The conceptual design studies concerning the FCC machine are published in [16] and [9] and provide detailed reports on the hyperparameters, dimensions and necessary installation of such a 100 TeV machine. Comparing these reports with the current LHC design and layout, shows significant similarities concerning the environment and installation in the tunnel. Thus, it is expected that many of the robotic interventions that are currently performed in the LHC machine will also be required for the FCC machine. Therefore all past and ongoing robotic interventions from more than ten years of robotic operations in the LHC tunnel were collected and are presented in this section. To stay within the scope of this article, the focus will be put on the requirements that arise from these operations and will not cover all details or causes of the single interventions. The interested reader is referred to the citations found in the corresponding sections. In order to provide a complete but still comprehensive list, interventions were grouped into task categories (see Sections 2.5.1-2.5.7) that lead to similar derived requirements on a robotic system. Table 2.2 lists these task groups, corresponding interventions and derived requirements. The interventions are split up into Maintenance and Emergency tasks and enumerated from M1 to M14 and E1 to E5, respectively. Relevant details about the tasks and interventions are discussed in the following sub sections.

2.5.1 Measuring Radiation, Monitor Tunnel Integrity

Inspections with different sensor are conducted in the LHC tunnel for early detection of potential problems or to ensure safe access for workers.

Measuring Radiation [14]: Industrial radiation sensor measures the radiation at beam level along

the 30 km long tunnel. The sensor is carried by an extendable arm of TIM, see Figure 2.3(d). This is mainly done to estimate the risk of sending people to the tunnel, estimate the cool down phase of the LHC or use this information as indication for other problems.

Monitor Tunnel Integrity [31]: Cameras and 3D Cameras mounted on mobile platforms are taking pictures of the tunnel walls and mapping them using SLAM techniques. 3D Cameras check the geometry of the tunnel and RNNs detect optical changes at the tunnel walls (e.g. new cracks). This is necessary since the huge underground facilities and especially the heavy experimental caverns are slowly drifting (~ 0.15 mm/year) [32], which could lead to structural instabilities.

2.5.2 Calibration of BLM Sensors

The Beam Loss Monitor sensors are 0.5 m long cylindrical radiation sensors of which around 3500 are installed in the LHC tunnel, see [13]. The sensors are able to detect particles that are deviating from the desired trajectory. Events like this are dangerous because they will increase the activation of material around the beam line and can lead to destruction of equipment. Thus, the BLM sensors can trigger a shutdown of the LHC and therefor have to be tested thoroughly. During the testing procedure a radio active source is brought in near proximity of all 3500 BLM sensors and the correct response will be checked in the CERN Control Center - CCC. Following a map of the BLM sensors the robot TIM (see Figure 2.3(d)) is approaching the sensors and detecting their exact positions using RNNs, taking the radio active source out of its shielding, extending the 9 DoF robotic arm and bringing the probe close to the sensor.

2.5.3 Measure Oxygen, Measure Alignment, Audio Inspection, Vacuum Leak Detection and General Inspection Tasks

These tasks basically require a exploitation of the whole tunnel environment with different tools. The goal is again to mitigate technical problems and hazards for workers. The oxygen measurement is performed to detect a hazardous deficiency before workers enter the tunnel. The audio inspection is performed for general detection of problems like e.g. assigning abnormal sounds to certain machine parts that move with a certain frequency by applying the Fourier transform to the recorded data.

The Vacuum Leak Detection will be started, when a vacuum pump fails to achieve the ultra-high vacuum of 10^{-9} bar in the responsible area of about 100 m [33]. The robot will flood this tunnel section with helium and if the vacuum pump system detects helium, the leakage search will be continued in this section. The robot will then continue to locally apply helium on delicate parts like connectors or flanges until the vacuum pump again detects helium and thus, the precise location of the leak was detected. Unrecognized vacuum leaks can lead to fatal destruction as it happened in the LHC tunnel in 2008, see [34].

The Alignment Measurement is necessary to ensure a high precision alignment of the axes of the cryostats to avoid potential beam loss. To measure the alignment, a system of fiducials on the cryostats has to be detected and the relative position between consecutive elements will be measured, see [35]. The subsequent alignment process and algorithms are described in [36] and [37].

2.5.4 Coupling of Docking Stations

Experimental caverns like CMS are so heavy that they sink over time (in LHC so far 4mm [32]) which pulls also pulls down the beam pipe and brings it out of alignment. In order to adjust the alignment of the beam pipe motors are installed at the girders. However, radiation sensitive control hardware can not be installed permanently and has to be brought in place by a robot and plugged like a docking station.

2.5.5 Transport Tools/Material

A robotic system should be able to carry heavy tools and material that might possibly be needed for installation or repair work to support workers. This will result in relatively high payloads for the robotic system. Furthermore the robotic system should support workers with the scope of human-robot collaboration whenever physically demanding tasks are involved.

2.5.6 Guide People, Follow & Monitor People, Search for Life, Patrols

In cases of emergency a robotic system should be able to find, guide and rescue people from the underground facilities. In many situations the rescue will have to be done in challenging conditions like open fire, smoke or gas exposure in the tunnel. Cameras detect the location of a person and follow this person, while radar measures the heart rate of this person, to monitor the health. Currently, always two people have to go to the tunnel together and in future it could only be one person, followed by the robot, which monitors this person's health. In order to make sure no personnel is present in the tunnel before the machine will be switched on, patrols are being performed and if people are encountered they would be sent out of the tunnel, which should be taken care of by several robotic systems.

2.5.7 Locate & Approach Fire, Extinguish Fire

A robotic system should be at any place in the tunnel within a max. time of 10 min in case of emergency and detect and approach fire fast to take first actions and thus minimize the damage. Furthermore the system should be able to be controlled in teleoperation mode such that rescue teams, like the CERN Firebrigade, can remotely assess the situation before physically going there.

2.6 Recommendations for a Robotic Solution for FCC

In order to provide a recommendation for an optimal robotic system, first of all requirements to which this recommendation represents an optimal solution had to be defined. Based on restrictions derived from the FCC environment (Section 2.3), a state of the art analysis for task specific robotic systems (Section 2.1) and experience from remote interventions in the LHC

tunnel, such requirements were defined. This lead to the general requirements G1-G10 (Section 2.4), maintenance requirements M1-M14 and emergency requirements E1-E5 (Section 2.5). In this section these requirements will be quantified whenever possible to compare different robotic systems and identify the best (optimal) solution. The following recommendations are written in a way, to preserve maximum freedom for future innovative developments, while narrowing down the set of possible systems to a minimum, according to the presented requirements.

The above mentioned requirements allow recommendations for locomotion, manipulator, end-effector, power supply, emergency system and integration, as presented in Sections 2.6.1 - 2.6.6. However, they are not detailed enough for recommendations about perception, cognition, localization, motion control or actuation of the robotic system.

2.6.1 Locomotion

In this section a rating catalog is presented, that allows to compare different locomotion solutions and rate them based on how well they fulfill a subset of the discussed requirements that are related to locomotion, see Figure 2.10. In the first column different concepts are listed and in the first row related requirements were grouped together in Geometry, Power Supply, Communication, Maneuverability, Radiation, Control, Emergency, Maintenance and Costs. The specific requirements are stated below, together with the labels G, M or E that allows to trace back the requirement to its origin. The impact or importance of the requirements are represented with the weighing factor. In the tabel ratings from 1 (bad) to 4 (good) are given for how well a locomotion solution fulfills the corresponding requirement. In order to identify the best solution, the rating values are multiplied with the weighing factor and summed up for each locomotion solution, which leads to a final rating (in percent of the maximum achievable value) shown in the most right column in Figure 2.10. The green and red cell filling visualize where the corresponding solutions perform very good or bad.

As shown in the rating catalog, the Rail Guided solution performs the best with respect to the identified requirements and hence will be recommended as the optimal solution for locomotion in the FCC tunnel. To stay within the scope of this article, not all of the ratings can be explained in detail, but the major advantages and disadvantages of the optimal solution should be discussed in the following.

A first big difference in the ratings can be seen for requirement M2: A rail guided solution facilitates very stable and robust movements along the tunnel axis, whereas mobile platforms, areal or legged robots are more prone to instabilities. This is of course directly related to the maximum speed at which such a system can safely travel through the tunnel and thus impacts E4.

Index Requirement	Geometry Req.			Power S.		Comm.		Maneuv.		Rad.	Control			Emergency			Maintenance				Costs					
	M1	M3	M4	G5	G1	G7	G6	M2	G8	G9	M12	G3	M6	E1	E3	E4	M11	M14	M10	G4	G10					
	Workspace ~3200x3600	Go through fire doors	Robust Coll. avoidance	Reliability / Redundancy	Autonomy	Intuitive HR Interface	Robust	Stable Movement	Dexterity	No further cont. by robot	Hapt./Precise Teleop.	Robust	Autonomous	Not Blocking Exit	Move in harsh env.	Intervention	Time in CoE	Modularity	Payload	Fast Intervention	Few Maint.	For robot. Construction	Material Env.	Sensors	Actuators	Construction Material
Weight	1	0.8	1	1	0.6	1	1	0.8	1	1	0.8	1	0.8	1	0.8	1	0.6	0.8	0.8	0.8	1	0.8	0.8	1	Summed up Rating in %	
Hol. Mobile Robot	2	3	2	2	4	2	3	1	4	3	4	4	1	1	1	1	4	2	2	2	3	4	4	3	53	
Non-Hol. Mobile Robot	2	3	2	2	4	2	3	3	2	3	4	4	1	1	1	2	4	2	2	2	3	4	4	3	54	
Rail Guided Robot	3	3	3	3	2	3	3	4	4	4	3	4	2	4	3	4	3	3	4	3	1	4	3	3	67	
Aereal Robot	4	3	1	2	3	2	2	1	4	1	2	2	1	3	4	3	1	1	3	2	4	2	4	4	51	
Legged Robot	3	2	2	2	2	2	2	1	4	3	1	2	2	1	3	2	4	2	2	1	4	1	1	1	45	
Legged Robot + Wheels	3	2	2	2	2	2	2	2	4	3		2	2	1	3	3	4	2	3	1	4	1	1	1	48	
Robot in Hyperloop	2	3	2	2	2	2	3	4	4	3	2	4	2	1	1	4	2	2	4	2	1	3	3	3	53	

Figure 2.10: Rating Catalog for Locomotion Solutions

Having a robust and stable base of the robotic system, also effects the precision of tele-operation M12 and the maximum payload M14, which is critical for e.g. the transport of material or tool changing systems. An important requirement that basically prohibits the usage of aerial vehicles is G9. Drones move the fluid that they are operating in with relatively high speeds to move forward, which would blow up contaminated dust in the tunnel and therefor increase radiation levels and increase the risk of contamination for workers. A general problem for ground based systems like the wheeled and legged platforms are that they are occupying a significant amount of space of the emergency ways and thus could potentially block them completely and prohibit workers to escape the facilities in case of e.g. oxygen deficiency or fire. Furthermore, rescue missions are harder to plan and more likely to fail for ground based systems, when debris or other obstacles are limiting accessibility as it happend after the vacuum leak incident as described in Section 2.5.3.

2.6.2 Manipulator

The requirements of Section 2.4 and 2.5 also demand a physical interaction with the environment and hence a manipulator. This section discusses an optimal manipulator to fulfill these requirements. In order to compare the immense amount of different manipulators and narrow down the possible choices for FCC, two dimensionless units, namely the relative dexterity d_r and the relative payload p_r will be introduced.

The relative dexterity $d_r = w_s/r_s$ puts the maximum workspace w_s of the robotic solution in relation to the robot space r_s , with r_s describing the space which is necessarily occupied by the robot to reach all positions and orientations in the workspace w_s . The relative payload $p_r = p_l/r_w$ puts the average payload p_l in relation to the weight r_w of the manipulator.

The results for d_r and p_r will be grouped into the three classes < 1 , ~ 1 and > 1 , as shown in Table 2.3 and allows for an abstraction in five distinct manipulator classes (I)-(V). This of course is not a complete list, but supports a comprehensive discussion and high level recommendations of an optimal manipulator topology. Class (I) in Table 2.3 contains parallel robots like e.g. hexapods, class (II) contains common industrial robots with a Degree of Freedom $DoF = \{6, 7\}$, high stiffness and hence high robot weight. Class (III) contains cable driven robots that also show a closed kinematic chain, but bigger workspace and class (IV) contains highly redundant, serial robots with $DoF \geq 8$. Finally class (V) contains serial robots with the highest DoF often also referred to as snake like robots.

In order to narrow down the possible manipulator choices for FCC, the requirements found in previous sections, can now be mapped onto lower bounds for d_r and p_r and thus feasible manipulator choices can be identified. In order to cover the full work space of the FCC machine tunnel (M1), which also resambles a complex, cluttered work space (M7) that requires obstacles avoidance, it is necessary that $r_s < w_s$ and thus $d_r > 1$. This condition is only satisfied by manipulator classes (IV) and (V). Furthermore these classes also comply with requirement M5 for dexterous manipulation during interventions. Additionally, class (IV) allows for a relatively low footprint of the robotic system when moving along the tunnel axis, which is a great advantage considering that the system has to pass through fire and section doors, see requirement M3. Considering the necessary payload M13 in combination with the recommendation for locomotion of the previous section (rail based solution) a relative payload of at least $p_r \sim 1$, hence $\{p_r \sim 1, p_r > 1\}$ is desired. Both, $d_r > 1 \wedge \{p_r \sim 1, p_r > 1\}$ are only satisfied by manipulator class

Table 2.3: Relative dexterity and relative payload for different manipulator topologies.

Class	(I)	(II)	(III)	(IV)	(V)
Rel. Dext.	< 1	~ 1	~ 1	> 1	> 1
Rel. Payload	~ 1	~ 1	> 1	~ 1	< 1
Example	Parallel	Industrial	Cable Driven	Highly Red.	Hyper Red.

(IV), see Table 2.3. This leaves one requirement M14 unsatisfied, since a manipulator class in $\{d_r > 1, p_r > 1\}$ does not exist. This can be accounted for by integrating an additional winch on the base of the robotic system, which thus facilitates the handling of heavy payload.

Summed up, only the topological class (IV) is able to satisfy the requirements on robot dexterity and relative payload, necessary for all operations in the FCC machine tunnel. Thus, it is recommended to integrate a highly redundant manipulator with a $DoF \geq 8$ in the robotic system and an additional winch on the base of the system for high payload manipulation.

2.6.3 End-Effector

Considering the many diverse tasks that the robotic system will have to manage and especially unplanned interventions that might require improvisation of the operator it is recommended to integrate a tool changing system on the robot. This addresses and satisfies requirements M8 and M9.

2.6.4 Power Supply

During normal operation, the power supply should be established via the rails of the robotic system. Since the system will be used in emergency cases with possible power cuts, it should be equipped with a battery system for self-sufficient operations.

2.6.5 Emergency System

In Sections 2.5.6 and 2.5.7 several emergency situations in which a robotic system should be able to intervene were presented. One of the main challenges for emergency interventions is the required speed of a robotic system of roughly 35 km/h in order to reach every point in the tunnel within 10 min. Furthermore, the availability of an emergency system is obviously critical for successful operations. Considering only one robotic system the availability would be significantly decreased, especially while the robot is used for maintenance work or material transport, during which the system would not be able to move to the location of the incident.

Summed up, it is recommended that the robotic system should consist of two independent robots, a main system for maintenance, inspection and material transport and an emergency system for fast inspections without a manipulator. Obviously it is necessary that the emergency system is able to bypass the main system.

2.6.6 Integration

When the study for a robotic system in the FCC tunnel was started, the CDR for FCC-ee [16] and FCC-hh [9] were already finished and thus, preliminary layouts of the tunnel cross-sections, as presented in Section 2.3.1, were already designed. This section provides recommendations regarding the integration of the robotic solution in the FCC baseline layout, with respect to the rail system (as a result of Section 2.6.1) and possible radiation safe parking spots during the operation of FCC.

Rail Placement

As already discussed in Section 2.3.1 and visualized in Figure 2.7 the layout of the machine tunnel allows the permanent installation of material on the upper part of the cross-section (area with red (- -) line). On first glance a possible option could be to install the support structure for the rails directly on top of the structural tunnel wall. However, this would require beams intersecting the smoke and helium extraction duct **A** as well as the ceiling **B**, which would impact the volumetric through put in the duct and require costly sealing of the ceiling. Therefore it makes sense to reuse the already existing support structure for the ceiling itself and reinforce some of the horizontal beams in order to carry the weight of the robotic system. Together with civil engineering a structural analysis of different combinations of beams was carried out and concluded that this variant is feasible. This would subsequently mean that the base of the robot will be positioned above the machine **D** or **I**, which lies in the allowed area to move along the tunnel axis (area with green (-.-) line) and comes with additional advantages: The access from this position is ideal for operations on both sides of the machine **D** or **I**. It is relatively simple to install automated hatches on a height of 3 m that allow the robot to pass through but is not easily accessible for people. This is important for the patrol procedures that are carried out before the start of the machine to ensure the safety of workers.

Summed up, it is recommended to mount the rails for the guidance along the tunnel axis under the ceiling **B**, reusing its structural beams.

Parking Spots

In Section 2.3.2 the hazards of radiation during the operation of particle accelerators like FCC were pointed out. Radiation in general, but especially possible penetration of advanced semiconductor hardware with particles of non-zero rest mass is difficult or at least very costly to mitigate. This means that the robotic system can not be parked in the FCC machine tunnel during operation. However, along the machine tunnel exist in total eight access or service caverns, that are well shielded from radiation and can be accessed from the machine tunnel via connection tunnels that diverge with a maximum angle of 15 ° from the main ring. The access and service caverns, see Figures 2.8 and 2.9, are bigger than the machine tunnel to facilitate convenient manipulation of the up to 12 m long magnets. These caverns allow the installation of parking spots for the robotic system. In addition the caverns are connected via access tunnels to the surface and would allow workers, even during operation of FCC, to perform maintenance work on the robotic system itself, while being shielded from radiation.

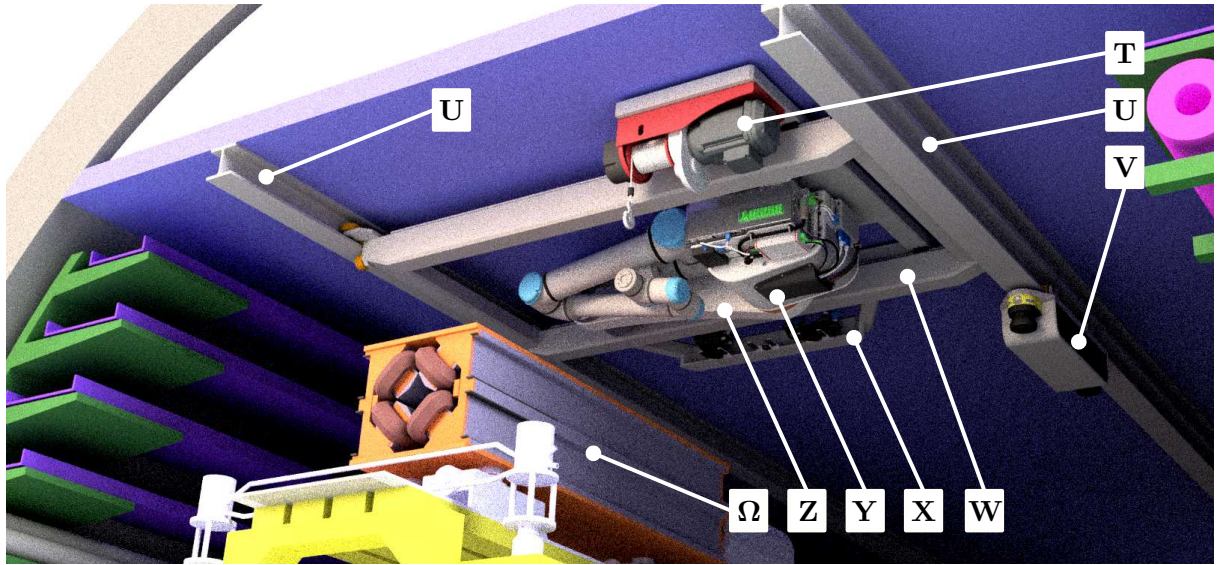


Figure 2.11: The visualization of the recommendation for a robotic system for FCC in the FCC-ee environment.

Summed up, it is recommended to use the eight access and service caverns as radiation safe parking, repair and deployment spots for the robotic system.

2.6.7 Illustration of Concept Robot

The illustration in Figure 2.11 should be understood as an abstract visualization of the high level concepts and a visual summary of the provided recommendations, rather than a technical drawing. The robotic system is shown in the FCC-ee environment, with the main components discussed in this section. **T** indicates the winch for heavy lifting e.g. material transport. **U** points at the right rail which consists of an additional rail in order to allow the emergency system **V** to by pass the main robot. **W** is the chassis of the robotic system which moves along the tunnel axis and in addition it provides a movement in radial direction of the collider ring. **X** indicates the tool changing system. **Y** indicates the robot base of the manipulator, that contains all motor drivers and motors and **Z** marks the foldable mechanical structure of the manipulator. Finally, **Ω** represents the installation in the main tunnel for the FCC-ee layout.

2.6.8 Summary

The following Table 2.4 provides a complete and comprehensive list of all recommendation given for the robotic system for FCC. It shows the recommendations for the full robotic system including the main system, emergency system and recommendations for integration. A system satisfying the above mentioned constraints will have a significant positive impact on the availability of the FCC machine. Furthermore it will significantly improve and contribute to the safety concept for FCC. Especially the improvements concerning availability were clearly emphasized and presented in a talk during the FCC Week 2023 in London [10].

Table 2.4: Summary of the recommendations for a robotic system for FCC. Bold letters refer to corresponding labels in Figure 2.11.

Robotic System - Recommendations	Main System	Locomotion	Rail Guided; Two rails for stability; (U)
		Manipulator	Highly redundant manipulator facilitating high dexterity and low footprint (Y and Z); Workspace: $3.35 \times 5.5 \times 10^5$ m; Payload > 15 kg;
		Power Supply	Continuous supply through rails; (U) Emergency supply through battery system; (Y)
		Additional Features	Winch for high payloads; (T) Tool changing system for various applications; (X)
	Emergency System		Small and fast inspection system; Capable of bypassing the main system; Minimum speed of 35 km/h; (V)
	Integration		Light weight design; Rails mounted on ceiling; Integration of 8 robotic systems; Radiation save parking spots in the 8 access and service caverns; Passing through fire and section doors via automated hatches; High level communication via rails;

2.7 Integration in the Official Base Line Design of the FCC Tunnel

The primary objective of this survey was to assess the feasibility and potential benefits of implementing a robotic system within the FCC. It was important to asses how such a system could impact FCC operations and what its conceptual design might entail. The process of identifying requirements and crafting recommendations was summarized in a presentation to the FCC study board.

Both, the recommendations and the solution-finding process were met with enthusiasm by the board. As a result, the decision was made to reserve a valuable portion of the cross-sectional area exclusively for the robotic system, see Figure 2.12. The reserved area indicates the maximum allowed space for permanently installed material (concerning the robotic system) and represents the maximum footprint of the robotic system while moving along the tunnel axis and e.g. passing the automated hatches in the section and fire doors. During interventions the robot manipulator will obviously also operate outside of this area.

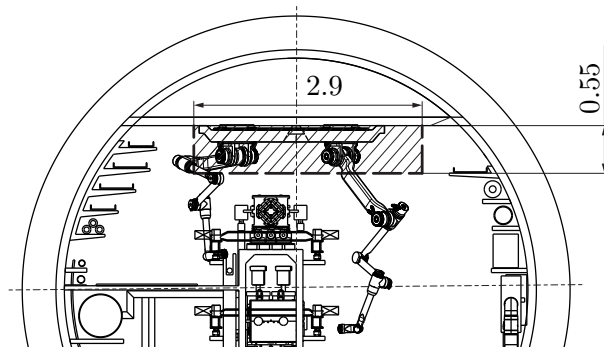


Figure 2.12: Reserved area for the robotic system in the cross-section (V-V from Figure 2.6) of the FCC tunnel. Here the FCC-ee configuration is shown. Dimensions in meter.

Subsequently, the robotic system was integrated into the FCC tunnel together with the integration team and finally it was incorporated in official base line layout of the tunnel cross-section, see Figure 2.13.

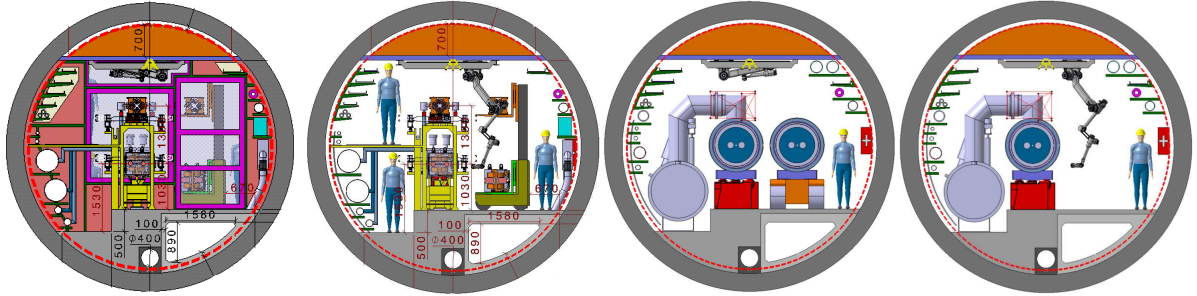


Figure 2.13: The official base line design used by the FCC study for communication of the current status. The two cross-sections on the left are showing the FCC-ee configuration and on the right the FCC-hh configuration.

Furthermore, recognizing the importance of continued development, the board has allocated a budget for robotics projects over the next three years. This funding will facilitate the creation of a prototype to serve as a proof of concept for further studies. In order to approve the allocation of budget a five year work package (see Appendix A.3) for the robotics development had to be presented, including possible collaborations with external institutions. In parallel, a mockup of the FCC tunnel will be developed within this same three-year timeframe, providing an environment for testing and refining the prototype.

These strategic decisions not only reflect the board's commitment to embracing innovative solutions but also underscore the significance of preparing for the future of the FCC with advanced robotics technology.

2.8 Conclusion

The comprehensive analysis concluded with an exploration of the approach to task-specific robotic systems, emphasizing their potential within various applications. The study on CERN's current robotics landscape provided valuable insights into the benefits of integrating robotics at the early design stage, pointing out how such foresight can yield efficient and robust solutions. The high number of adapted robotic solutions within the MRO section, underscores the advantages of employing purpose-built systems.

Subsequently, a focus was put on the specifics of the FCC environment, presenting the unique requirements on the robotic system. These requirements, alongside general prerequisites, were collected into a set of General Requirements. Drawing from experience during remote interventions within the LHC, these requirements were extrapolated to the FCC, ensuring that the recommendations align with anticipated challenges.

The presented recommendations, containing various aspects of a robotic system, ranging from locomotion and manipulation to the end effector, power supply, emergency systems, and integration, were carefully layed out and presented to the FCC Study board. The recommendations

were well received and the integration of the robotic system into the FCC tunnel baseline layout underscores the practicality and feasibility of the proposed solutions.

Looking ahead, a focus will be put on crafting a "Remote Maintenance - Code of Practice" document. This guideline will provide detailed procedures and installation interfaces, ensuring the execution of remote robotic interventions and enhancing efficiency.

Furthermore, it is essential to recognize the importance of an holistic approach to automation. Beyond robotic systems, initiatives such as IoT, Systems Engineering, and Digital Twins should be considered. This broader approach ranges from augment availability to reduce downtime during emergencies, and deliver substantial improvements across the board.

Summed up, the study strongly suggests for the integration of automated systems within the FCC tunnel. This significantly increases availability, fortify safety protocols, and reduce the financial effort associated with manual task execution. Moreover, it is crucial to envision the potential of remote interventions two decades into the future and proactively prepare for the possibilities that lie ahead.

Modelling & Control of Multibody Systems

ABSTRACT. This chapter provides a comprehensive overview of the theoretical concepts, which are necessary to control an electro-mechanical system. The overarching goal of this chapter is to provide an overview of Kinematics, Kinetics, Control and Trajectories, following a clear line of argument. Subsequent chapters will refer to equations and concepts introduced here, directing their focus towards the relevant and innovative advancements.

Chapter Content

3.1	Position & Orientation .	37
3.2	Translation & Rotation	39
3.3	Direct Kinematics . . .	40
3.4	Inverse Kinematics . . .	43
3.5	Kinetics	51
3.6	Control	52
3.7	Mobile Platform with Manipulator	56

Achieving effective control and accurate simulation of multibody systems hinges on the initial development of a precise mathematical representation. In the realm of robotics, these systems are often partially actuated, leading to a resultant net force that propels the system into acceleration. This classification positions them within the dynamic systems category, encompassing both Kinematics (Section 3.3) — an elucidation of geometric movement — and Kinetics (Section 3.5) — an analysis of forces influencing robot motion.

The Kinematics of a robotic system on the special Euclidean group $SE(3)$ can be either described by transformation matrices or by the screw theory. The approach with transformation matrices includes different concept like the Denavit-Hartenberg (DH) parameters using homogenous transformation matrices or a separate representation of translation and orientation with vectors and rotation matrices. Another approach employs quaternions in the form of dual vectors or dual quaternions.

It's important to recognize that all approaches have their respective strengths and weaknesses. The choice between these representations depends on factors such as the complexity of the robotic system, the type of analysis or control being performed, the familiarity of the user with the representation, and the specific requirements of the task.

This thesis utilizes transformation matrices to model the robot kinematics, because of existing tool chains that had already been developed at JKU's Institute of Robotics and CERN's MRO section integrates various open-source libraries and commercial products in the CRF, which also build on the transformation matrices approach.

The DH parameters are very useful to describe the kinematics of a robot with the minimum amount of variables, but pay for this with some limitations on the flexibility of the models, which e.g. do not allow to perform a parameter identification of a robot. Therefor the DH approach will be not further discussed in this thesis.

Given all the aforementioned arguments the kinematic description concept of choice for this thesis are transformation matrices and in the following the basic tools for kinematic robot modellation using specifically the separate representation of translation and orientation with vectors and rotation matrices will be presented.

Once this foundational model is established, it can be categorized based on its inherent characteristics, paving the way for the selection of appropriate control techniques as presented in Section 3.7. Finally, Section 3.7 will unveil the methodologies employed to generate seamless trajectories for the robotic systems, which serve as input to the control mechanism.

3.1 Position & Orientation

The position of a point P on a rigid body, as shown in Figure 3.1, described with respect to an inertial coordinate frame I will be denoted as ${}_I\mathbf{r}_{IP}$. The prefix I defines the coordinate frame in which the vector \mathbf{r} is represented and the postfix IP defines the start and end of the vector. Now if we assume two vectors ${}_I\mathbf{r}_{IP}$ and ${}_B\mathbf{r}_{PQ}$, the first one represented in the inertial frame I pointing from I to P and the second one represented in frame B pointing from P to Q , are given, then the vector from point I to Q represented in the inertial frame can be written as

$${}_I\mathbf{r}_{IQ} = {}_I\mathbf{r}_{IP} + \underbrace{{}_B\mathbf{r}_{PQ}}_{{}_I\mathbf{r}_{PQ}} \in \mathbb{R}^3. \quad (3.1)$$

Before adding two vectors it is important to ensure they are represented in the same coordinate frame. This can be achieved by a transformation of the vector in the desired frame using rotation matrices leading to

$${}_I\mathbf{r}_{PQ} = \mathbf{R}_{IB} {}_B\mathbf{r}_{PQ}. \quad (3.2)$$

The rotation matrix \mathbf{R}_{IB} transforms the vector from a representation in the frame B to a representation in frame I .

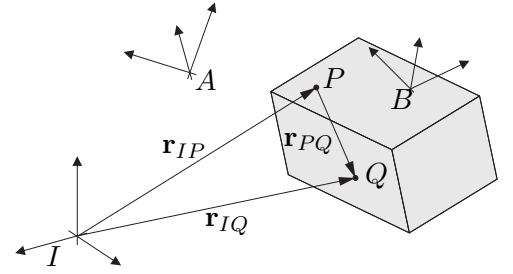


Figure 3.1: Rigid body in space.

A rotation matrix \mathbf{R}_{IB} can be conveniently derived by writing the three unit vectors of the base of frame B represented in frame I in the form

$$\mathbf{R}_{IB} = \begin{bmatrix} {}_I\mathbf{e}_{B,1} & {}_I\mathbf{e}_{B,2} & {}_I\mathbf{e}_{B,3} \end{bmatrix} \in \mathbb{R}^{3 \times 3}. \quad (3.3)$$

Figure 3.2 visualizes the process of decomposing the unit vectors of base B and representing them in with respect to the frame I , which is rotated around ${}_I\mathbf{e}_{I,1}$ by an angle ϕ . The resulting rotation matrix can be written in the form

$$\mathbf{R}_{IB} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.4)$$

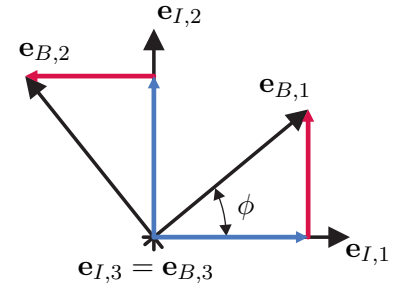


Figure 3.2: Transformation of the unit vectors of frame B to frame I .

Rotation matrices are orthogonal matrices and the multiplication of two rotation matrices results again in a rotation matrix, hence $\mathbf{R}_{IB}^T = \mathbf{R}_{IB}^{-1}$ and $\det(\mathbf{R}) = 1$ holds.

3.1.1 Orientation Representations

Since subsequent rotations around different axis of a coordinate frame are non-commutative, it is necessary to specify conventions for rotations to lead to unique results. Rotations around the axes \mathbf{e}_1 , \mathbf{e}_2 and \mathbf{e}_3 of a right handed coordinate system by the corresponding angles α , β and γ will be represented by rotation matrices \mathbf{R}_α , \mathbf{R}_β and \mathbf{R}_γ .

The Kardan and Euler angle convention entails a sequential series of rotations. Initially, a Kardan rotation transpires around the axis denoted as \mathbf{e}_1 , succeeded by successive rotations around \mathbf{e}_2 and \mathbf{e}_3 and an Euler rotation follows the sequence \mathbf{e}_3 , \mathbf{e}_1 and \mathbf{e}_3 . This sequential process results in the final rotation matrices

$$\mathbf{R}_{Kardan} = \mathbf{R}_\gamma \mathbf{R}_\beta \mathbf{R}_\alpha \quad (3.5)$$

and

$$\mathbf{R}_{Euler} = \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{R}_\psi. \quad (3.6)$$

Kardan and Euler angles continue to maintain their significance in the realm of robotics due to their inherent intuitiveness and visual clarity. However, it's important to acknowledge that these conventions do possess significant limitations, including representation singularities and the tendency to yield unfavorable motion outcomes when employed in path planning.

Another convention that still suffers from representation singularities, but does lead to better results when employed in path planning is the axis-angle representations. This convention describes the orientation using a rotation vector

$$\mathbf{u} = \phi \mathbf{n} \in \mathbb{R}^3 \quad (3.7)$$

with the unit vector $\|\mathbf{n}\| = 1$ and ϕ representing the the axis of orientation and the angle, respectively. The resulting rotation matrix can be written as

$$\mathbf{R}_{AngleAxis} = \left[\mathbf{I} + (1 - \cos\phi)\tilde{\mathbf{n}}\tilde{\mathbf{n}} + \tilde{\mathbf{n}}\sin\phi \right] \quad (3.8)$$

with the cross-product matrix $\tilde{\mathbf{u}}$. The operator $\tilde{(\cdot)}$ will be used to redefine the cross-product of two vectors $\mathbf{a} = [x, y, z]^T$ and \mathbf{b} as a matrix multiplication in the form

$$\mathbf{a} \times \mathbf{b} = \tilde{\mathbf{a}}\mathbf{b} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \mathbf{b}. \quad (3.9)$$

A widely employed approach involves quaternions, which offer the advantage of evading representation singularities. A quaternion can be defined by four Euler parameters and is structured with a scalar component η and a vector $\boldsymbol{\epsilon} = [\epsilon_2, \epsilon_3, \epsilon_4]^T$, fulfilling the condition:

$$\eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 = 1 \quad (3.10)$$

This composition leads to the following definition:

$$\mathcal{Q} = \eta + i\epsilon_2 + j\epsilon_3 + k\epsilon_4 = \{\eta_1, \boldsymbol{\epsilon}\} \quad (3.11)$$

By employing the Euler-Rodrigues equation, the rotation matrix can be deduced in the format:

$$\mathbf{R}_{Quaternion} = \left[\mathbf{I} + 2(\eta\tilde{\boldsymbol{\epsilon}} + \tilde{\boldsymbol{\epsilon}}\boldsymbol{\epsilon}) \right] \quad (3.12)$$

This quaternion-based representation avoids representation singularities. Nevertheless, its effective use in trajectory planning requires thoughtful consideration due to its unique characteristics.

It is worth mentioning that for a given orientation of a rigid body, all the above introduced

orientation representations (if not in a singularity) lead to a unique rotation matrix:

$$\mathbf{R}_{Kardan} = \mathbf{R}_{Euler} = \mathbf{R}_{AngleAxis}^T = \mathbf{R}_{Quaternion}^T \quad (3.13)$$

3.1.2 Error Measure

The error measure $\boldsymbol{\theta}$ can be split up in a position and orientation error $\boldsymbol{\theta} = [\boldsymbol{\theta}_P^T \ \boldsymbol{\theta}_O^T]^T$. The positional error can be simply calculated by element-wise subtracting the first three entries of a desired and actual position vector e.g. ${}^I\mathbf{r}_{des} - {}^I\mathbf{r}_{act}$. For the orientation error a measure needs to be defined that avoids representation singularities and will lead to asymptotically stable error dynamics. This can be achieved with a measure based on the axis-angle representatino or quaternions, as presented in the following sections.

Axis-Angle

The orientation error $\boldsymbol{\theta}_O = \mathbf{u}_e = \phi_e \mathbf{n}_e$ with the error angle ϕ_e and corresponding axis \mathbf{n}_e . For small errors the orientation error can be set to $\boldsymbol{\theta}_O = \sin \phi_e \mathbf{n}_e$, which will simplify the following steps. This furthermore allows to define the error measure using rotation matrices in the form $\Delta \mathbf{R} = \mathbf{R}_{ref}(t) \mathbf{R}_{act}^T(\mathbf{q}(t))$ with $\mathbf{R}_{ref} = \begin{bmatrix} \mathbf{a} & \mathbf{b} & \mathbf{c} \end{bmatrix}$ and $\mathbf{R}_{act} = \begin{bmatrix} \mathbf{d} & \mathbf{e} & \mathbf{f} \end{bmatrix}$ leading to the final error measure

$$\boldsymbol{\theta}_O = -\frac{1}{2} \left(\tilde{\mathbf{a}}\mathbf{d} + \tilde{\mathbf{b}}\mathbf{e} + \tilde{\mathbf{c}}\mathbf{f} \right). \quad (3.14)$$

Quaternion

To represent the orientation error $\boldsymbol{\theta}_O$, first the two quaternions $\mathcal{Q}_{ref} = \{\eta_{ref}, \boldsymbol{\epsilon}_{ref}\}$ and $\mathcal{Q}_{act} = \{\eta_{act}, \boldsymbol{\epsilon}_{act}\}$, corresponding to \mathbf{R}_{ref} and \mathbf{R}_{act} have to be defined. Then, an error quaternion

$$\Delta \mathcal{Q} = \{\Delta \eta, \Delta \boldsymbol{\epsilon}\} = \mathcal{Q}_{ref} * \mathcal{Q}_{act}^{-1} \quad (3.15)$$

can be calculated and used to define the orientation error in the form

$$\boldsymbol{\theta}_O = \Delta \boldsymbol{\epsilon} = \eta_{act}(\mathbf{q}) \boldsymbol{\epsilon}_{ref} - \eta_{ref} \boldsymbol{\epsilon}_{act}(\mathbf{q}) - \tilde{\boldsymbol{\epsilon}}_{ref} \boldsymbol{\epsilon}_{act}(\mathbf{q}), \quad (3.16)$$

which is the vector part of the quaternion $\Delta \mathcal{Q}$.

3.2 Translation & Rotation

To get the absolute velocity of a point P the vector of position has to be differentiated in the inertial coordinate system,

$${}^I\mathbf{v}_P = {}^I\dot{\mathbf{r}}_{IP}. \quad (3.17)$$

However, a more practical representation of ${}_I\mathbf{v}_P$ can be found by differentiating the vector of position while represented in any other coordinate system than the inertial frame. This leads to

$$\begin{aligned} {}_B\mathbf{v}_P &= \mathbf{R}_{BI} \dot{{}_I\mathbf{r}_{IP}} = \\ &= \mathbf{R}_{BI} \frac{d}{dt} (\mathbf{R}_{IB} {}_B\dot{\mathbf{r}}_{IP}) = \\ &= \underbrace{\mathbf{R}_{BI}\mathbf{R}_{IB}}_{\mathbf{E}} {}_B\dot{\mathbf{r}}_{IP} + \underbrace{\mathbf{R}_{BI}\dot{\mathbf{R}}_{IB}}_{{}_B\tilde{\boldsymbol{\omega}}_{IB}} {}_B\mathbf{r}_{IP} = {}_B\dot{\mathbf{r}}_{IP} + {}_B\tilde{\boldsymbol{\omega}}_{IB} {}_B\mathbf{r}_{IP}. \end{aligned} \quad (3.18)$$

As already shown above, the angular velocity can be calculated to

$${}_B\boldsymbol{\omega}_{IB} = \mathbf{R}_{BI}\dot{\mathbf{R}}_{IB}, \quad (3.19)$$

which is correct, but the computation complexity can be reduced by taking into account that a specific order of subsequent rotations that has been defined and results in the following expressions for Kardan-angles, Euler-angles, the axis angle representation and for quaternions:

$${}_B\boldsymbol{\omega}_{IB} = \mathbf{R}_\gamma \mathbf{R}_\beta \mathbf{e}_1 \dot{\alpha} + \mathbf{R}_\gamma \mathbf{e}_2 \dot{\beta} + \mathbf{e}_3 \dot{\gamma} \quad \text{for Kardan - angles } \alpha, \beta, \gamma \quad (3.20)$$

$${}_B\boldsymbol{\omega}_{IB} = \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{e}_1 \dot{\psi} + \mathbf{R}_\phi \mathbf{e}_2 \dot{\theta} + \mathbf{e}_3 \dot{\phi} \quad \text{for Euler - angles } \phi, \theta, \psi \quad (3.21)$$

$${}_B\boldsymbol{\omega}_{IB} = \dot{\phi} \mathbf{n} + [\sin \phi \mathbf{I} - (1 - \cos \phi) \tilde{\mathbf{n}}] \dot{\mathbf{n}} \quad \text{for Axis - Angle parameters } \mathbf{n}, \phi \quad (3.22)$$

$${}_B\boldsymbol{\omega}_{IB} = 2[\eta \dot{\epsilon} - \tilde{\epsilon} \dot{\epsilon} - \epsilon \dot{\eta}] \quad \text{for quaternion } \{\eta, \epsilon\} \quad (3.23)$$

3.3 Direct Kinematics

Direct kinematics refers to the description of the end effector position and orientation in the Euclidean space as a function of the minimal coordinates of a mechanical system. Often the joint angles also form the minimal coordinates of a robotic system.

Definition 1 (Minimal Coordinates) *The minimal coordinates represent a minimal set of variables*

$$\mathbf{q} \in \mathcal{C} \subset \mathbb{R}^r, \quad (3.24)$$

such that a mechanism's configuration can be uniquely described. Vector $\mathbf{q} = (q_1, \dots, q_r)^T$ contains the r minimal coordinates, r denoting the degree of freedom (DoF) of a system and \mathcal{C} represents the configuration space of all possible combinations of the minimal coordinates.

The direct kinematics or the mapping from joint to task space results in

$$\mathbf{z} = \mathbf{f}(\mathbf{q}) \in \mathcal{T} \subset \mathbb{R}^t \quad (3.25)$$

with the task space \mathcal{T} and t denoting the dimension of the task space. Commonly t will correspond to the DoF of a rigid body in the 3 dimensional Euclidean space ($t = 6$), but in general could be less for reduced task spaces or greater for multi end effector systems.

3.3.1 The Kinematic Chain

Figure 3.3 shows a 2 DoF serial mechanism which will be used in the following to visually support the introduction of the concept of the kinematic chain. The coordinate frames are numbered from $0 = I \dots 2 = r = E$, the joint angles q_1 and q_2 are relative to the previous link, the end effector position is referred to as \hat{E} , the vector $\mathbf{r}_{I\hat{E}}$ points from the inertial frame to the end effector position and the rotation matrix \mathbf{R}_{IE} describes the orientation between the inertial frame I and the end effector frame E .

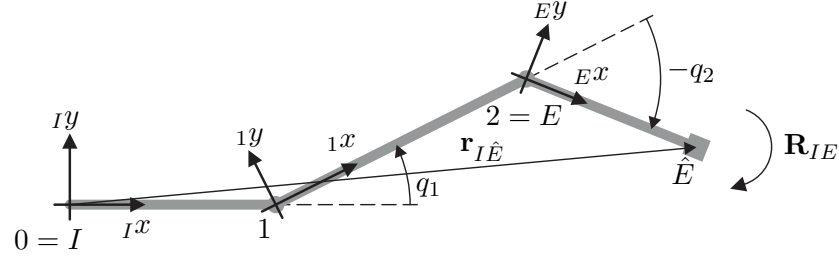


Figure 3.3: A 2 DoF serial mechanism.

The translational part $\mathbf{r}_{I\hat{E}}$ of the direct kinematics \mathbf{f} can be derived from (3.1) and the rotational part \mathbf{R}_{IE} can be found by solving the inverse problem to (3.5), (3.6), (3.8) or (3.12), depending on the chosen orientation representation. A compact solution for the direct kinematics of a serial mechanism can be found by recursively summing up the position vectors and multiplying the rotation matrices between the robot joints, starting from the end effector frame E until the inertial frame I , in the form

$$\mathbf{R}_{n-1,E} = \mathbf{R}_{n-1,n} \mathbf{R}_{n,E} \quad \text{for } n \in \{r \dots 1\} \quad (3.26)$$

and

$${}^{n-1}\mathbf{r}_{n-1,\hat{E}} = {}^{n-1}\mathbf{r}_{n-1,n} + \mathbf{R}_{n-1,n} {}^n\mathbf{r}_{n,\hat{E}} \quad \text{for } n \in \{r \dots 1\}. \quad (3.27)$$

The recursive terms in (3.27) and (3.26) are highlighted in blue and the start for the recursive term in (3.27) is $\mathbf{R}_{r,E} = \mathbf{R}_{EE} = \mathbf{I}$ and ${}^r\mathbf{r}_{r,\hat{E}} = {}^E\mathbf{r}_{E,\hat{E}}$ for (3.26).

The velocities of the end effector can be described in a similar form, using the definitions for the translational and rotational velocities, (3.18) and (3.19), and again formulating the recursive kinematic chains

$${}^{n-1}\boldsymbol{\omega}_{n-1,E} = \underbrace{{}^{n-1}\mathbf{u}_{n-1} \dot{q}_{n-1}}_{t_1} + \underbrace{\mathbf{R}_{n-1,n} {}^n\boldsymbol{\omega}_{n,E}}_{t_2} \quad \text{for } n \in \{r \dots 1\} \quad (3.28)$$

with the rotation axis ${}_a\mathbf{u}_b$ of joint b represented in frame a and

$${}^{n-1}\mathbf{v}_{n-1,\hat{E}} = \underbrace{{}^{n-1}\dot{\mathbf{r}}_{n-1,n}}_{t_3} + \underbrace{{}^{n-1}\tilde{\boldsymbol{\omega}}_{n-2,n-1} {}^{n-1}\mathbf{r}_{n-1,n}}_{t_4} + \underbrace{\mathbf{R}_{n-1,n} {}^n\mathbf{v}_{n,\hat{E}}}_{t_5} \quad \text{for } n \in \{r \dots 1\}. \quad (3.29)$$

The recursive terms in (3.28) and (3.29) are again highlighted in blue and the start for the recursive term in (3.28) is ${}^r\boldsymbol{\omega}_{r,E} = {}^E\boldsymbol{\omega}_{E,E} = 0$ and for (3.29) it is ${}^r\mathbf{v}_{r,\hat{E}} = {}^E\mathbf{v}_{E,\hat{E}} = 0$ considering rigid bodies. $\dot{q}_0 = 0$ by definition. In case the corresponding joint to \dot{q}_i resembles a linear axis, then the \dot{q}_i in t_1 in (3.28) will be set to 0.

3.3.2 Geometric Jacobian

The Jacobian matrix and its inverse are powerful tools that provide a linear mapping between the task and configuration space $\mathcal{T} \longleftrightarrow \mathcal{C}$ of a mechanism. In general the Jacobian matrix can be derived in two ways, yielding either the geometric or analytic Jacobian. To find the analytic Jacobian, one has to simply compute the derivative of the direct kinematics (3.25) by the minimal coordinates \mathbf{q} . However, this will lead to a Jacobian matrix that contains the orientation representation singularities, which is clearly not desirable. Therefore, the more common geometric Jacobian $\mathbf{J}(\mathbf{q})$, that does not possess such singularities, will be introduced here and furthermore used throughout this thesis in the form

$$\dot{\mathbf{z}} = \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{pmatrix} = \left(\frac{\partial \dot{\mathbf{z}}}{\partial \dot{\mathbf{q}}} \right) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_r \end{pmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (3.30)$$

with the end effector velocity $\dot{\mathbf{z}}$ and the joint velocities $\dot{\mathbf{q}}$. The geometric Jacobian can be derived by performing the derivation in (3.30) or purely by geometric considerations, which become clear when taking a closer look at the recursive definitions of the velocities in (3.28) and (3.29) and thus no additional derivation will be required.

During the first iteration of (3.28) the term $t_2 = 0$ and it can be observed that for every rotational joint the term t_1 will be added up and propagated via the rotation transformations until the inertial frame. Furthermore, the joint velocities \dot{q}_i are already isolated, which corresponds to the formulation (3.30) and thus makes it easy to extract the geometric Jacobian from the computation of the velocities. This becomes even more clear when rearranging (3.28) in vector notation

$${}^I\boldsymbol{\omega}_{IE} = \begin{bmatrix} {}^I\mathbf{u}_1 & {}^I\mathbf{u}_2 & \dots & {}^I\mathbf{u}_E \end{bmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_r \end{pmatrix}. \quad (3.31)$$

The recursive computation of the translational velocities allows for similar insights. During the first iteration the term $t_5 = 0$ and for rotational axes and rigid bodies also the term $t_3 = 0$. Substituting the solution for $\boldsymbol{\omega}$ from (3.28) in t_4 shows that again t_1 will be propagated from the end effector until the inertial frame, however this time not only via the rotation transformations but also the cross product $\boldsymbol{\omega} \times \mathbf{r}$. For linear axes the term $t_4 = 0$ and term $t_3 = {}_{n-1}\mathbf{u}_{n-1} \dot{q}_{n-1}$. The translational velocities for rotational joints can therefore be written in vector form

$${}^I\mathbf{v}_{IE} = \begin{bmatrix} {}^I\tilde{\mathbf{u}}_1 & {}^I\mathbf{r}_{1\hat{E}} & {}^I\tilde{\mathbf{u}}_2 & {}^I\mathbf{r}_{2\hat{E}} & \dots & {}^I\tilde{\mathbf{u}}_E & {}^I\mathbf{r}_{E\hat{E}} \end{bmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_r \end{pmatrix}. \quad (3.32)$$

Substituting (3.31) and (3.32) in (3.30) allows to identify the geometric Jacobian $\mathbf{J}(\mathbf{q})$ without

additional computational effort, once the velocities have been computed, from the formulation

$${}_I\dot{\mathbf{z}}_{I\hat{E}} = \begin{pmatrix} {}_I\mathbf{v}_{I\hat{E}} \\ {}_I\boldsymbol{\omega}_{IE} \end{pmatrix} = \underbrace{\begin{bmatrix} {}_I\tilde{\mathbf{u}}_1 & {}_I\mathbf{r}_{1\hat{E}} & {}_I\tilde{\mathbf{u}}_2 & {}_I\mathbf{r}_{2\hat{E}} & \cdots & {}_I\tilde{\mathbf{u}}_E & {}_I\mathbf{r}_{E\hat{E}} \\ {}_I\mathbf{u}_1 & & {}_I\mathbf{u}_2 & & \cdots & & {}_I\mathbf{u}_E \end{bmatrix}}_{\mathbf{J}(\mathbf{q})} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_r \end{pmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (3.33)$$

with $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{t \times r}$. As mentioned above, for every linear axis the corresponding column of $\mathbf{J}(\mathbf{q})$ has to be replaced with $\begin{bmatrix} {}_I\mathbf{u}_1^T & \mathbf{0}^T \end{bmatrix}^T$.

3.4 Inverse Kinematics

Inverse kinematics is of fundamental importance whenever a robot end effector is required to follow a desired trajectory in task space, by controlling the robots minimal coordinates. In other words, the inverse kinematics is employed to map from task to configuration space $\mathcal{T} \rightarrow \mathcal{C}$.

For specific mechanisms it is possible to find such a mapping by exploiting algebraic and geometric strategies, leading to an explicit, analytic expression that provides an instantaneous solution for the inverse kinematics problem. However, in the most general case this approach isn't viable for mechanical systems and will thus, not be further discussed in this thesis.

The differential kinematics provide a more general technique for the inverse kinematics problem and can be formulate by rearranging the equation for the direct kinematics on velocity level (3.30), leading to the differential inverse kinematics

$$\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})^{-1}\dot{\mathbf{z}}. \quad (3.34)$$

In order to compute this mapping the Jacobian needs to be non-singular and square, such that the inverse exists. As mentioned in Section 3.3.2 the geometric Jacobian is employed in order to avoid representation singularities. However, a robot always possess inherent singularities depending on the topology and geometry of the mechanical structure, which has to be accounted for when performing the inverse kinematics mapping. Techniques for singularity avoidance are presented in Section 3.4.6.

In case the dimension of a manipulators configuration space is greater than its task space dimension, $r > t$, the mechanism is denoted as redundant and the Jacobian $\mathbf{J} \in \mathbb{R}^{t \times r}$ is non square. Redundant structures can be desirable in order to increase flexibility and dexterity, but require more sophisticated techniques for redundancy resolution. Figure 3.4 provides a non-extensive list of the most common redundancy resolution techniques, with references to dedicated literature.

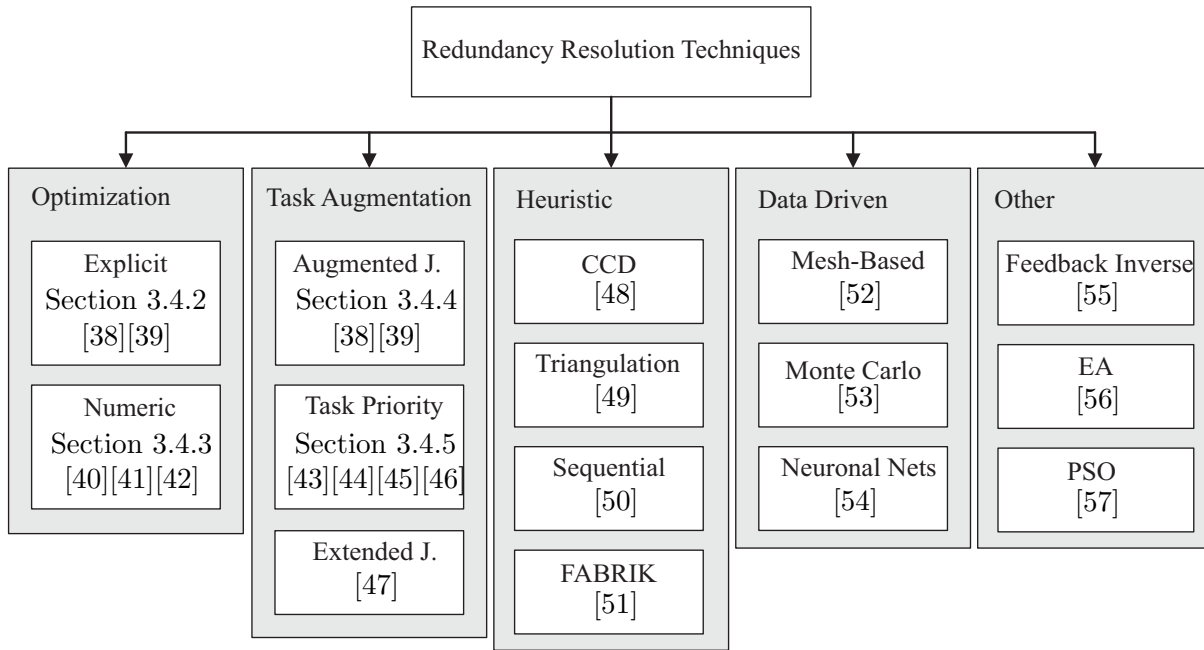


Figure 3.4: Methods for Inverse Kinematic Solutions

The redundancy resolution via optimization and task augmentation allows to incorporate additional tasks, next to following a desired end effector trajectory, such as collision avoidance, joint limits, locking certain joints or other deterministic measures. Therefor these techniques are very powerful when robots are operating in confined space or operators need more control of the robots configuration during teleoperation, than just tracking the desired end effector position and orientation.

The heuristic methods usually posses lower computational complexity, but the resulting configuration of the mechanism heavily depends on the implemented strategy and can not be adjusted by additional constraints or tasks. Therefor these techniques are commonly employed for time critical applications that do not require to find a specific solution in the configuration space, as for example for game engines. Data driven techniques aim to probe or sample the mapping from task to configuration space and store or learn the derived data. The last pillar collects other interesting techniques like Evolutionary Algorithms (EA), Particle Swarm Optimization (PSO) or the Feedback Inverse, which implements the mapping with the Jacobian \mathbf{J} in the feedback loop and makes use of the implicit inversion, when computing the closed-loop transfer function.

In the following Section 3.4.1 useful properties of the Jacobian, which will be exploited in the several of the redundancy resolution techniques, will be discussed. Then, the explicit and numeric solutions via optimization will be discussed in Sections 3.4.2 and 3.4.3 and the augmented Jacobian and task priority approach will be presented in Sections 3.4.4 and 3.4.5. Finally, some notes on singularity avoidance and numeric stabilization of inverse kinematics algorithms will be given in Sections 3.4.6 and 3.4.7.

3.4.1 Nullspace & Range of the Jacobian

In the case of redundant mechanical structures infinitely many combinations of \mathbf{q} for one desired task position and orientation \mathbf{z} exist. Some useful properties of the Jacobian for redundant systems can be derived by analyzing the kernel $\ker(\mathbf{J})$ and the image $\text{im}(\mathbf{J})$ as shown in Figure 3.5. The kernel of \mathbf{J} corresponds to the null space

$$\mathcal{N}(\mathbf{J}) = \ker(\mathbf{J}) \quad (3.35)$$

and denotes arbitrary joint velocities $\dot{\mathbf{q}}$ that do not generate any velocities $\dot{\mathbf{z}}$ in task space. The image of \mathbf{J} corresponds to the range of \mathbf{J}

$$\mathcal{R}(\mathbf{J}) = \text{im}(\mathbf{J}) \quad (3.36)$$

and denotes the velocity $\dot{\mathbf{z}}$ that can be generated by $\dot{\mathbf{q}}$.

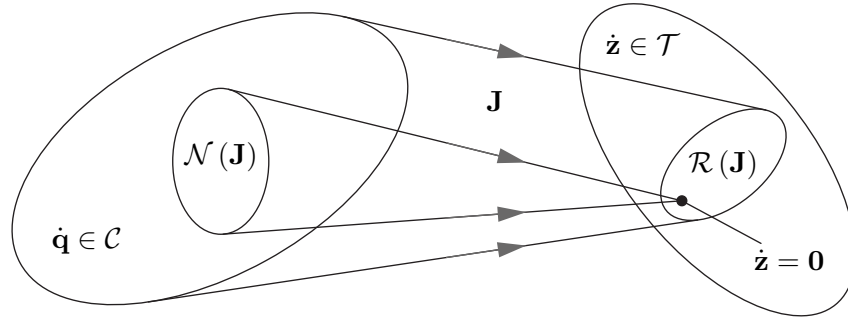


Figure 3.5: Mapping of the Jacobian matrix

If a mapping \mathbf{N} can be found such that \mathbf{N} projects into the nullspace of \mathbf{J} , then for a given solution $\dot{\mathbf{q}}^*$ of (3.34) the joint coordinates can be written as

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}^* + \mathbf{N}\dot{\mathbf{q}}_0. \quad (3.37)$$

Premultiplying (3.37) with \mathbf{J} yields

$$\mathbf{J}\dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{q}}^* + \mathbf{J}\mathbf{N}\dot{\mathbf{q}}_0. \quad (3.38)$$

Since $\mathcal{R}(\mathbf{N}) \equiv \mathcal{N}(\mathbf{J})$, any vector $\dot{\mathbf{q}}_0 \in \mathcal{R}(\mathbf{N})$ or $\dot{\mathbf{q}}_0 \in \mathcal{N}(\mathbf{N})^1$ will be mapped into the nullspace of \mathbf{J} . Furthermore for a vector $\mathbf{x} \in \mathcal{N}(\mathbf{J})$ equation $\mathbf{J}\mathbf{x} = \mathbf{0}$ holds and thus with

$$\mathbf{J} \underbrace{\mathbf{N}\dot{\mathbf{q}}_0}_{\in \mathcal{N}(\mathbf{J})} = \mathbf{0} \quad (3.39)$$

equation (3.38) yields

$$\mathbf{J}\dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{q}}^* + \mathbf{J}\mathbf{N}\dot{\mathbf{q}}_0 = \mathbf{J}\dot{\mathbf{q}}^* = \dot{\mathbf{z}}. \quad (3.40)$$

This means that a redundant robot can perform internal motions which are not effecting the end-effector. This behavior can be exploited in order to e.g. avoid obstacles while still following the desired end effector trajectory.

¹ $\mathbf{0}$ is subspace of every nullspace, since $\mathbf{A}\mathbf{0} = \mathbf{0}$

3.4.2 Minimum Norm Solution

Though a solution can be determined by formulating a constrained optimization problem,

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} \\ \text{s.t.} \quad & \mathbf{J} \dot{\mathbf{q}} - \dot{\mathbf{z}} = \mathbf{0} \end{aligned} \quad (3.41)$$

which leads to a solution such that the joint velocities are minimal and the mobility of each joint is defined by the positive-definite weighing matrix \mathbf{W} . Energy optimal motions can be generated by setting the weighting matrix to the mass matrix of the robot $\mathbf{W} = \mathbf{M}$, since according to the optimization problem (5.5) the objective function is equal to the kinetic energy.

Reformulating (5.5) as a Langrange function

$$\mathcal{L}(\dot{\mathbf{q}}, \boldsymbol{\lambda}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} + \boldsymbol{\lambda}^T (\dot{\mathbf{z}} - \mathbf{J} \dot{\mathbf{q}}) \quad (3.42)$$

with the conditions

$$\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} \right)^T = \mathbf{0} \quad \text{and} \quad \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} \right)^T = \dot{\mathbf{z}} - \mathbf{J} \dot{\mathbf{q}} = \mathbf{0} \quad (3.43)$$

one obtains

$$\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} \right)^T = \mathbf{W} \dot{\mathbf{q}} - \mathbf{J}^T \boldsymbol{\lambda} = \mathbf{0} \quad (3.44)$$

and thus

$$\dot{\mathbf{q}} = \mathbf{W}^{-1} \mathbf{J}^T \boldsymbol{\lambda}. \quad (3.45)$$

Substituting (3.45) in (3.43) yields

$$\dot{\mathbf{z}} = \mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T \boldsymbol{\lambda}. \quad (3.46)$$

Finally, substituting the expression for $\boldsymbol{\lambda}$ from (3.46) in (3.45) leads to

$$\dot{\mathbf{q}} = \underbrace{\mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T)^{-1}}_{\mathbf{J}^\dagger} \dot{\mathbf{z}} = \mathbf{J}^\dagger \dot{\mathbf{z}}, \quad (3.47)$$

with the pseudo inverse \mathbf{J}^\dagger . Thus, (3.47) provides an explicit solution for the inverse kinematics problem for redundant robots, by reaching the desired task trajectory with minimal joint velocities. This solution however, does not allow to take additional task, like joint limits, collision avoidance, etc. into account. If one wants to consider additional tasks when solving the inverse kinematics problem, this can be added to the objective function of (5.5) in the form of an artificial potential energy $h(\mathbf{q})$ with the time derivative

$$\frac{d}{dt} h(\mathbf{q}) = \frac{\partial h(\mathbf{q})}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} = \nabla h(\mathbf{q})^T \dot{\mathbf{q}}. \quad (3.48)$$

This leads to the new formulation of the optimization problem in the form

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} + \nabla h(\mathbf{q})^T \dot{\mathbf{q}} \\ \text{s.t.} \quad & \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} - \dot{\mathbf{z}} = \mathbf{0}. \end{aligned} \quad (3.49)$$

The rate of change of the potential energy will be minimized by moving in direction of the gradient and thus will maximize $h(\mathbf{q})$. To solve this optimization problem, again a Lagrange

function similar to (3.42) can be employed and yields

$$\dot{\mathbf{q}} = \mathbf{W}^{-1} \mathbf{J}^T \left(\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T \right)^{-1} \dot{\mathbf{z}} + \underbrace{(\mathbf{I} - \mathbf{W}^{-1} \mathbf{J}^T \left(\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T \right)^{-1} \mathbf{J})}_{\mathbf{N}} \mathbf{W}^{-1} \nabla h(\mathbf{q}) \quad (3.50)$$

and with $\mathbf{J}^\dagger = \mathbf{W}^{-1} \mathbf{J}^T \left(\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T \right)^{-1}$ simplifies to

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{z}} + \underbrace{(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})}_{\mathbf{N}} \mathbf{W}^{-1} \nabla h(\mathbf{q}). \quad (3.51)$$

\mathbf{I} denotes the identity matrix and \mathbf{N} maps into the nullspace of \mathbf{J} . This explicit solution of the inverse kinematics problem now allows, with the second term of (5.7), to take additional tasks into account. It is worth mentioning that the task of tracking the desired end effector position has priority and the additional or secondary task can therefor not be guaranteed to always fall in the global optimum of $h(\mathbf{q})$.

Table 3.1 provides examples of additional tasks that can be considered in (5.7). The column $h(\mathbf{q})$ contains the formulation of the artificial potential and column $\nabla h(\mathbf{q})$ lists its gradient when an analytic formulation is possible. The constants c_L , \mathbf{c}_P , c_S , c_C allow for tuning of the sensitivity with respect to the other objectives and \mathbf{c}_P , \mathbf{p}_L map the constraints to the corresponding joints. The gradient of e.g. the *Distance from Singularity* can only be calculated numerical because of the high computational costs and the gradient of *Collision Avoidance* can in general only be calculated numerical because the information about the environment is passed in the form of data sets (e.g. point clouds) and not analytic expressions.

Secondary Tasks	$h(\mathbf{q})$	$\nabla h(\mathbf{q})$
Joint Limits Max	$-\mathbf{p}_L e^{c_L(q-q_{max})}$	$-\mathbf{p}_L c_L e^{c_L(q-q_{max})}$
Joint Limits Min	$-\mathbf{p}_L e^{-c_L(q-q_{min})}$	$\mathbf{p}_L c_L e^{-c_L(q-q_{min})}$
Desired Joint Position	$-\frac{1}{2} \mathbf{c}_P (q - q_{des})^2$	$\mathbf{c}_P (q_{des} - q)$
Distance from Singularity	$c_S \sqrt{\det \{ \mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}) \}}$	numerical
Collision Avoidance	$-e^{-c_C(dist(\mathbf{q})-d_{min})}$	numerical
Torque Optimization	$\boldsymbol{\tau}^T \mathbf{T} \boldsymbol{\tau}$	numerical

Table 3.1: Secondary tasks and their corresponding artificial potential.

The inverse kinematics technique described in this section provides the explicit solution (5.7) for the joint velocities $\dot{\mathbf{q}}$, but does not allow to employ inequality constraints in the optimization problem, such as joint limits or collision avoidance. These inequality constraints were instead considered by adding a secondary task in the form of an artificial potential energy to the objective function. Since the higher priority task forces the tracking of the end effector trajectory, this

technique can return results that violate the joint limits if the tracking of the end effector requires it. Furthermore, since the artificial potential energy resembles a convex function, the solution (5.7) aims to always pull a joint angle towards its neutral position, which might not necessarily be the desired behavior when considering joint limits. However, this effect can be mitigated by employing a convex function for the Joint Limits task, as shown in Table 3.1, that possess a high gradient close to the joint limits and a low gradient in between.

3.4.3 Numeric Solution via Optimization

This technique aims to directly consider inequality constraints such as joint limits or collision avoidance in the optimization problem. As a result, this method is unable to provide an explicit solution for the joint velocities $\dot{\mathbf{q}}$ and numeric optimization techniques have to be applied.

The problem can be formulated in the following form

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} \\ \text{s.t.} \quad & \mathbf{J} \dot{\mathbf{q}} - \dot{\mathbf{z}} = \mathbf{0} \\ & \mathbf{c}_{lb} \leq \mathbf{A} \dot{\mathbf{q}} \leq \mathbf{c}_{ub} \\ & \dot{\mathbf{q}}_{lb} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{ub}, \end{aligned} \tag{3.52}$$

and with the joint velocity limits $\dot{\mathbf{q}}_{lb}$, $\dot{\mathbf{q}}_{ub}$ and collision avoidance or joint position limits \mathbf{c}_{lb} , \mathbf{c}_{ub} and resembles a convex, time variant, quadratic optimization problem, for $\mathbf{W} > 0$, that can be solved with quadratic programming solvers. More details on the problem formulation can be found in [41], [40].

3.4.4 Augmented Jacobian

The augmented Jacobian \mathbf{J}_a aims to add additional tasks to the inverse kinematics problem and thus append $t - r$ additional lines to the geometric Jacobian \mathbf{J} such that $\mathbf{J}_a \in \mathbb{R}^{r \times r}$ will be square. The relation of the newly introduced task \mathbf{z}_1 and the minimal coordinates \mathbf{q} can be considered as a direct kinematics equation

$$\mathbf{z}_1 = \mathbf{g}(\mathbf{q}) \tag{3.53}$$

with the continuous nonlinear vector function \mathbf{g} . Considering the time derivative

$$\dot{\mathbf{z}}_1 = \frac{\partial \mathbf{g}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_1 \dot{\mathbf{q}} \tag{3.54}$$

the augmented Jacobian can be written as

$$\mathbf{J}_a = \begin{bmatrix} \mathbf{J} \\ \mathbf{J}_1 \end{bmatrix}. \tag{3.55}$$

The augmented Jacobian \mathbf{J}_a can then be substituted for the geometric Jacobian \mathbf{J} in (3.34) to obtain a solution for the inverse kinematics. Furthermore it is worth noting that the augmented Jacobian can as well embed the extended Jacobian which then allows for cyclic configurations, see [47].

3.4.5 Task Priority

If one wants to consider multiple tasks in the inverse kinematics, then it is useful to assign priorities to those tasks, such that they do not interfere with each other. For example it could be of interest to assign the highest priority to a task that avoids collisions, then the next lower priority to the tracking of the the end effector trajectory and the lowest priority to a task that positions the mechanism in a configuration such that the error propagation throughtout the mechanical structure is minimal.

In general, let us consider n tasks $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$ and take their time derivatives following (3.54), to find the corresponding Jacobian matrices $\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_n$. Then, the successively growing augmented Jacobian

$$\mathbf{J}_{1,i} = \begin{bmatrix} \mathbf{J}_1^T & \mathbf{J}_2^T & \dots & \mathbf{J}_i^T \end{bmatrix} \quad (3.56)$$

and its nullspace projectors

$$\mathbf{N}_i = \mathbf{I} - \mathbf{J}_{1,i}^\dagger \mathbf{J}_{1,i}, \quad (3.57)$$

with the pseudo inverse \mathbf{J}^\dagger from (3.47) can be defined. Then the final inverse kinematics solution results in

$$\dot{\mathbf{q}} = \mathbf{J}_1^\dagger \dot{\mathbf{z}}_{1,ref} + \mathbf{N}_1 \mathbf{J}_2^\dagger \dot{\mathbf{z}}_{2,ref} + \dots + \mathbf{N}_n \mathbf{J}_n^\dagger \dot{\mathbf{z}}_{n,ref}. \quad (3.58)$$

3.4.6 Representation, Kinematic & Algorithmic Singularities

Representation Singularities can be avoided by employing the geometric Jacobian, which in contrast to the analytic Jacobian, does not possess any representation singularities (originating from the orientation representations), since it is solely based on geometric considerations. Every mechanism however, suffers from kinematic singularities depending on the topology and geometry of mechanical structure, which therefor have to be handled in the IK algorithms. If secondary objectives (Section 3.4.4-3.4.5) will be introduced in an IK algorithm for redundancy resolution, possible algorithmic singularities have to be considered.

To mitigate the undesirable effects when a mechanism configuration is close to a kinematic singularity a damped pseudo inverse can be employed, which comes with the cost of a deviation from the desired end effector trajectory. An extensive research on such mitigation strategies can be found in [58]. In this thesis one of the simplest yet efficient approaches was applied which is presented in the following.

In order to mitigate the effect of kinematic singularities, a variable damping factor will be added to the pseudo inverse (3.47) to ensure that the inverse always exist. Thus, the damped pseudo inverse

$$\mathbf{J}^\dagger = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + \alpha \mathbf{I})^{-1}, \quad (3.59)$$

with a value α that increases close to singularities and is set zero otherwise to guarantee precise tracking of the desired trajectory, will be defined. A common choice of α is

$$\alpha = \begin{cases} \alpha_0 \left(1 - \frac{w_{kin}}{w_0}\right)^2 & w_{kin} < w_0 \\ 0 & w_{kin} \geq w_0 \end{cases} \quad (3.60)$$

where $\alpha_0 = \text{const.}$, $w_0 = \text{const.}$ and the kinematic manipulability measure

$$w_{kin}(\mathbf{q}) = \sqrt{\det\{\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q})\}}, \quad (3.61)$$

which measures the distance to the singularity. The damped least-squares pseudo inverse is used to invert non-square matrices and provides a stable inversion of \mathbf{J} close to and even in singularities.

Algorithmic singularities can be simply avoided by ensuring linear independency or compatibility between all task $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$ such that

$$\text{rank}(\mathbf{J}_1^\dagger) + \text{rank}(\mathbf{J}_2^\dagger) + \dots + \text{rank}(\mathbf{J}_n^\dagger) = \text{rank} \begin{bmatrix} \mathbf{J}_1^\dagger & \mathbf{J}_2^\dagger & \dots & \mathbf{J}_n^\dagger \end{bmatrix} \quad (3.62)$$

holds. A more rigorous investigation of the constraints on multiple tasks, with respect to algorithmic singularities is presented in Chapter 5.

3.4.7 Numeric Stabilization

When solving the non-linear inverse kinematics problem numeric imperfections could cause the computed solution to drift away from the exact solution. By introducing an error

$$\dot{\boldsymbol{\theta}} = \dot{\mathbf{z}}_d - \mathbf{J}\dot{\mathbf{q}}, \quad (3.63)$$

and adding the term $\mathbf{K}\boldsymbol{\theta}$ via a feed-back loop to the desired task position $\dot{\mathbf{z}}_d$ in equation (3.50) the error dynamics result in

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{J}^\dagger(\dot{\mathbf{z}}_d + \mathbf{K}\boldsymbol{\theta}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\nabla h(\mathbf{q}) \\ \mathbf{J}\dot{\mathbf{q}} &= \dot{\mathbf{z}}_d + \mathbf{K}\boldsymbol{\theta} + \underbrace{\mathbf{J}(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})}_{\mathbf{0}}\nabla h(\mathbf{q}) \\ \mathbf{0} &= \dot{\boldsymbol{\theta}} + \mathbf{K}\boldsymbol{\theta}. \end{aligned} \quad (3.64)$$

This system is asymptotically stable for every pos.-def. matrix

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_O \end{bmatrix} \quad (3.65)$$

and leads to the so called Closed-Loop Inverse Kinematics (CLIK). With both error measures (3.14) or (3.16) the final Jacobian inverse solution

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \begin{bmatrix} I\mathbf{v}_{I\hat{E},ref} + \mathbf{K}_P\boldsymbol{\theta}_P \\ I\boldsymbol{\omega}_{IE,ref} + \mathbf{K}_O\boldsymbol{\theta}_O \end{bmatrix} \quad (3.66)$$

can be shown to lead to asymptotically stable error dynamics. For the error in axis-angle representation (3.14) this choice of an error measure $\boldsymbol{\theta}_O$ is an approximation but can be shown to leading to asymptotically stable error dynamics via Lyapunov. However, due to this approximation it is important to ensure that only sufficiently small errors occur and all elements in \mathbf{q} are not unbalanced due to e.g. different units and it is preferred to use the error measure (3.16) based on quaternions.

3.5 Kinetics

An accurate dynamic model of a manipulator describes the relation between the joint motor torques and the resulting motion of the manipulator and thus plays a key role in the design process of a robotic system. It is used for model based control algorithms and simulations to verify a given mechanical structure or to test certain tasks before designing a real system. Several methods to determine such a dynamic model exist and are well discussed in literature, like [59], [60] or [39]. The most popular might be the analytic Lagrange 2 - Method, but in this thesis the synthetic method of the Projection Equation, which was developed at the Institute of Robotics, at Johannes Kepler University Linz (see [60]), is used to determine a dynamic model.

The Projection Equation is a synthetic method, that sums up the linear and angular momentum of all subsystems of the robot, yielding the equation of motion for the whole system. The key advantages are:

The linear and angular momentum for every subsystem can be described in a different coordinate system.

The linear momentum and angular momentum of the *same* subsystem can be described in a different coordinate system.

No constraint forces enter calculation.

This makes the Projection Equation a very flexible, intuitive and easy to interpret systematic. It comes down to

$$\sum_{i=1}^N \underbrace{\left[\left(\frac{\partial {}_R\mathbf{v}_{IS}}{\partial \dot{\mathbf{q}}} \right)^T \left(\frac{\partial {}_R\boldsymbol{\omega}_{IS}}{\partial \dot{\mathbf{q}}} \right)^T \right]}_{A_i} \underbrace{\left[\begin{matrix} {}_R\dot{\mathbf{p}} + {}_R\tilde{\boldsymbol{\omega}}_{IR} {}_R\mathbf{p} - {}_R\mathbf{f}^e \\ {}_R\dot{\mathbf{L}} + {}_R\tilde{\boldsymbol{\omega}}_{IR} {}_R\mathbf{L} - {}_R\mathbf{M}^e \end{matrix} \right]}_{B_i} = \mathbf{0} \quad \text{with} \quad (3.67)$$

$N \dots$	Number of subsystems to be summed up
${}_R\mathbf{v}_{IS} \dots$	Absolute linear velocity of mass center, represented in body fixed coordinate system R with respect to the inertial frame I
${}_R\mathbf{p} \dots$	Linear momentum represented in body fixed coordinate system R ($\mathbf{p} = m {}_R\mathbf{v}_{IS}$ with mass m)
${}_R\mathbf{L} \dots$	Angular momentum represented in body fixed coordinate system R ($\mathbf{L} = \mathbf{J}_S {}_R\boldsymbol{\omega}_{IR}$ with moment of inertia \mathbf{J}_S)
${}_R\mathbf{f}^e \dots$	Acting forces represented in body fixed coordinate system R
${}_R\mathbf{M}^e \dots$	Acting torques represented in body fixed coordinate system R

With A_i the linear and angular momentum are projected into the direction of free motion. The linear and angular momenta, represented in the body-fixed coordinate frame R are

$$\begin{pmatrix} {}_R\mathbf{p} \\ {}_R\mathbf{L} \end{pmatrix} = \begin{bmatrix} m\mathbf{I} & \mathbf{0} \\ \mathbf{0} & {}_R\mathbf{J}^S \end{bmatrix} \begin{pmatrix} {}_R\mathbf{v}_{IS} \\ {}_R\boldsymbol{\omega}_{IS} \end{pmatrix} \quad (3.68)$$

with the identity matrix \mathbf{I} , the mass m and the inertia tensor at center of mass ${}_R\mathbf{J}^S$ in frame R and the acting forces and torques ${}_R\mathbf{f}^e$ and ${}_R\mathbf{M}^e$ on the sub-system.

The term B_i in (3.67) represents the law of conservation of linear and angular momentum for each sub-system, consisting of motor and link, can be independently expressed with respect to the body-fixed coordinate frame of the respective sub-system. The term A_i transforms the momenta of every sub-system according to the directions of free motion, depending on the mechanical constraints of each joint. Summing up over all N sub-systems finally leads to the equation of motion for the entire robot

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q} \quad (3.69)$$

with the mass matrix \mathbf{M} , the actuator forces \mathbf{Q} and $\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}})$, which contains gravitational, Coriolis and centrifugal terms. The dynamic system (3.69) can be represented in the convenient state-space representation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} \dot{\mathbf{q}} \\ \mathbf{M}(\mathbf{q})^{-1} \left(\mathbf{B}\mathbf{u} + \mathbf{J}^T(\mathbf{q}) {}_I\mathbf{F}_c - \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) \right) \end{pmatrix}, \quad (3.70)$$

with the state vector $\mathbf{x} = \left(\mathbf{q}^T, \dot{\mathbf{q}}^T \right)^T \in \mathbb{R}^{r \times r}$ and $\mathbf{u} = \mathbf{B}^{-1}\mathbf{Q}$.

If one is interested in a solution for \mathbf{x} classic Ordinary Differential Equation (ODE) solvers such as Euler, Simpson, Runge-Kutta, etc. can be employed. However, applying these techniques the inverse of the mass matrix has to be computed in every iteration, which can be very computationally heavy, i.e. the Gauss-Jordan elimination resembles an algorithm with a computation complexity of $\mathcal{O}(n^3)$. Considering the specific structure of model (3.70) more sophisticated strategies such as the recursive inverse with a computation complexity of $\mathcal{O}(n)$ [60] can be applied.

3.5.1 Inverse Dynamics

The inverse dynamics is a common tool to design some of the most simple non-linear control techniques such as torque feedforward or computed torque control. For flat or fully actuated systems like e.g. (3.76), the expression for \mathbf{u} as stated in (3.74) directly represents the inverse dynamics

$$\mathbf{u} = \mathbf{B}^{-1} \left(\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) \right) \quad (3.71)$$

for system (3.76).

3.6 Control

This section is dedicated to modeling and system classification, with a particular emphasis on their application in the context of robotic control. These critical steps lay the foundation for identifying and implementing the most effective control actions for a robotic systems.

3.6.1 Flat Systems

The class of differential flat systems possess very useful properties, that allows to control such highly non-linear systems in a particularly easy fashion, because such flat system can be linearized by endogenous state feedback and thus, simple linear control strategies can be applied.

A system is considered differentially flat when it possesses a set of outputs, equal in number to the inputs, that fully characterize both the entire system state and the inputs without necessitating the integration of the system, see Definition 2.

Definition 2 (Differential Flat Systems) *A system $\mathbf{f}(\mathbf{x}, \mathbf{u})$ with a state vector $\mathbf{x} \in \mathbb{R}^n$, input vector $\mathbf{u} \in \mathbb{R}^m$ and $\text{rank} \left(\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \right) = m$, is differentially flat if one can identify an output vector $\mathbf{y} \in \mathbb{R}^m$ that can be expressed as a function of the states \mathbf{x} , the input \mathbf{u} and derivatives of \mathbf{u} in the form*

$$\mathbf{y} = \phi(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \dots, \mathbf{u}^{(a)}) \quad (3.72)$$

and the states \mathbf{x} and input \mathbf{u} can be expressed as functions of the output \mathbf{y} and its derivatives in the form

$$\mathbf{x} = \psi(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(b)}) \quad (3.73)$$

$$\mathbf{u} = \eta(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(b+1)}), \quad (3.74)$$

and

$$\dim(\mathbf{y}) = m. \quad (3.75)$$

The output \mathbf{y} is then called flat output.

Example 1 (Dynamic model of a fully actuated mechanical structure) *Consider the equation of motion (3.69) and $\mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^m$ in the form*

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}\mathbf{u}. \quad (3.76)$$

With the states $\mathbf{x} = (\mathbf{q}^T, \dot{\mathbf{q}}^T)^T$ and the output

$$\mathbf{y} = \phi(\mathbf{x}) = \mathbf{q}, \quad (3.77)$$

the functions ψ and η result in

$$\mathbf{x} = \psi(\mathbf{y}, \dot{\mathbf{y}}) = \begin{pmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{pmatrix} \quad \text{and} \quad \mathbf{u} = \eta(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) = \mathbf{B}^{-1} (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}})), \quad (3.78)$$

with $a = 0$ and $b = 2$. This shows that all conditions from Definition 2 are satisfied, thus the output \mathbf{y} is flat and after a feedback linearization, linear control strategies can be applied to control system (3.76).

3.6.2 Stability of the Tracking Problem

A common problem in robotics is the tracking of a reference trajectory \mathbf{z}_{ref} with e.g. the end effector of a robot. Stability tests for dynamic systems can be employed to obtain advice on

applicable control actions or simply identify the behavior of such systems with respect to their stability. In order to avoid a stability proof for along the entire reference trajectory, one can define an error $\mathbf{e} = \mathbf{z}_{ref} - \mathbf{z}$ between the reference trajectory and the actual position \mathbf{z} , and instead test stability of the error dynamics in the origin $\mathbf{e} = \mathbf{0}$. It is important to mention that stability here is understood as stability defined by Lyapunov.

A simple form of differentially flat systems, such as (3.76), can be linearized with endogenous state feedback. The error dynamics of such non-linear models can then be formulated as a linear system

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} \quad (3.79)$$

and therefor techniques from linear control theory can be applied to test for stability or obtain advice for control actions.

If it is not possible to perform such a feedback linearization or e.g. the model parameters in (3.76) are imperfect, then the error dynamics can only be formulated in the non-linear form

$$\dot{\mathbf{e}} = \mathbf{f}(\mathbf{e}). \quad (3.80)$$

To test stability of (3.81), the direct (second) method of Lyapunov, presented in Definition 3, can be employed.

Definition 3 (Direct Method of Lyapunov) *Let $\mathbf{e}_e = \mathbf{0}$ be an equilibrium point of (3.76) and $\mathcal{D} \subseteq \mathbb{R}^n$ an open set in the neighborhood of $\mathbf{0}$. If a function $V(\mathbf{e}) : \mathcal{D} \rightarrow \mathbb{R}$ such that $V(\mathbf{e})$ is positive definite on \mathcal{D} and $\dot{V}(\mathbf{e})$ is negative semidefinite, then the equilibrium point $\mathbf{e}_e = \mathbf{0}$ is stable. If $\dot{V}(\mathbf{e})$ is negative definite, then the equilibrium point $\mathbf{e}_e = \mathbf{0}$ is asymptotically stable.*

In case of adaptive control strategies or teleoperation the error dynamics (3.81) additionally become time dependent and thus forming non-autonomous, non-linear error dynamics in the form

$$\dot{\mathbf{e}} = \mathbf{f}(t, \mathbf{e}). \quad (3.81)$$

Such systems can not be tested for stability anymore with Lyapunov's direct method. Furthermore, the definition of Lyapunov stability for non-autonomous systems changes. However, a method very similar to Lyapunov's approach called Barbalat's lemma can be employed to test stability of non-autonomous, non-linear systems as shown in Definition 4.

Definition 4 (Barbalat's Lemma) *If a scalar function $V(t, \mathbf{e}) : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$ satisfies the conditions*

1. $V(t, \mathbf{e})$ is bounded from below,
2. $\dot{V}(t, \mathbf{e}) \leq 0$ and
3. $\dot{V}(t, \mathbf{e})$ is uniformly continuous in time,

then $\lim_{t \rightarrow \infty} \dot{V}(t, \mathbf{e}) = 0$ holds.

For a well defined $V(t, \mathbf{e})$ with Barbalat's Lemma one can infer that with $\lim_{t \rightarrow \infty} \dot{V}(t, \mathbf{e}) = 0$ also the error $\lim_{t \rightarrow \infty} \mathbf{e} = 0$ and thus, according to Lyapunov's definition of stability for non-autonomous systems the error system is asymptotically stable with respect to the equilibrium point $\mathbf{e} = 0$.

However, proving the third condition in Barbalat's Lemma, $\dot{V}(t, \mathbf{e})$ being uniformly continuous in time, can be a very complex task and can conveniently be replaced by the sufficient condition for uniform continuity presented in Definition 5.

Definition 5 (Sufficient Condition for Uniform Continuity) *A differentiable function $f(t)$ is uniformly continuous, if the derivative $\frac{d}{dt}f(t)$ is bound.*

3.6.3 Holonomic & Non-Holonomic Systems & Coordinates

Holonomism or non-holonomism are important properties of systems and coordinates, since each requires distinct control strategies. Holonomic systems and coordinates are in general easier to handle than non-holonomic ones.

A combination of coordinates \mathbf{q} and corresponding velocities $\dot{\mathbf{s}}$, is called holonomic if $\dot{\mathbf{s}}$ can be integrated to a unique set of coordinates \mathbf{q} , such that

$$\mathbf{q} = \int \dot{\mathbf{s}} dt. \quad (3.82)$$

For a combination of non-holonomic coordinates it is not possible to integrate $\dot{\mathbf{s}}$ to a unique set of coordinates \mathbf{q} . Even though $\dot{\mathbf{s}}$ can be integrated, it will not lead to a unique \mathbf{q} and therefor pseudo coordinates can be defined, such that

$$\boldsymbol{\pi} := \int \dot{\mathbf{s}} dt. \quad (3.83)$$

Obviously, if this integral (for non-holonomic coordinates) does not lead to a unique set of $\mathbf{q} \neq \boldsymbol{\pi}$, the integral must be path-dependent or in other words, the argument $\boldsymbol{\pi}$ is not curl free. Thus, a combination of coordinates is holonomic if

$$\nabla \times \dot{\mathbf{s}} \equiv \mathbf{0}. \quad (3.84)$$

However, the formulation of a mathematical model of the kinetics of an actuated mechanism simplifies, if non-holonomic coordinates, e.g. the angular velocities $\boldsymbol{\omega}$ are employed.

A system is usually called holom or non-holonom if the corresponding Pfaffian system of velocity constraints is holonom or non-holonom. A Pfaffian system has the form

$$\mathbf{0} = \mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \left[\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m \right] \dot{\mathbf{q}}, \quad (3.85)$$

for $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{n \times m}$. To test if a Pfaffian system is holonom Frobenius' theorem, shown in Defintion 6, can be employed.

Definition 6 (Frobenius' Theorem) *A regular smooth distribution is integrable if and only if it is involutive.*

If the vectors of a Pfaffian system are involutive can be checked by employing Theorem 7 utilizing the Lie brackets.

Definition 7 *A Pfaffian system $\mathbf{0} = \mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = [\mathbf{a}_1, \mathbf{a}_2]\dot{\mathbf{q}}$ is holonom, if and only if the result of the Lie bracket*

$$\mathbf{a}_3 = [\mathbf{a}_1, \mathbf{a}_2]_{\mathcal{L}} := \mathcal{L}_{\mathbf{a}_1}\mathcal{L}_{\mathbf{a}_2} - \mathcal{L}_{\mathbf{a}_2}\mathcal{L}_{\mathbf{a}_1} = \frac{\partial \mathbf{a}_1}{\partial \mathbf{q}}\mathbf{a}_2 - \frac{\partial \mathbf{a}_2}{\partial \mathbf{q}}\mathbf{a}_1 \quad (3.86)$$

does not expand the $\text{span}\{\mathbf{a}_1, \mathbf{a}_2\}$ and thus

$$\text{span}\{\mathbf{a}_1, \mathbf{a}_2\} = \text{span}\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\} \quad (3.87)$$

holds.

An alternative and sometimes faster way to check if a Pfaffian system is holonom can be applied following Theorem 8.

Definition 8 *A regular Pfaffian system is holonom if the necessary and sufficient condition*

$$\left(\mathbf{A}^\perp\right)^T \left[\left(\frac{\partial \mathbf{A}_k}{\partial \mathbf{q}} \right) - \left(\frac{\partial \mathbf{A}_k}{\partial \mathbf{q}} \right)^T \right] \mathbf{A}^\perp \equiv 0 \quad \forall k \in [1, n] \quad (3.88)$$

holds. \mathbf{A}_k represents the k -th row of \mathbf{A} and \mathbf{A}^\perp denotes the orthogonal complement to \mathbf{A} .

3.7 Mobile Platform with Manipulator

The holonomic mobile platform with four mecanum wheels and a six DoF manipulator is often employed throughout this thesis for verification and validation purposes of newly developed algorithms. In the following, the modelling of such a robotic platform, applying the concepts introduced in the previous sections, will be presented.

Figure 3.6(a) shows the robotic system SPSBot, which is used in the SPS tunnel for radiation measurement and testing of BLM sensors with the attached manipulator. The SPSBot consists of four mecanum wheels and a six DoF Kinova manipulator. The coordinate frame I represents the inertial frame, frame B is fixed to the rigid robot base, frame M denotes the base of the six DoF manipulator and the end effector frame is indicated with E .

In Figure 3.6(b) a detailed drawing of one of the mecanum wheels is presented, which will be important to understand the mapping from the angular wheel velocities to the actual movement of the robot base. The mecanum wheel is presented with respect to the body fixed base frame B and for the sake of simplicity another frame W , that is aligned with the rotation axis of the wheel, has been introduced.

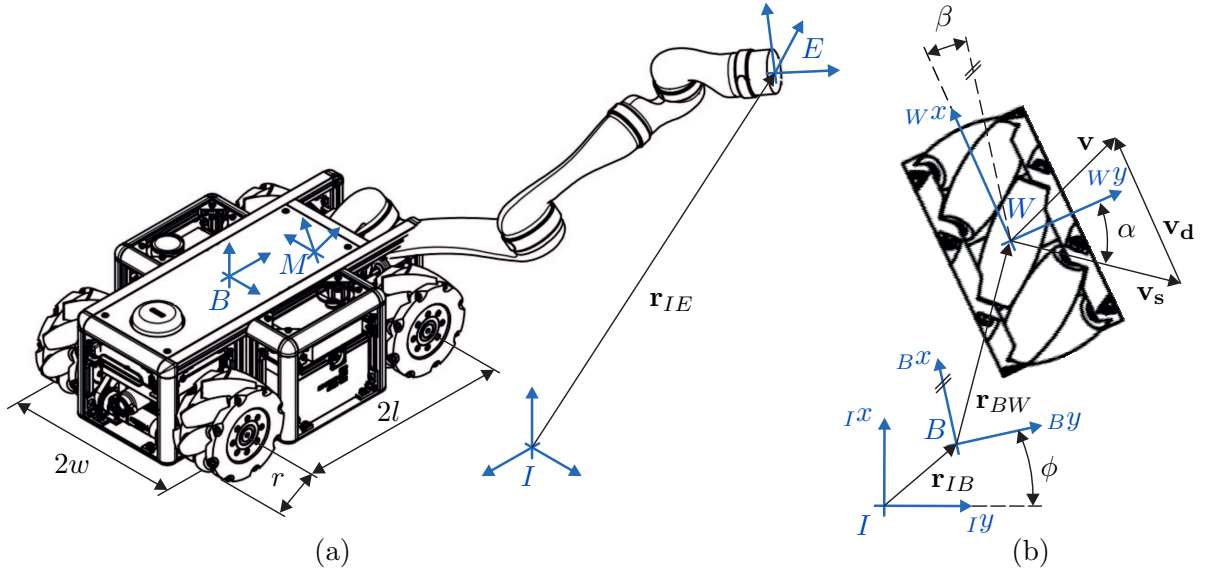


Figure 3.6: (a) The SPSBot from the CERN robotic fleet has four mecanum wheels and a six DoF manipulator attached to its base. (b) A detailed view on one of the four mecanum wheels.

In order to analyze the kinematics of the robot, first of all we need to take a closer look at the velocities related to the mecanum wheel in Figure 3.6(b). In general the wheel can be rotated with respect to the body fixed base frame by an angle β , therefore another frame W was introduced to simplify the relations between the velocities on the wheel. A mecanum wheel possesses multiple smaller wheels that are mounted on the radius of the bigger wheel and are rotated by an angle α .

The velocity in drive direction (Wx -direction) is denoted as \mathbf{v}_d and the velocity in sliding direction (perpendicular to small wheels axes, which are rotated by α) is denoted as \mathbf{v}_s , leading to the resulting velocity

$${}^W\mathbf{v} = {}^W\mathbf{v}_s + {}^W\mathbf{v}_d = \begin{pmatrix} {}^Wv_x \\ {}^Wv_y \end{pmatrix} = \|\mathbf{v}_s\| \begin{pmatrix} -\sin \alpha \\ \cos \alpha \end{pmatrix} + \|\mathbf{v}_d\| \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \quad (3.89)$$

Solving (3.89) for $\|\mathbf{v}_s\|$ and $\|\mathbf{v}_d\|$ yields

$$\|\mathbf{v}_s\| = \frac{v_y}{\cos \alpha} \quad \text{and} \quad \|\mathbf{v}_d\| = {}^Wv_x + {}^Wv_y \tan \alpha. \quad (3.90)$$

Then with the wheel radius r , the wheel velocity \dot{q}_W can be formulated as

$$\dot{q}_W = \frac{\|\mathbf{v}_d\|}{r} = \frac{1}{r}({}^Wv_x + {}^Wv_y \tan \alpha) = \begin{bmatrix} \frac{1}{r} & \frac{\tan \alpha}{r} \end{bmatrix} {}^W\mathbf{v}. \quad (3.91)$$

Thus, with (3.91) an equation was derived that describes, how a resulting velocity ${}^W\mathbf{v}$ in the frame W relates to the angular wheel velocity \dot{q} of a mecanum wheel. Now, everything that is left to show, is how a desired translation ${}^I\mathbf{v}_{IB} = \dot{\mathbf{r}}_{IB}$ and rotation ${}^I\boldsymbol{\omega}_{IB}$ of the chassis can be expressed in the frame W . Following the concepts introduced in Section 3.2, ${}^W\mathbf{v}$ can be obtained in the form

$${}^W\mathbf{v} = \mathbf{R}_{WB} (\mathbf{R}_{BII}\mathbf{v}_{IB} + (\mathbf{R}_{BII}\boldsymbol{\omega}_{IB}) \times {}^B\mathbf{r}_{BW} + {}^B\dot{\mathbf{r}}_{BW}). \quad (3.92)$$

Considering that the wheels and base are connected rigidly we can set $\dot{\mathbf{r}}_{BW} = 0$ and using the base velocity ${}_I\dot{\mathbf{z}}_B = ({}_I v_x, {}_I v_y, {}_I \omega_z)^T$ and ${}_B\mathbf{r}_{BW} = (x_W, y_W)^T$, (3.92) can be rewritten in the form

$${}_W\mathbf{v} = \mathbf{R}_{WB}^* \begin{bmatrix} 1 & 0 & -y_W \\ 0 & 1 & x_W \end{bmatrix} \mathbf{R}_{BI} {}_I\dot{\mathbf{z}}_B. \quad (3.93)$$

Substituting (3.93) in (3.91) finally yields

$$\dot{q}_{W,i} = h_i(\phi) {}_I\dot{\mathbf{z}}_B = \begin{bmatrix} \frac{1}{r_i} & \frac{\tan \alpha_i}{r_i} \end{bmatrix} \begin{bmatrix} \cos \beta_i & \sin \beta_i \\ -\sin \beta_i & \cos \beta_i \end{bmatrix} \begin{bmatrix} 1 & 0 & -y_{W,i} \\ 0 & 1 & x_{W,i} \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} {}_I\dot{\mathbf{z}}_B, \quad (3.94)$$

with the index i for the i -th wheel. By evaluating (3.94) we obtain

$$h_i(\phi) = \frac{1}{r_i \cos \alpha_i} \begin{bmatrix} \cos(\alpha_i + \beta_i + \phi) \\ \sin(\alpha_i + \beta_i + \phi) \\ x_{W,i} \sin(\beta_i + \alpha_i) - y_{W,i} \cos(\beta_i + \alpha_i) \end{bmatrix}^T. \quad (3.95)$$

To obtain the Jacobian matrix for the entire robotic system, the wheel angles will be collected in the vector $\mathbf{q}_W = (q_{W,1}, q_{W,2}, q_{W,3}, q_{W,4})^T$ and for all m wheels on the robotic platform one $h_i(\phi)$ will be stacked as a row entry in the matrix $\mathbf{H}(\phi) \in \mathbb{R}^{m \times 3}$, such that

$$\dot{\mathbf{q}}_W = \mathbf{H}(\phi) {}_I\dot{\mathbf{z}}_B = \begin{bmatrix} h_1(\phi) & h_2(\phi) & h_3(\phi) & h_4(\phi) \end{bmatrix}^T {}_I\dot{\mathbf{z}}_B, \quad (3.96)$$

with

$$\mathbf{H}(\phi) = \frac{1}{r} \begin{bmatrix} \sqrt{2} \sin(\pi/4 + \phi) & -\sqrt{2} \cos(\pi/4 + \phi) & -l - w \\ \sqrt{2} \cos(\pi/4 + \phi) & \sqrt{2} \sin(\pi/4 + \phi) & l + w \\ \sqrt{2} \sin(\pi/4 + \phi) & -\sqrt{2} \cos(\pi/4 + \phi) & l + w \\ \sqrt{2} \cos(\pi/4 + \phi) & \sqrt{2} \sin(\pi/4 + \phi) & -l - w \end{bmatrix} \quad (3.97)$$

A very simple form of (3.96) can be derived when using the parameters from the specific robot shown in Figure 3.6 with $x_{W,i} = l$, $y_{W,i} = w$, the radii $r_i = r$, the angles $\alpha_i = \frac{\pi}{4}$ and $\beta_i = 0$ for the and representing the desired chassis velocities in the base frame ${}_B\dot{\mathbf{z}}$, hence setting $\phi = 0$, results in

$$\dot{\mathbf{q}}_W = \mathbf{H}(0) {}_B\dot{\mathbf{z}}_B = \frac{1}{r} \underbrace{\begin{bmatrix} 1 & -1 & -l - w \\ 1 & 1 & l + w \\ 1 & -1 & l + w \\ 1 & 1 & -l - w \end{bmatrix}}_{\mathbf{J}_B^{-1}} {}_B\dot{\mathbf{z}}_B. \quad (3.98)$$

The matrix $\mathbf{H}(0)$ directly represents the inverse Jacobian for the wheeled base \mathbf{J}_B^{-1} and an expression for the Jacobian can be obtained by employing the pseudo inverse

$${}_B\dot{\mathbf{z}}_B = \left(\mathbf{H}(0)^T \mathbf{H}(0) \right)^{-1} \mathbf{H}(0)^T \dot{\mathbf{q}}_W = \frac{r}{4} \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -\frac{1}{l+w} & \frac{1}{l+w} & \frac{1}{l+w} & -\frac{1}{l+w} \end{bmatrix}}_{\mathbf{J}_B} \dot{\mathbf{q}}_W. \quad (3.99)$$

Thus, with the previous we have now derived the Jacobian matrix $\mathbf{J}_B \in \mathbb{R}^{3 \times 4}$ for the base of the SPSBot. The Jacobian matrix $\mathbf{J}_M \in \mathbb{R}^{6 \times 6}$ for the six DoF manipulator can be obtained by

following the recursive procedure of the kinematic chain as demonstrated in Section 3.3.2 using (3.33). In the last step, concerning kinematics, the two Jacobian matrices need to be combined to obtain the final geometric Jacobian $\mathbf{J} \in \mathbb{R}^{6 \times 10}$. The missing three rows in \mathbf{J}_B have to be filled with zero rows accordingly, yielding $\mathbf{J}_B^* \in \mathbb{R}^{6 \times 4}$. The combined system with $\mathbf{q}_K = [\mathbf{q}_W^T, \mathbf{q}_M^T]^T$, $\mathbf{q}_M = [q_5, \dots, q_{10}]^T$ and the end effector position and orientation ${}_I\mathbf{z}_E$ can be written in the form

$$\dot{\mathbf{z}}_E = \begin{bmatrix} \mathbf{J}_B^* & \mathbf{J}_M \end{bmatrix} \dot{\mathbf{q}}_K. \quad (3.100)$$

To solve the inverse kinematics problem, the techniques presented in Section 3.4 for redundant mechanisms can be applied.

In order to analyze the dynamics of the SPSBot it is first necessary to choose the describing minimal coordinates and minimal velocities. This choice is not unique and has significant influence on the complexity of the resulting model. The minimal coordinates can be chosen as $\mathbf{q} = [\mathbf{q}_W^T, x, y, \gamma, \mathbf{q}_M^T]^T$ and in order to simplify the dynamic model we can choose the non-holonomic minimal velocities $\dot{\mathbf{s}} = [\dot{x}, \dot{y}, \dot{\gamma}, \dot{\mathbf{q}}_M^T]^T$, with the reduced dimension according to the four velocity constraints in (3.96) such that

$$\dot{\mathbf{q}} = \frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{s}}} \dot{\mathbf{s}} = \frac{1}{r} \left[\begin{array}{ccc|c} \sqrt{2} \sin(\pi/4 + \phi) & -\sqrt{2} \cos(\pi/4 + \phi) & -l - w & 0 \\ \sqrt{2} \cos(\pi/4 + \phi) & \sqrt{2} \sin(\pi/4 + \phi) & l + w & 0 \\ \sqrt{2} \sin(\pi/4 + \phi) & -\sqrt{2} \cos(\pi/4 + \phi) & l + w & 0 \\ \sqrt{2} \cos(\pi/4 + \phi) & \sqrt{2} \sin(\pi/4 + \phi) & -l - w & 0 \\ r & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ 0 & 0 & r & 0 \\ \hline & \mathbf{0} & & r\mathbf{I} \end{array} \right] \dot{\mathbf{s}}. \quad (3.101)$$

Following (3.67) the equation of motion can be derived and with the state $\mathbf{x} = (\mathbf{q}^T, \dot{\mathbf{s}}^T)^T$ the state-space representation yields the holonom dynamic system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} \frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{s}}} \dot{\mathbf{s}} \\ \mathbf{M}(\mathbf{q})^{-1} (\mathbf{B}\mathbf{u} - \mathbf{g}(\mathbf{q}, \dot{\mathbf{s}})) \end{pmatrix}. \quad (3.102)$$

As already shown in Example 1 a system of the form (3.102) resembles a flat system and thus can be feedback linearized and the stability test of the linear error system $\dot{\mathbf{e}} = \mathbf{A}\mathbf{e}$ can be tested with linear control strategies.

A Novel Design Optimization Technique for Simultaneous Optimization of Topology and Geometry

ABSTRACT. Big research facilities are often confronted with very specialized automation problems in hazardous, radioactive and semi-structured environments for its particle accelerators, test rigs or other experiments. These frequently lead to specific requirements that do not allow the usage of common industrial robots. Thus, a design problem with almost no restrictions on the actual robot topology, but very hard requirements concerning workspace, allowed robot space, payload, robot weight and accuracy (due to elasticity/error propagation) has to be solved. This chapter presents an approach to this problem, which was applied to find an optimal robotic design for inspection and maintenance tasks in CERN's Future Circular Collider (FCC) and inspection of RF cavities.

Chapter Content

4.1	Design Optimization Algorithm	62
4.2	Application: Cavity In- spection Robot	68
4.3	Application: FCC Ma- nipulator	76
4.4	Conclusion	79

Related Publications

- | | | |
|------|---------------------------------|--|
| [61] | Gamper H., Luthi A., et al. | Kinematic Model Pruning: A Design Optimization Technique for Simultaneous Optimization of Topology and Geometry - MDPI |
| [62] | Gamper H., Luthi A., et al. | Design Optimization of Quality Inspection Robots for Particle Accelerator Components - ECCOMAS |
| [63] | Luthi A., Macpherson A., et al. | Camera Placement in a Short Working Distance Optical Inspection System for RF Cavities - SRF |
-

Power plants and big industrial or scientific facilities like the European Organization for Nuclear Research (CERN), are often confronted with very special automation problems in complex environments for their laboratories, experiments or test rigs. These frequently lead to specific requirements that do not allow the usage of standard industrial robots. Thus, a design problem with almost no restrictions on the actual robot topology, but very hard requirements concerning other parameters like workspace, allowed robot space, dexterity and accuracy have to be solved.

This section presents an approach to this problem by setting up a design optimization algorithm that combines optimization strategies and model pruning in order to find an optimal design. A initial surrogate model defines the design space which will be optimized by model pruning and lead to a optimized topology and geometry as visualized in Figure 4.1.



Figure 4.1: Workflow of the design optimization algorithm

The algorithm is able to optimize the degree of freedom, the robot link lengths and other kinematic or dynamic criteria, while ensuring that the mechanical structure reaches the desired end-effector position, avoids self collisions and collisions with its surrounding. Similar design optimization problems concerning manipulability measures, error measures and torque minimization are discussed in [64] and [65], but without minimizing the DoF of the mechanical structure. Other work that optimizes also the topology mostly referred to as structural synthesis can be found in [66] and [67], but require a much bigger framework whereas the presented algorithm only requires simple modifications in the objective function of a common optimization problem. This will be demonstrated with two applications:

Design optimization of a surface inspection robot for Radio Frequency (RF) Cavities as used in the Large Hardron Collider (LHC), the Linear Accelerator (LINAC) or the Future Circular Collider (FCC).

Design optimization of a manipulator for the 100 km long FCC.

Section 4.1 describes the generic algorithm in detail and in Section 4.2 and 4.3 it is applied to two specific problems, the cavity inspection and the manipulator for FCC. The last Section 4.4 summarizes the results and draws conclusions concerning the existing and future work.

4.1 Design Optimization Algorithm

The presented algorithm is capable of modifying a initial and thus, non-optimal robot topology and geometry in a way, such that the final result provides an optimal and practically feasible solution with respect to certain objectives:

Minimize the DoF.

Minimize the length of each robot link.

Minimize kinematic and dynamic criteria of the mechanical structure.

These objectives need to be constrained in a way, such that the robot is able to reach all desired positions, avoid self collisions as well as with the environment. The algorithm is in general applicable under following constraints:

The mechanical structure can contain arbitrary joint types, but only translational joints or joints with parallel rotational axes can be used for reducing the DoF.

The additional kinematic and dynamic performance criteria subject to optimization should in best case be complementary, but never be in contradiction with each other.

In Section 4.1.1 a parametrized model of the initial mechanical structure, including kinematics and dynamics, will be defined. General assumptions on the collision avoidance are discussed in Section 4.1.2. The formulation of the optimization problem is shown in Section 4.1.3 and the applied objective function is described in Section 4.1.4.

4.1.1 Kinematic and Dynamic Model

The forward kinematics $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^6$ with n degree of freedom, can be written in the form

$$\mathbf{z} = \mathbf{f}(\mathbf{p}, \mathbf{q}), \quad (4.1)$$

with the generalized joint coordinates $\mathbf{q} \in \mathbb{R}^n$, the Cartesian position and orientation $\mathbf{z} \in \mathbb{R}^6$ and the N geometric parameters in $\mathbf{p} \in \mathbb{R}^N$. An explicit solution for the inverse kinematics is not needed, since it will be taken into account by non-linear equality constraints in the optimization problem.

The forward dynamics or the equation of movement for the entire robot can be written as

$$\mathbf{M}(\mathbf{p}, \mathbf{q}) \ddot{\mathbf{q}} + \mathbf{g}(\mathbf{p}, \mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}, \quad (4.2)$$

with the mass matrix $\mathbf{M}(\mathbf{p}, \mathbf{q})$, the non-linear term $\mathbf{g}(\mathbf{p}, \mathbf{q}, \dot{\mathbf{q}})$ containing gravitational, centrifugal and Coriolis terms and the actuator forces and torques \mathbf{Q} .

4.1.2 Collision Avoidance

To reduce the computational cost of the simulation, it was assumed that the mechanical design either prevents two consecutive links to collide or the collision avoidance is included in the upper and lower bounds of the corresponding joint angles. Thus, for a robot with n_R serial links, the

number of possible self collisions c_{RR} to be monitored or in other words the number of link pairs that have to be checked to identify self collisions is

$$c_{RR} = \frac{1}{2} \frac{n_R!}{(n_R - 2)!}. \quad (4.3)$$

To monitor collisions between n_R robot links and n_E collision objects in the environment of the robot in general

$$c_{RE} = n_R n_E \quad (4.4)$$

checks have to be performed.

4.1.3 Problem Formulation

The optimization was set up as a non-linear global optimization problem with non-linear equality and inequality constraints

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{p}} \quad & J(\mathbf{x}, \mathbf{p}) \\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x}, \mathbf{p}) - \mathbf{z}_d = \mathbf{0} \\ & -\mathbf{c}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0} \\ & \mathbf{ub}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0} \\ & \mathbf{lb}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0} \end{aligned} \quad (4.5)$$

with the objective function $J(\mathbf{x}, \mathbf{p})$ and the parameters \mathbf{x} and \mathbf{p} . The vector \mathbf{x} contains the generalized coordinates in joint space for μ different desired Cartesian positions. The vector \mathbf{p} contains the N geometric parameters, in particular the lengths of the robot links. Thus,

$$\mathbf{x} = [\mathbf{q}_1^T, \mathbf{q}_2^T, \dots, \mathbf{q}_\mu^T]^T \in \mathbb{R}^{n\mu} \quad (4.6)$$

and

$$\mathbf{p} \in \mathbb{R}^N. \quad (4.7)$$

The inverse kinematics is included with the equality constraint

$$\mathbf{f}(\mathbf{x}, \mathbf{p}) - \mathbf{z}_d = \mathbf{0}. \quad (4.8)$$

The vector function

$$\mathbf{c}(\mathbf{x}, \mathbf{p}) \in \mathbb{R}^{c_{RR}+c_{RE}} \quad (4.9)$$

contains the minimal distances according to self-collisions and collisions with the environment. The vector functions

$$\mathbf{ub}(\mathbf{x}, \mathbf{p}), \mathbf{lb}(\mathbf{x}, \mathbf{p}) \in \mathbb{R}^{n\mu+N} \quad (4.10)$$

are upper and lower boundaries on the joint angles and link lengths.

4.1.4 Objective Function

As already discussed at the beginning of Section 4.1, the desired objective function should minimize the DoF, the robot link lengths and other kinematic and dynamic criteria. This is

expressed as linear combination of the objectives

$$J(\mathbf{x}, \mathbf{p}) = \underbrace{\mathbf{k}^T \mathbf{g}(\mathbf{p})}_{J_1(\mathbf{p})} + \underbrace{\mathbf{\Gamma}^T(\mathbf{x}, \mathbf{p}) \mathbf{K} \mathbf{\Gamma}(\mathbf{x}, \mathbf{p})}_{J_2(\mathbf{x}, \mathbf{p})}. \quad (4.11)$$

In the following Sections 4.1.4 and 4.1.4 the effect of J_1 and J_2 on the optimization result will be discussed. The term J_1 penalizes the length of the robot links with a mapping $\mathbf{g} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ and the weighting factor $\mathbf{k} \in \mathbb{R}^N$. The term J_2 represents ν kinematic and dynamic criteria summarized in $\mathbf{\Gamma}(\mathbf{x}, \mathbf{p}) \in \mathbb{R}^{n\mu\nu}$, which is weighted with the diagonal matrix $\mathbf{K} \in \mathbb{R}^{(n\mu\nu) \times (n\mu\nu)}$.

Minimizing the DoF and Link Lengths

The goal of term J_1 is to penalize the robot link lengths in a way such that both, the link lengths and the DoF of the robot will be minimized. In order to achieve this behavior and minimize the DOF of the robot, a set of candidate links are introduced where the two joints connecting them with respective predecessor and successor are of the same type and their axes are parallel. Then, the total link length of the mechanical structure will be minimized and if possible (with respect to all constraints) link lengths will be set to zero and thus indicate a possible reduction of DoF. This requires a specific type of function for \mathbf{g} in (4.11), as defined in the following Definition 9.

Definition 9 (Pruning Function) A vector function $\mathbf{g} = [g_1(l_1) \ g_2(l_2) \ \dots \ g_N(l_N)] : \mathbb{R}^N \rightarrow \mathbb{R}^N$ with argument $\mathbf{p} = [l_1 \ l_2 \ \dots \ l_N]^T \in \mathbb{R}^N$ that satisfies

$$\frac{\partial \mathbf{g}}{\partial \mathbf{p}} > 0 \quad \forall \ l_i > 0, \quad i \in \{1, 2, \dots, N\} \quad (4.12)$$

and

$$\frac{\partial^2 \mathbf{g}}{\partial \mathbf{p}^2} < 0 \quad \forall \ l_i > 0, \quad i \in \{1, 2, \dots, N\}. \quad (4.13)$$

Condition (4.12) ensures that the link lengths will be minimized and condition (4.13) facilitates minimization of the number of DoF in the mechanical structure. In the following Example 2, the behavior of the optimization when using a *Pruning Function* will be demonstrated and compared to the behavior when using a quadratic function for \mathbf{g} in (4.11).

Example 2 The end-effector of a $N = 2$ link planar manipulator with two DoF should be able to reach exactly one point at position $\|\mathbf{z}_d\| = 5$. In Figure 4.2(a) the initial design with two DoF and the link lengths l_1^a and l_2^a is shown. Departing from this initial design, the optimization should find an optimal design with minimal link lengths and a minimal number of DoF. An objective function $J(\mathbf{x}, \mathbf{p}) = J_1(\mathbf{p}) = \mathbf{k}^T \mathbf{g}(\mathbf{p})$ with $\mathbf{k}^T = [1 \ 1]$ will be used. For demonstration purposes the optimization will be launched two times with different functions for \mathbf{g} . Once with a *Pruning Function*

$$\mathbf{e}(\mathbf{p}) = [e(l_1) \ e(l_2)]^T = [\arctan(l_1) \ \arctan(l_2)]^T. \quad (4.14)$$

and once with a quadratic function $\mathbf{h} : \mathbb{R}^N \rightarrow \mathbb{R}^N$,

$$\mathbf{h}(\mathbf{p}) = \begin{bmatrix} h(l_1) & h(l_2) \end{bmatrix}^T = \begin{bmatrix} l_1^2 & l_2^2 \end{bmatrix}^T, \quad (4.15)$$

with the well known properties

$$\frac{\partial \mathbf{h}}{\partial \mathbf{p}} > 0, \quad \frac{\partial^2 \mathbf{h}}{\partial \mathbf{p}^2} > 0 \quad \forall \quad l_i > 0, \quad i \in \{1, 2\}. \quad (4.16)$$

Note that the second derivative of the quadratic function is, unlike the Pruning Function, greater than zero.

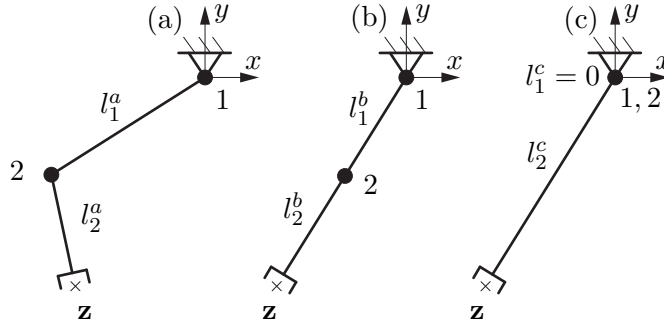


Figure 4.2: Optimization effects using different function types for J_1

The result of the optimization with the quadratic objective function \mathbf{h} is shown in Figure 4.2(b). It is clear that the optimization converged to an optimum, since the links lie on a straight line from the base to the desired end-effector position in Euclidean space. But it is also clear that only one DoF would be sufficient to reach this position. Thus, the quadratic objective function minimizes the total length $l_1 + l_2$, but splits up this length equally over both links such that $l_1 = l_2$ and hence does not minimize the DoF.

This behavior becomes more clear when looking at the surface and contour plots of the objective function $J_1(\mathbf{p})$ over the variables l_1 and l_2 as shown in Figures 4.3 and 4.4, respectively. The black line in the $l_1 l_2$ -plane represents the possible combinations of l_1 and l_2 with which the desired end-effector position can be reached and $l_1 + l_2 = \|\mathbf{z}_d\| = 5$ holds. The projection of this line on the surface of $J_1(\mathbf{p})$ leads to the curved line $J_1([l_1, \|\mathbf{z}\| - l_1])$ from which the optimal combination of l_1 and l_2 has to be chosen. The dashed red line visualizes the space of optimal combinations of l_1 and l_2 for arbitrary \mathbf{z} , which is identical with the space of combinations where $l_1 = l_2$. This becomes even more clear when looking at the contour plot in Figure 4.4. The black line $l_1 + l_2 = \|\mathbf{z}_d\|$ shows a -45° slope and the contour lines are circles of different diameter centered at the origin. Thus, the optimal solution can be found where the line $l_1 + l_2 = \|\mathbf{z}_d\|$ intersects exactly once with a contour line or in other words, where the line $l_1 + l_2 = \|\mathbf{z}_d\|$ is a tangent to the contour line. For arbitrary \mathbf{z} , this leads to optimal solution $l_1 = l_2$.

In order to minimize the DoF the distance should not be split up equally, but assigned to only one link, while the other link length will be set to 0. The corresponding joint to this link can then be removed, which means decreasing the DoF by 1. This can be achieved by using the Pruning Function (4.14) for \mathbf{g} in (4.11). Running the same optimization problem again with the new objective function, the total link length $l_1 + l_2$ is still a minimum (straight line in Euclidean space), but is now assigned to only one link. This indicates that not more than one DoF is

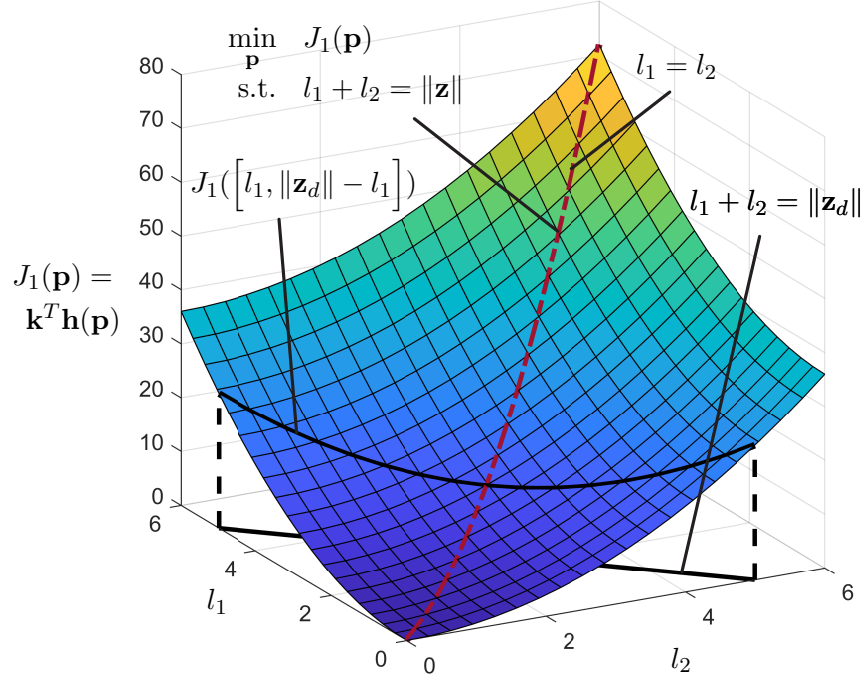


Figure 4.3: Surface plot of objective function $J_1(\mathbf{p}) = \mathbf{k}^T \mathbf{h}(\mathbf{p})$

necessary to reach the position \mathbf{z} , see Figure 4.2(c). Thus, the corresponding joint i with $l_i = 0$ can be removed and the link lengths as well as the DoF was minimized.

Again, this behavior becomes more clear when looking at the surface and contour plots of the objective function $J_1(\mathbf{p})$ over the variables l_1 and l_2 as shown in Figures 4.5 and 4.6, respectively. The black line in the $l_1 l_2$ -plane represents the possible combinations of l_1 and l_2 with which the desired end-effector position can be reached and $l_1 + l_2 = \|\mathbf{z}_d\| = 5$ holds. The projection of this line on the surface of $J_1(\mathbf{p})$ leads to the curved line $J_1([l_1, \|\mathbf{z}\| - l_1])$ from which the optimal combination of l_1 and l_2 has to be chosen. The dashed red lines visualize the space of optimal combinations of l_1 and l_2 for arbitrary \mathbf{z} . Compared to Figure 4.3 it can now be seen that for every \mathbf{z}_d two solutions ($l_1 = 0, l_2 = \|\mathbf{z}_d\|$) and ($l_1 = \|\mathbf{z}_d\|, l_2 = 0$) with the same priority exist. Both solution lie at the lower boundaries of the parameters l_1 and l_2 . This becomes even more clear when looking at the contour plot in Figure 4.6. The black line $l_1 + l_2 = \|\mathbf{z}_d\|$ shows again a -45° slope, but now the contour lines are shaped in way such that the optimal solutions lie at the boundaries of $l_1 + l_2 = \|\mathbf{z}_d\|$. The optimal solution can be found by restricting the parameters with upper boundaries in the optimization problem leading to only one solution e.g. ($l_1 = \|\mathbf{z}_d\|, l_2 = 0$).

Minimizing Kinematic and Dynamic Performance Criteria

The term J_2 of (4.11) accounts for arbitrary kinematic or dynamic criteria summarized in vector $\Gamma(\mathbf{x}, \mathbf{p})$ and multiplied with a weighting matrix \mathbf{K} . It is important that multiple criteria do not influence each other too much or in best case are complementary, in order to avoid ill conditioned

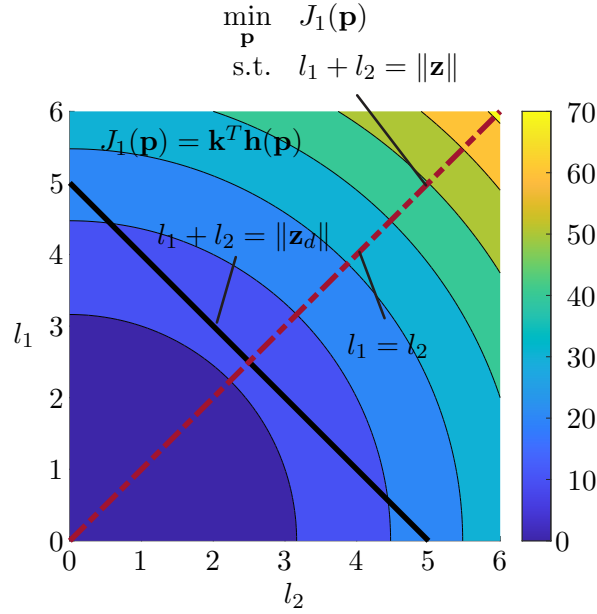


Figure 4.4: Contour plot of objective function $J_1(\mathbf{p}) = \mathbf{k}^T \mathbf{h}(\mathbf{p})$

optimization problems. Possible criteria could be the motor torque, distance from singularities, error propagation through the mechanical structure or kinematic and dynamic manipulabilities. Examples for such criteria are shown in Sections 4.2.3 and 4.3.2.

4.2 Application: Cavity Inspection Robot

Radio Frequency Cavities (see Figure 4.7) are performs the linear acceleration of charged particles in straight sections of accelerator machines and thus make up one of the key elements in a collider complex [68]. The cavities structure and geometry define their specific radio-frequency at which the strong electromagnetic field, created inside the tubes, oscillates to accelerate each particle passing through. The inner surface quality of the cavities is critical to withstand the high energy densities, since every scratch or crack would lead to higher local resistance, thus a rapid increase in temperature during operation and in the end to failure of the system. Therefore, some kind of automated, mechanical structure has to follow the complex cavity geometry and take record of the surface quality after full assembly of the cavities. Finding the optimal topology of such a mechanical structure with respect to certain constraints like collision avoidance for different cavity types and minimal error propagation in direction perpendicular to the cavity surface, is a great example for the generic problem described in the introduction of this chapter. Currently, several different system have been developped and are able to partially scan cavities, see [69], [70], [71] and [72]. However, those previously developed cavity inspection systems were extensively tested at CERN, but did not satisfy the specific requirements concerning the level of automation, accuracy, repeatability and how much of the inner cavity surface could be inspected and mapped. The first prototype of the system developed at CERN and presented in [73], was not able to scan all three cavities with one robotic arm, but had to use two different arms. The aim of this design optimization was to find one topology and geometry that can handle all three cavity types and thus increase robustness and level of automation, while decreasing the cost of such a system.

The main challenge for a robotic system is the complex workspace and especially the difference

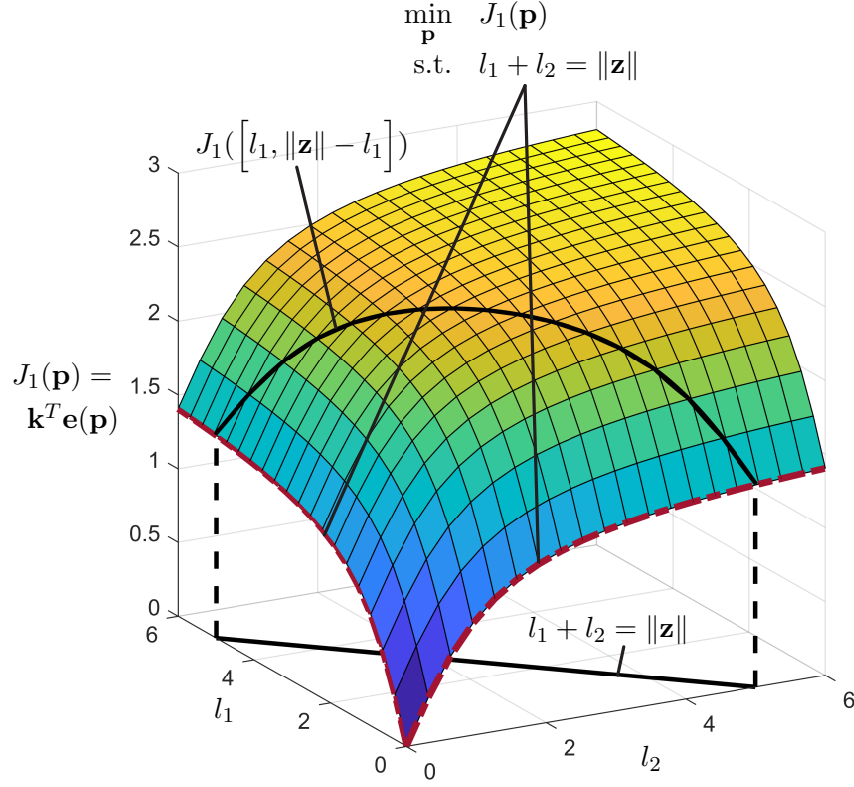


Figure 4.5: Surface plot of objective function $J_1(\mathbf{p}) = \mathbf{k}^T \mathbf{e}(\mathbf{p})$

in diameter of the entrance of the smallest cavity (FCC) and the point with maximum diameter of the biggest cavity (LHC). Furthermore, the system has to detect surface anomalies of only $10\mu\text{m}$. A 18MP camera with liquid lens, allowing it to focus between 20 to 25mm, will be used. In order to provide one full image of the inner surface, the pictures will be stitched together after the inspection. Thus, accuracy error of the end-effector position tangential to cavity surface should be not more than 1.2mm to obtain only 10% of overlapping error and not more than 1mm in direction perpendicular to the cavity surface, which would change the contained surface area in the image.

In Section 4.2.1 the initial model and geometry and the projection equation [59] to derive the equation of motion is described. Section 4.2.2 illustrates the collision checking procedure and Section 4.2.3 presents the applied kinematic criteria for this example. Then, initial states and results of the optimization are shown in Section 4.2.4 and 4.2.5, respectively.

4.2.1 Model

The surrogate model defining the design space was set up with one translational and four rotational joints with parallel rotation axes summing up to 5 DoF as shown in Figure 4.8. Table 4.1 lists the initial link lengths, where $l_{i,j}$ describes the lengths from joint i to joint j . These lengths were evaluated by launching an optimization problem, as described in Section 4.2.4, that returns feasible initial states. This initial set up was used as starting point for the actual optimization problem in (5.5).

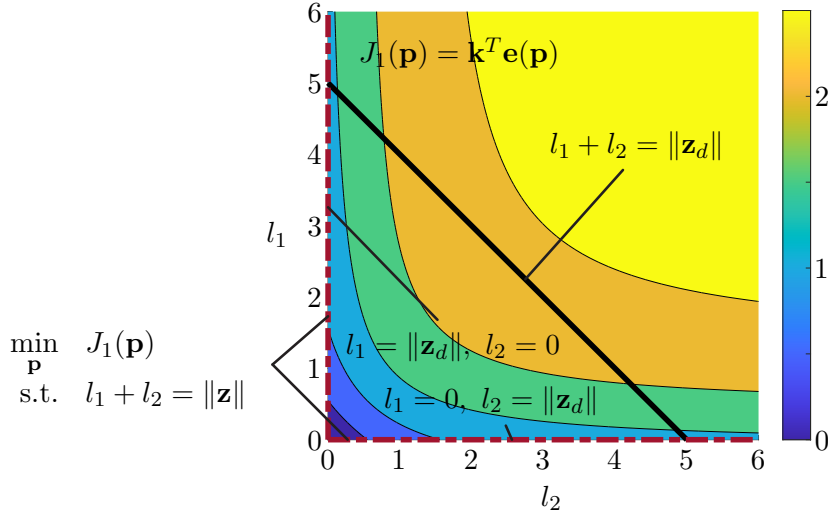


Figure 4.6: Contour plot of objective function $J_1(\mathbf{p}) = \mathbf{k}^T \mathbf{e}(\mathbf{p})$

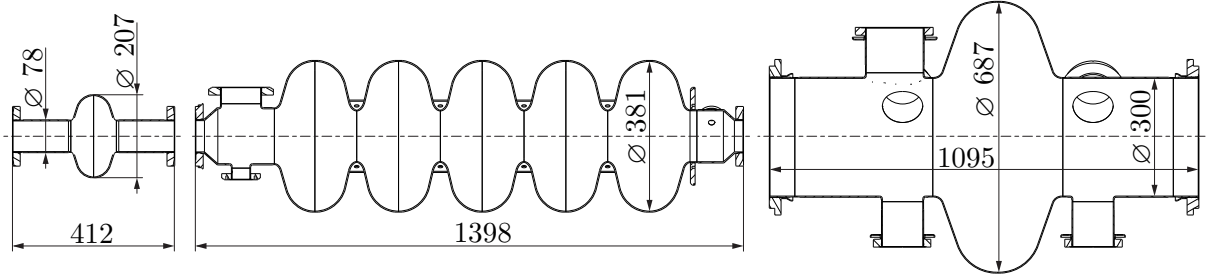


Figure 4.7: Cavity Types for FCC, LINAC and LHC (left to right, all units in mm)

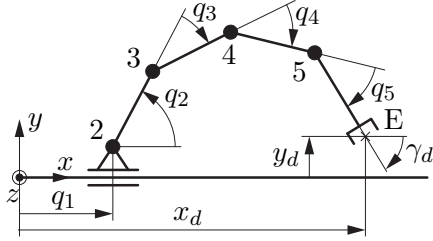


Figure 4.8: Design space

Table 4.1: Initial geometry

Length	[mm]
$l_{2,3}$	144.6
$l_{3,4}$	80.3
$l_{4,5}$	80.3
$l_{5,E}$	16.0

The generalized joint coordinates are set to

$$\mathbf{q} = [q_1, q_2, q_3, q_4, q_5]^T \in \mathbb{R}^{5 \times 1} \quad (4.17)$$

and the desired Cartesian position and orientation are

$$\mathbf{z}_d = [x_d, y_d, \gamma_d]^T \in \mathbb{R}^{3 \times 1}, \quad (4.18)$$

with the position x_d and y_d and the orientation around the z-axis γ_d always keeping the end-effector orientation perpendicular to the cavity surface. The optimization parameter was set to

$$\mathbf{x} = [\mathbf{q}_{1,1}^T, \mathbf{q}_{1,2}^T, \dots, \mathbf{q}_{1,\mu}^T, \mathbf{q}_{2,1}^T, \mathbf{q}_{2,2}^T, \dots, \mathbf{q}_{2,\mu}^T, \mathbf{q}_{n_{cav},1}^T, \mathbf{q}_{n_{cav},2}^T, \dots, \mathbf{q}_{n_{cav},\mu}^T]^T, \quad (4.19)$$

with the $n_{cav} = 3$ different cavities and μ different positions leading to $\mathbf{x} \in \mathbb{R}^{(\mu n_{cav}) \times 1}$. The parameter vector was defined as

$$\mathbf{p} = [l_{2,3} \quad l_{3,4} \quad l_{4,5} \quad l_{5,E}]^T. \quad (4.20)$$

The dynamic model was found by applying the projection equation (3.67) [59] and formulating it with the parameters \mathbf{p} as an argument, leading to the equation of motion for the robot

$$\mathbf{M}(\mathbf{p}, \mathbf{q}) \ddot{\mathbf{q}} + \mathbf{g}(\mathbf{p}, \mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}, \quad (4.21)$$

in the same form as already presented in (4.2).

4.2.2 Collision Avoidance

The collisions between robot and environment were calculated by discretizing the cavity surfaces and checking the minimal distances between robot links and points on the cavity surface. First all points on the cavity surface will be transformed into the body fixed coordinate frames for each robot link and then, to decrease the computation time, only for points that could possibly collide, the minimal distance will be calculated. Collisions are taken into account in function $\mathbf{c}(\mathbf{x}, \mathbf{p})$ in (5.5). Self collisions between robot links were not considered, since it was assumed that this is mechanically impossible.

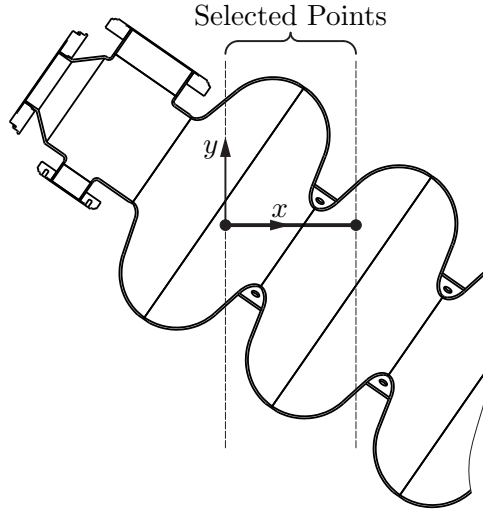


Figure 4.9: Collision detection for one robot link

4.2.3 Kinematic and Dynamic Performance Criteria

It is crucial for the cavity inspection robot to provide a very stable base for the camera that is being used for the inspection process, since the errors that should be detected in the surface can be only micro fractures. Thus, the term J_2 of (4.11) was set up to optimize the error propagation through the mechanical structure in a certain direction of interest. Error propagation describes how certain errors caused by e.g. the elasticity of the gears, backlash or control are

being forwarded to the end-effector. This can be quantified using the directional kinematic manipulability [74]

$$w_j = \sum_{i=1}^3 \left| \mathbf{n}_j^T \mathbf{u}_{j,i} \sigma_{j,i} \right| \quad (4.22)$$

with the unit vector \mathbf{n}_j representing the direction of interest (perpendicular to the cavity surface) and the major and minor axes of the manipulability ellipsoid $\sigma_{j,i} \mathbf{u}_{j,i}$ obtained from the singular value decomposition of the geometric Jacobian

$$\mathbf{J}_j(\mathbf{q}_j, \mathbf{p}) = \left[\left(\frac{\partial \mathbf{v}_E}{\partial \dot{\mathbf{q}}} \right)^T \left(\frac{\partial \boldsymbol{\omega}_E}{\partial \dot{\mathbf{q}}} \right)^T \right]^T, \quad (4.23)$$

with the linear and angular end-effector velocities \mathbf{v}_E and $\boldsymbol{\omega}_E$, respectively. Looking at the mapping from joint to Cartesian space via the Jacobian and replacing the small changes in joint angles $\Delta \mathbf{q}$ with an error \mathbf{e} , the error in Cartesian space is

$$\Delta \mathbf{z}_e = \mathbf{J}(\mathbf{q}) \mathbf{e}. \quad (4.24)$$

This means that if a robot is in a singularity like the 2-link arm in Figure 4.2(b), the error propagation in direction of the link is 0 and thus, the accuracy only depends on manufacturing tolerances of the mechanical parts of the robot. As a result the optimization algorithm will try to find configurations for which the accuracy is less dependent on the quality of gears or control.

Finally, the directional kinematic manipulability measure can be written in vector form according to (4.11) as

$$\boldsymbol{\Gamma}(\mathbf{x}, \mathbf{p}) = \left[w_1, w_2, \dots, w_\mu \right]^T, \quad (4.25)$$

with the weighting matrix $\mathbf{K} \in \mathbb{R}^{\mu \times \mu}$.

4.2.4 Initial Configuration

The initial states like joint angles and link lengths heavily influence the performance of optimization algorithms. It is important to provide feasible (in terms of the given constraints) initial states as a starting point for the optimization solvers. Starting points could be generated by either an inverse kinematics method or as it was done here, by running the optimization with the objective function (4.11) set to $J(\mathbf{x}, \mathbf{p}) = 0$. Thus, the initial states will be feasible with respect to all constraints, but non-optimal. The initial configurations and link lengths used as a starting point for the optimization solvers are shown for the three cavities in Figures 4.10, 4.11 and 4.12.

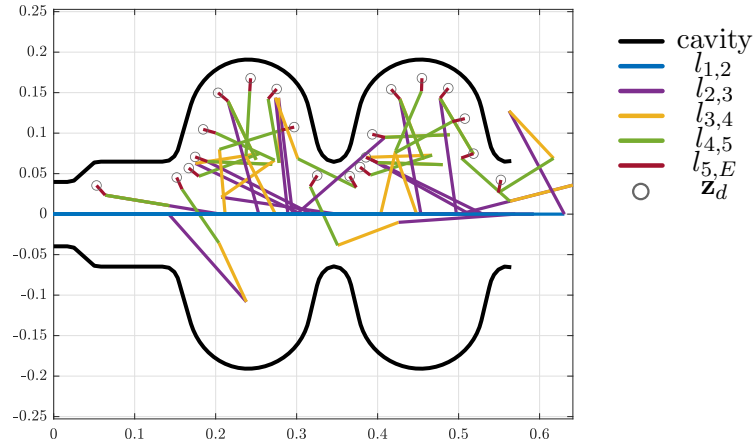


Figure 4.10: Initial states for the LINAC cavity (units in m)

The black curves represent the cavities and the colored lines represent the robot links. The gray circles indicate the desired end-effector position (x_d, y_d) from vector $\mathbf{z}_d = \begin{bmatrix} x_d & y_d & \gamma_d \end{bmatrix}^T$.

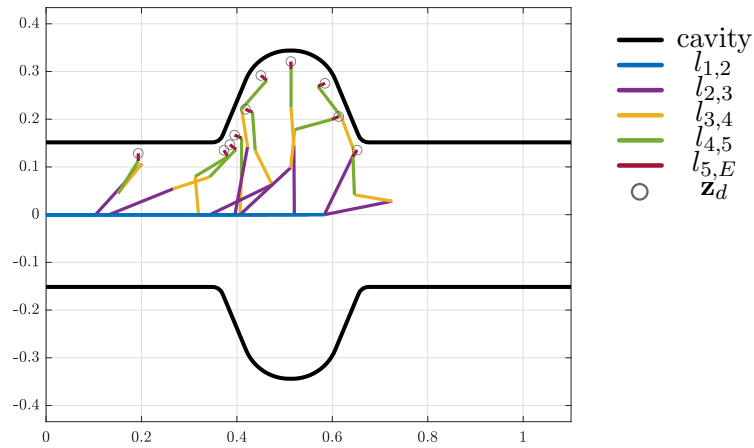


Figure 4.11: Initial states for the LHC cavity (units in m)

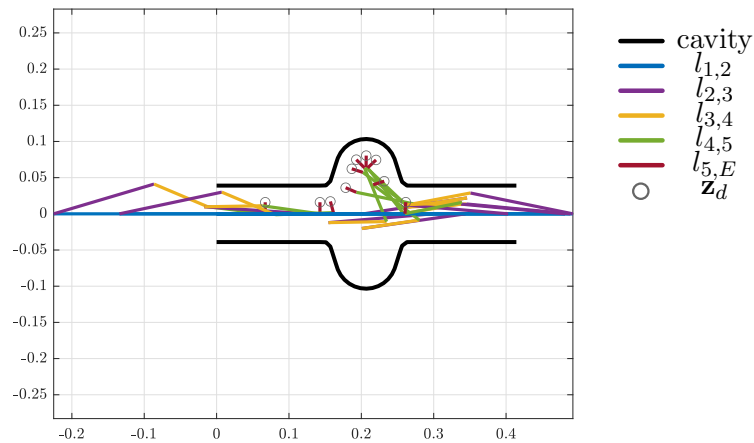


Figure 4.12: Initial states for the FCC cavity (units in m)

4.2.5 Optimization Results

The optimization departed from starting points defined for every desired end-effector position and cavity. The starting points were found by an optimization (described in Section 4.2.4) and include an initial topology, geometry as well as configuration of the arm. Matlab's *fmincon* function was used as a local optimization solver, in this case applying the interior-point algorithm [75]. Furthermore the *GlobalSearch* or *MultiStart* method can be applied to solve the global optimization problem [76]. In a comparison with evolutionary/genetic algorithms, the *GlobalSearch* and *MultiStart* methods lead to better results.

In Section 4.2.5 the design optimization is launched using only the LINAC environment for demonstration purposes and in Section 4.2.5 the robotic arm is optimized to operate in all three cavities.

LINAC Cavity

Here the design optimization is done only for the LINAC cavity in order to demonstrate the behaviour of the algorithm with a simpler, hence more intuitive example. The initial states match the ones presented in Figure 4.10 and the corresponding optimized design is visualized in Figure 4.13.

As mentioned in Section 4.2.1 the desired Cartesian position and orientation $\mathbf{z} \in \mathbb{R}^3$ is of dimension 3 and thus a mechanical structure with exactly 3 DoF is able to reach all desired end-effector positions \mathbf{z} in an environment without obstacles or other constraints. As shown in Figure 4.13 the restriction caused by the LINAC cavity allow a 3 DoF robotic arm to reach all position. Thus, the optimization reduced the tentative topology by 2 DoF, which is visible in Figure 4.14, but more clear in Table 4.2 where the lengths set to zero correspond to the removed DoF.

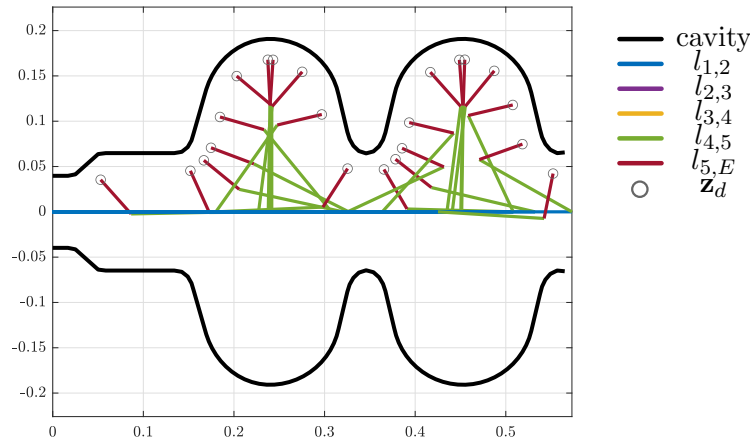


Figure 4.13: Optimized design for the LINAC cavity (units in m)

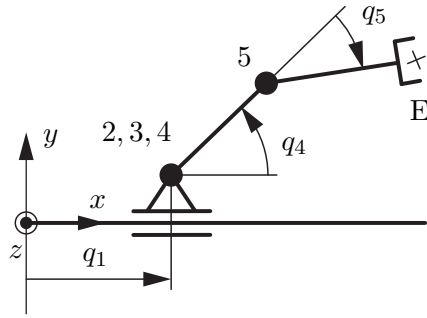


Figure 4.14: Optimized topology - LINAC

Table 4.2: Optimized geometry - LINAC

Length	[mm]
$l_{1,2}$	q_1
$l_{2,3}$	0
$l_{3,4}$	0
$l_{4,5}$	117.3
$l_{5,E}$	50.5

Furthermore, the length of the robotic arm has been minimized as it becomes clear when looking at the furthest point and observing that all links lie on a straight line from joint 2 to the desired end-effector position and this line is perpendicular to the x-axis. In Figure 4.14 the schematic drawing of the optimal topology is shown with some joints coinciding with others and thus illustrating the reduction of DoF.

Full System

The design optimization for the full system was performed by considering all three cavities as collision objects in the non-linear inequality constraints of the optimization problem. Again, the results of the optimization are shown in Figures 4.15, 4.16 and 4.17 for LINAC, LHC and FCC respectively, where the colored lines represent the robot links, the black curves indicate the surface of the cavities and the gray circles show the desired end-effector position. The resulting topology is shown in Figure 4.18 and the corresponding link lengths are listed in Table 4.3.

As it is clearly visible in Table 4.3 at least one link length was set to zero, which means that our requirement for the tentative topology of starting the design optimization with at least one DoF higher than the expected optimal solution holds and thus an optimized design for the robotic arm was found.

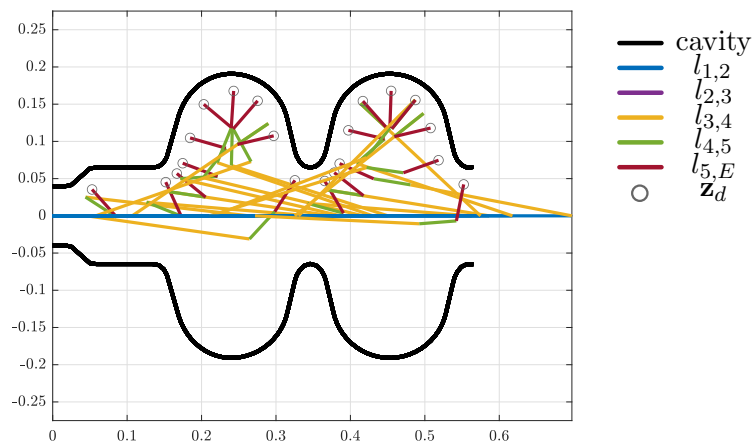


Figure 4.15: Optimized design for the LINAC cavity (units in m)

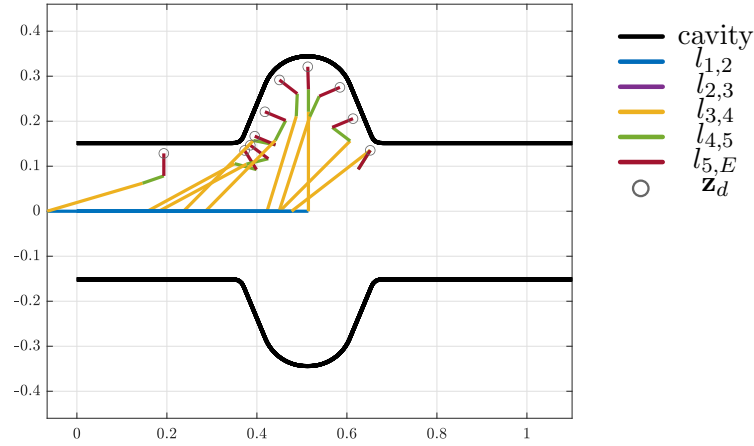


Figure 4.16: Optimized design for the LHC cavity (units in m)

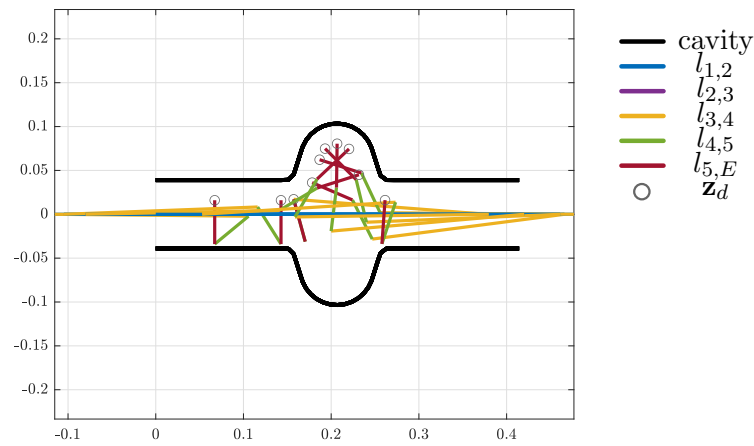


Figure 4.17: Optimized design for the FCC cavity (units in m)

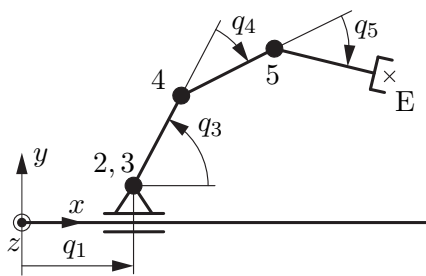


Figure 4.18: Optimized topology

Table 4.3: Optimized geometry

Length	[mm]
$l_{1,2}$	q_1
$l_{2,3}$	0
$l_{3,4}$	221.0
$l_{4,5}$	51.0
$l_{5,E}$	50.0

4.3 Application: FCC Manipulator

A detailed report on the design optimization of the FCC Manipulator (a robotic system for CERN's Future Circular Collider) was already published in [77]. Here, only a brief summary of the assumptions and results will be presented to demonstrate the algorithms capabilities.

The Future Circular Collider - FCC is designed to unlock observations in higher energy ranges than it is possible now with current accelerator machines at CERN [78]. This particle accelerator will be able to generate a center of mass energy of 100 TeV and will have a circumference of 100 km, see [16] and [9]. The 2020 Update of the European Strategy for Particle Physics listed the further investigation of the FCC as one of three main priorities and thus launched a TDR. One of the studies, launched within the TDR, concerns the automation of maintenance, inspection and emergency handling along the 100 km long FCC tunnel. The automation of these tasks plays a significant role for downtime, reliability and safety of particle accelerators and decreases the radiation exposure of workers.

The tasks such a automated system has to handle and the environment it will operate in are well defined, but no restrictions on the actual design of the manipulator in terms of topology and geometry are given. Thus, the presented algorithm was applied to optimize the tentative design of the FCC robot. In Section 4.3.1 the model and some assumptions are described. In Section 4.3.2 the applied objective function in terms of kinematic and dynamic criteria will be analyzed and finally some results are shown in Section 4.3.3.

4.3.1 Manipulator & Environment

The process of defining the surrogate model as shown in Figure 4.19 is described in detail in [77]. Here it should just be mentioned that joints 1 and 2 are translational joints, 3-6 and 7-10 form two planar mechanism in order to fold the arm. Joints 11-13 represent a robotic wrist and thus can be decoupled in point 12, which simplifies the optimization problem with respect to the end-effector orientation. The planar chains allow for minimizing the link lengths between joints with parallel axes, and thus possibly to eliminate joints connected by links with zero length, hence eventually reduce the DOF. The initial link lengths in Table 4.4 have been found by the same means as described in Section 4.2.4.

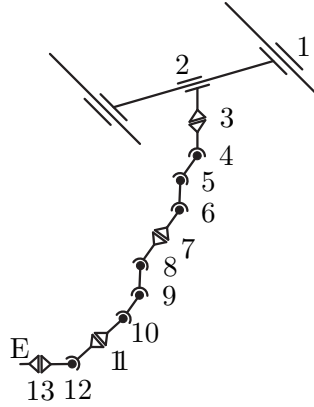


Figure 4.19: Design space

Table 4.4: Initial geometry

Length	[mm]
$l_{1,2}$	q_2
$l_{2,4}$	288
$l_{4,5}$	500
$l_{5,6}$	500
$l_{6,8}$	400
$l_{8,9}$	400
$l_{9,10}$	400
$l_{10,12}$	200
$l_{12,E}$	100

The environment of the FCC tunnel cross-sections and the robot links were approximated by convex geometric primitives, here specifically Matlab's AlphaShapes [79], which can be passed to a function to calculate the minimal distance between two convex geometric primitives. The approximation of the FCC environment with cylinders and boxes is indicated by red, dashed lines in Figures 4.20 and 4.21 for the FCC-ee and FCC-hh machine, respectively. The manipulator should be able to reach all points of interest (I - V), which represent the most diverse remote maintenance tasks at the current Large Hadron Collider - LHC, see [35], [36], [37] and [34].

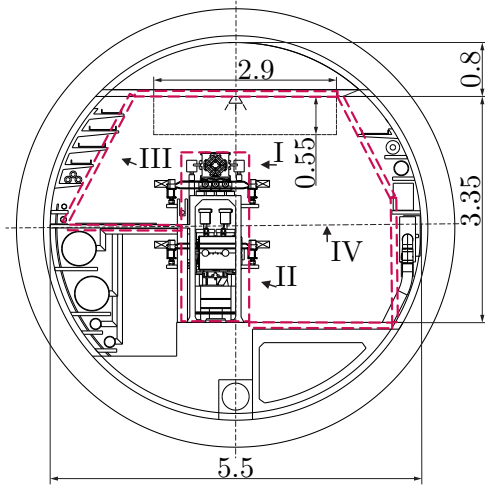


Figure 4.20: FCC-ee Cross-Section Lay-out

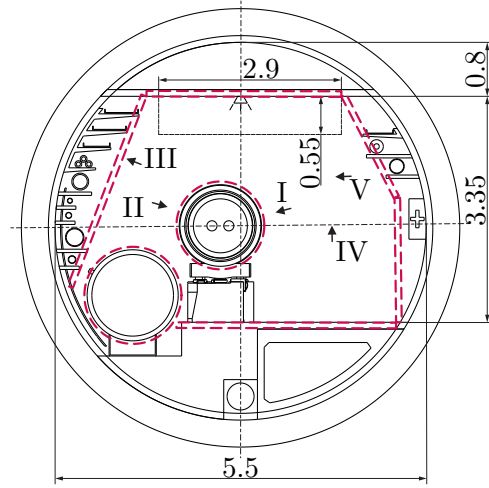


Figure 4.21: FCC-hh Cross-Section Lay-out

4.3.2 Kinematic and Dynamic Performance Criteria

A main objective for the design of the manipulator was to reduce the motor torques since the workspace shown in Figures 4.20 and 4.21 requires a relatively long robotic arm, compared to the desired payload and weight of the robot. Thus, the dynamic measure applied in term J_2 in the objective function (4.11) were the motor torques of each joint. A dynamic robot model in the form (4.21) was used and included in the objective function (4.11) with

$$\Gamma(\mathbf{x}, \mathbf{p}) = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{p}) \quad (4.26)$$

and the weighting martix $\mathbf{K} \in \mathbb{R}^{n \times n}$.

4.3.3 Optimization Results

The final results of the design optimization showed a reduced topology by two DoF. This is shown in Figure 4.24 by the coinciding joints 5/6 and 8/9, which corresponds to the lengths $l_{5,6} = l_{8,9} = 0$ in Table 4.5 of optimized geometry. In Figures 4.22 and 4.23 the optimization results are visualized using the collision objects (Alpha Shapes) for the two different accelerator machines FCC-ee and FCC-hh.

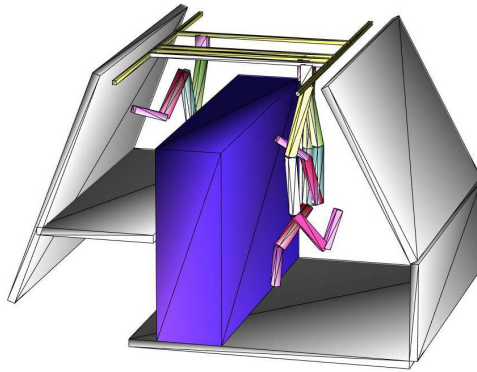


Figure 4.22: Optimization results FCC-ee

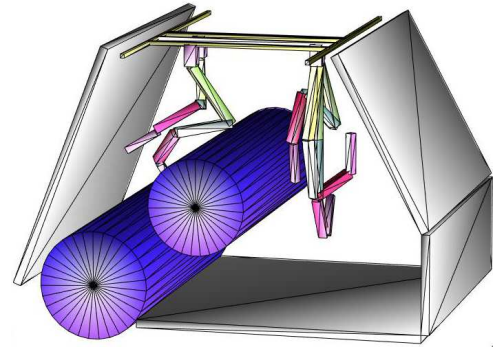


Figure 4.23: Optimization results FCC-hh

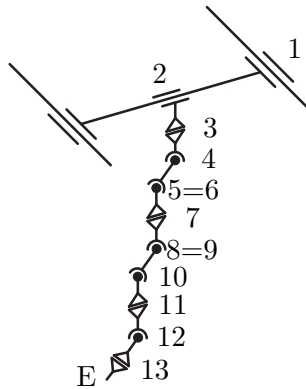


Figure 4.24: Optimized topology

Table 4.5: Optimized geometry

Length	[mm]
$l_{1,2}$	q_2
$l_{2,4}$	288
$l_{4,5}$	927
$l_{5,6}$	0
$l_{6,8}$	754
$l_{8,9}$	0
$l_{9,10}$	635
$l_{10,12}$	518
$l_{12,E}$	100

4.4 Conclusion

A method was presented that allows for optimization of the topology and geometric parameters of a wide variety of robotic manipulators. The algorithm uses well known and readily available optimization methods without any additional, complex frameworks. The simplicity of the presented method is one of the main advantages, as no inverse kinematics are computed and for the case without additional dynamic performance criteria, no dynamic model is needed. Thus, this method can be performed providing only the forward kinematics and (if necessary) a collision detection. This allows for very quick prototyping and one does not have to rely on either complex frameworks or a "educated guess" for a new manipulator design, but instead can quickly produce quantified results based on deterministic performance criteria. As demonstrated with examples in Section 4.2 and 4.3, the algorithm lead to good results for both the cavity inspection arm and the FCC manipulator.

The choice of additional kinematic and dynamic criteria needs to be done carefully in order to not generate ill conditioned optimization problems or even contradictions with the requirements. The weighting matrices in (4.11) have major impact on the final results and can as well lead to

infeasible solution in certain cases. Finding guidelines for these optimization parameters based on the mechanical structure and environment would simplify this heuristic process. The current implementation only allows optimization for serial robotic arms with translational joints or joints with parallel rotation axes. An extension of the presented method to a more general use case with closed kinematic chains and other joint types will be subject to future work.

A Novel Inverse Kinematics Technique with Smooth Task Switching

ABSTRACT. This chapter presents an inverse kinematics method that combines two well-known Jacobian based methods, the task-priority framework and an optimization-based approach, such that tracking and optimization tasks can be executed simultaneously. The novelty of the proposed algorithm lies in the ability to smoothly switch between different tasks and thus to allow for a seamless and safe transition during robot operation. This has shown to improve the efficiency and user experience, especially during tele-operated interventions in complex environments.

Chapter Content

5.1	Related Work	82
5.2	Smooth Task Switching Algorithm	83
5.3	Stability Analysis	85
5.4	Test Environment	87
5.5	Results	89
5.6	Conclusion	95

Related Publications

[80]	Gamper H., Rodrigo-Pérez L., et al.	A Redundant Inverse Kinematics with Smooth Task Switching - RAL
[26]	Gamper H., Forkel D., et al.	Adaptive Radiation Survey using an Autonomous Robot executing LiDAR Scans in the Large Hadron Collider - ISRR

The teleoperation control of redundant robots requires dedicated inverse kinematic resolutions, that support the operators during operations in order to increase the efficiency of the task execution and at the same time decrease the psychological stress of the operators. As an example, the robotic fleet at CERN is used for remote interventions and maintenance tasks throughout the entire CERN complex. Many robotic interventions are unplanned and still require human perception and cognition to assess the situation and act accordingly, hence many operations are conducted using teleoperated robots. With the increasing complexity of interventions, the deployed manipulators also increase in dexterity, leading to highly redundant topologies. The desired support features of the inverse kinematics algorithm, besides the tracking of a desired trajectory for the end effector (EE), were requested by CERN robot operators after difficult and complex interventions. The requirements can be assigned to either tracking or optimization tasks, in the following referred to as \mathbf{z} and τ , respectively.

Examples of requested features that resemble tracking tasks \mathbf{z} are the tracking of a desired trajectory under the constraint of one or more locked joints, taking joint limits into account or avoiding collisions with the environment. Examples for optimization tasks τ are the tracking of a EE trajectory while minimizing the error propagation throughout the manipulator in certain directions, the generation of cyclic trajectories or positioning the robot in a configuration that minimizes the motor torques.

Operators also requested to control a mobile robot by either moving only the base or only the robotic arm, thus splitting the interventions in prepositioning and fine manipulation sequences. Additionally, it was important for our operators to seamlessly switch between these tasks and not being forced to stop the movement between the tasks. This requires a seamless task switching using continuous switching profiles.

The proposed solution combines two widely used techniques for solving the inverse kinematics of redundant robots, the optimization-based approach [47] and the task prioritization that relies on the augmented Jacobian [43] in a closed-loop inverse kinematics (CLIK) tracking problem. These techniques have been introduced in Sections 3.4.2 - 3.4.5. It further facilitates smooth task switching and adjustment of the agility of joints. The algorithm can be used for online resolution of the inverse kinematics but is of course applicable offline or to preplanned trajectories.

A discussion on related work is presented in Section 5.1. In Section 5.2 the proposed algorithm is presented and the stability proof is shown in Section 5.3. In Section 5.4 the materials, methods and the testing environment is discussed. The experimental results validate what was claimed in the theoretical part and are presented in Section 5.5 and conclusions are given in Section 5.6.

5.1 Related Work

Computationally efficient and robust inverse kinematics algorithms have been an active field of research for decades and is still being investigated by researches until today. Extensive publications on reviews of existing methods (analytic, Jacobian based, Newton based, quadratic programming, data driven, statistical and heuristic) have been published in [58], [81], [82] and [47]. Analytic approaches are very limited in terms of degree of freedom ($\text{DoF} \leq 6$), heuristic approaches do not allow to include secondary objectives and data-driven, quadratic programming or statistical methods are more complex and computationally heavy, which is problematic in

real-time applications. Especially for redundant kinematics the Jacobian based methods allow to implement practical techniques of redundancy resolution and reviews on closed-loop inverse kinematics techniques [58] have shown that the damped least-squares pseudo-inverse is still one of the most simple, computationally efficient and robust approaches. Therefore, the presented inverse kinematics algorithm was based on the Jacobian method in a closed-loop, damped, least-squares pseudo-inverse form.

Resolving the redundancy of robotic structures in useful ways, has been an active field of research on its own. Capable tools like the augmented or extended Jacobian [43], quadratic programming frameworks [42], the task priority [83] and set-based task priority [45] framework are taking additional objectives or measures into account, to exploit redundancy. However, none of these techniques allow for dynamic dis- or enabling of the additional objectives, which is especially useful during tele-operated interventions in complex environments. The proposed algorithm in Section 5.2 presents a simple, yet capable and practical way to toggle additional objectives or in other words, introduces smooth task switching.

5.2 Smooth Task Switching Algorithm

Denote with $\mathbf{q}(t) \in \mathbb{R}^\delta$ the vector of time dependent joint coordinates (generalized coordinates), where δ is the DoF of the robotic manipulator and $r \leq \delta$ the dimension of the task space. The tracking error is introduced as

$$\boldsymbol{\theta}(t, \mathbf{q}(t)) = \mathbf{z}_{ref}(t) - \mathbf{z}(\mathbf{q}(t)) \in \mathbb{R}^r, \quad (5.1)$$

where \mathbf{z}_{ref} and \mathbf{z} describe the reference and the actual tracking task. If the task describes the EE pose, it should be noticed that \mathbf{z} are local coordinates, which is valid and singularity-free for small tracking errors. In full generality, the error is defined on the Lie group $SE(3)$, representing EE pose by means of homogenous transformations. As discussed in [39, Sec. 3.7.3 - Angle and axis], the above formulation is a valid approximation for small errors, and will be used in the following. The time derivative of the tracking error is

$$\dot{\boldsymbol{\theta}} = \frac{\partial \boldsymbol{\theta}}{\partial t} + \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \dot{\mathbf{z}}_{ref} - \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}, \quad (5.2)$$

with the Jacobian $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{r \times \delta}$. To ensure the stability of the tracking task an error feedback is applied, designing the asymptotically stable error dynamics

$$\mathbf{0} = \dot{\boldsymbol{\theta}} + \mathbf{K}\boldsymbol{\theta}, \quad (5.3)$$

for the diagonal matrix $\mathbf{K} > 0$. Substituting (5.3) into (5.2) leads to the well-known CLIK architecture for the inverse problem

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \left(\dot{\mathbf{z}}_{ref} + \mathbf{K}\boldsymbol{\theta} \right). \quad (5.4)$$

In order to deal with non-square matrices, avoid singularities and set a priority for the movement of the different joints, the weighted and damped, least-squares pseudo inverse \mathbf{J}^\dagger , as a result of

the optimization problem, see Section 3.4.2, will be applied for the Jacobian inverse, leading to

$$\dot{\mathbf{q}} = \underbrace{\mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T + \alpha(\mathbf{q}) \mathbf{I})^{-1}}_{\mathbf{J}^\dagger} (\dot{\mathbf{z}}_{ref} + \mathbf{K} \boldsymbol{\theta}), \quad (5.5)$$

with the weighing matrix

$$\mathbf{W} = \text{diag}\{\sigma_{W1}, \sigma_{W2}, \dots, \sigma_{W\delta}\}. \quad (5.6)$$

The definition of the weights $\boldsymbol{\sigma}_W = [\sigma_{W1}, \sigma_{W2}, \dots, \sigma_{W\delta}]^T$ are described in Section 5.2.1. The choice of the damping factor $\alpha(\mathbf{q})$ is shown in Section 3.4.6. In the case of a redundant kinematic system, an artificial potential $\nabla \tau^T \dot{\mathbf{q}}$ can be added to the cost function of the optimization problem, leading to

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger (\dot{\mathbf{z}}_{ref} + \mathbf{K} \boldsymbol{\theta}) + \underbrace{(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})}_{\mathbf{N}} \mathbf{W}^{-1} \frac{\partial \tau}{\partial \mathbf{q}}, \quad (5.7)$$

with the null-space projector \mathbf{N} . In order to assign a priority to the tracking tasks, the augmented Jacobian and its corresponding null-space projector

$$\mathbf{J}_{1,2} = \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{bmatrix} \in \mathbb{R}^{n \times \delta} \text{ and } \mathbf{N}_{1,2} = (\mathbf{I} - \mathbf{J}_{1,2}^\dagger \mathbf{J}_{1,2}) \in \mathbb{R}^{\delta \times \delta}, \quad (5.8)$$

with $n \leq \delta$ have to be introduced [47]. This allows to incorporate the two tracking tasks with the corresponding references $\mathbf{z}_{ref,1}$, $\mathbf{z}_{ref,2}$ and actual tasks $\mathbf{z}_1, \mathbf{z}_2$. The solution for the velocity simultaneously minimizing τ is

$$\dot{\mathbf{q}} = \mathbf{J}_1^\dagger (\dot{\mathbf{z}}_{ref,1} + \mathbf{K}_1 \boldsymbol{\theta}_1) + \underbrace{\sigma_\theta(t) (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1)}_{\mathbf{N}_1} \mathbf{J}_2^\dagger (\dot{\mathbf{z}}_{ref,2} + \mathbf{K}_2 \boldsymbol{\theta}_2) + \underbrace{\sigma_\tau(t) (\mathbf{I} - \mathbf{J}_{1,2}^\dagger \mathbf{J}_{1,2})}_{\mathbf{N}_{1,2}} \mathbf{W}^{-1} \frac{\partial \tau}{\partial \mathbf{q}}, \quad (5.9)$$

with the switching parameters $\sigma_\theta(t)$ and $\sigma_\tau(t)$, as defined in Section 5.2.1, for smooth enabling/disabling of the tracking and optimization tasks.

5.2.1 Switching Parameter

To provide smooth switching of the tasks and weights, any trajectory from differentiability class C^3 can be applied. Here, a base profile $\sigma_B(t) \in \mathbb{R}$ will be defined on $t \in \mathbb{R}$ such that $\sigma_B(t) \in [0, 1]$ and $\sigma_B(t)$ is strictly increasing. Applying a \sin^2 function satisfies all requirements and can be piece-wise defined in acceleration as

$$\ddot{\sigma}_B(t) = \begin{cases} 0 & t < 0 \\ a_{max} \sin(\frac{2\pi}{t_S} t)^2 & t < \frac{t_S}{2} \\ -a_{max} \sin(\frac{2\pi}{t_S} t)^2 & t < t_S \\ 0 & \text{else,} \end{cases} \quad (5.10)$$

with the maximum acceleration $a_{max} > 0$ and the switching time t_S . This leads to the final switching profile

$$\sigma(t) = \sigma_S + \sigma_B(t)(\sigma_E - \sigma_S), \quad (5.11)$$

with the start and end weights $\sigma_S, \sigma_E \in \mathbb{R}$, which will be used for each $\sigma_{W_1}(t) \dots \sigma_{W_\delta}(t)$ in (5.6) and for $\sigma_\theta(t)$ and $\sigma_\tau(t)$ in (5.9). This enables the user to switch tasks on and off by only providing the duration for the switching process.

5.3 Stability Analysis

The stability proofs for both the regulation and the optimization task have been shown in several papers before, see [47], [46], [83]. Here, the stability proof for the combined system and especially taking into account the newly introduced switching parameters will be demonstrated. In Section 5.3.1, it will be demonstrated that the error dynamics of the regulation tasks are asymptotically stable and in Section 5.3.2 an upper bound for the gain \mathbf{K} for discrete systems is given.

5.3.1 Stability of the Tracking Tasks

First, equation (5.9) can be simplified, by omitting the third term resembling the optimization task. This step is justified because of the following two points: First, the optimization task always needs to be set up as the lowest priority task and thus does not influence the error dynamics of the higher priority tasks. Second, the optimization task has no reference and thus it is impossible to define the error dynamics for this term. However, to guarantee a feasible optimization task it needs to be linearly independent (compatible) [47, Sec. 10.5.3] from higher priority tasks, that is satisfying

$$\text{rank} \left(\mathbf{J}_{1,2}^\dagger \right) + \text{rank} \left(\frac{\partial \tau}{\partial \mathbf{q}} \right) = \text{rank} \left[\mathbf{J}_{1,2}^\dagger \quad \frac{\partial \tau}{\partial \mathbf{q}} \right]. \quad (5.12)$$

In order to prove the stability of the tracking tasks in (5.9) we rewrite the task errors in the form

$$\bar{\boldsymbol{\theta}} = \begin{bmatrix} \boldsymbol{\theta}_1^T & \boldsymbol{\theta}_2^T \end{bmatrix}^T \quad (5.13)$$

and the partial time derivatives of the error in the form

$$\frac{\partial \bar{\boldsymbol{\theta}}}{\partial t} = \begin{bmatrix} \frac{\partial \boldsymbol{\theta}_1}{\partial t}^T & \frac{\partial \boldsymbol{\theta}_2}{\partial t}^T \end{bmatrix}^T = \dot{\mathbf{z}}_{ref}. \quad (5.14)$$

With this notation, we can now rewrite (5.9) in the form

$$\dot{\bar{\boldsymbol{\theta}}} = - \underbrace{\begin{bmatrix} \mathbf{K}_1 & \mathbf{0} \\ \mathbf{J}_2 \mathbf{J}_1^\dagger \mathbf{K}_1 & \sigma_\theta(t) \mathbf{J}_2 \mathbf{N}_1 \mathbf{J}_2^\dagger \mathbf{K}_2 \end{bmatrix}}_{\mathbf{A}(t)} \bar{\boldsymbol{\theta}} + \underbrace{\begin{bmatrix} \mathbf{I} - \mathbf{J}_1 \mathbf{J}_1^\dagger & \mathbf{0} \\ -\mathbf{J}_2 \mathbf{J}_1^\dagger & \sigma_\theta(t) (\mathbf{I} - \mathbf{J}_2 \mathbf{N}_1 \mathbf{J}_2^\dagger) \end{bmatrix}}_{\mathbf{B}(t)} \dot{\mathbf{z}}_{ref} \quad (5.15)$$

yielding a non-autonomous error system. The lower right term in matrix $\mathbf{B}(t)$ disappears if and only if the task \mathbf{J}_2 is fully represented in the nullspace \mathbf{N}_1 . This will be assumed so that

$$\mathbf{J}_2 \mathbf{N}_1 \mathbf{J}_2^\dagger = \mathbf{J}_2 \mathbf{J}_2^\dagger, \quad (5.16)$$

leading to $\mathbf{N}_1 = \mathbf{I}$ in case \mathbf{J}_2 has full rank [46]. Then, the diagonal elements of matrix $\mathbf{B}(t)$ disappear because \mathbf{J}_1^\dagger and \mathbf{J}_2^\dagger are the right inverse of \mathbf{J}_1 and \mathbf{J}_2 , respectively. The off diagonal elements in $\mathbf{A}(t)$ and $\mathbf{B}(t)$ disappear due to the annihilation condition $\mathbf{J}_2\mathbf{J}_1^\dagger = \mathbf{0}$ [83].

In order to test the error system for stability, we define a scalar Lyapunov candidate function

$$V(t, \bar{\boldsymbol{\theta}}) = \frac{1}{2} \bar{\boldsymbol{\theta}}^T \bar{\boldsymbol{\theta}} \quad (5.17)$$

which is bounded from below by definition and its time derivative is

$$\dot{V}(t, \bar{\boldsymbol{\theta}}) = \bar{\boldsymbol{\theta}}^T \dot{\bar{\boldsymbol{\theta}}} = -\bar{\boldsymbol{\theta}}^T \mathbf{A}(t) \bar{\boldsymbol{\theta}}. \quad (5.18)$$

The upper left submatrix of $\mathbf{A}(t)$ needs to be positive semidefinite and thus satisfy $\mathbf{K}_1 > 0$. The lower right submatrix is positive definite if the gains satisfy $\mathbf{K}_2 > 0$, $\sigma_\theta(t) > 0$, $\sigma_W(t) > 0$ and if the first two tasks are linearly independent

$$\text{rank}(\mathbf{J}_1^\dagger) + \text{rank}(\mathbf{J}_2^\dagger) = \text{rank} \begin{bmatrix} \mathbf{J}_1^\dagger & \mathbf{J}_2^\dagger \end{bmatrix}. \quad (5.19)$$

and not annihilating [83], hence

$$\mathbf{J}_1 \mathbf{J}_2^\dagger \neq \mathbf{0}. \quad (5.20)$$

In case $\sigma = 0$ the task is disabled, will not be tracked and thus is obviously not asymptotically stable. The proof can then be similarly shown with only the first tracking task. If the two sub-matrices of the main diagonal of $\mathbf{A}(t)$ are positive definite, then the matrix $\mathbf{A}(t)$ itself is positive definite and thus,

$$\dot{V}(t, \bar{\boldsymbol{\theta}}) < 0. \quad (5.21)$$

Since (5.15) resembles a non-autonomous system we employ Barbalat's Lemma to test for stability. Thus, we need to show that $\dot{V}(t, \bar{\boldsymbol{\theta}})$ is uniformly continuous, which can be derived from the sufficient condition of

$$\ddot{V}(t, \bar{\boldsymbol{\theta}}) = -\bar{\boldsymbol{\theta}}^T \left(\mathbf{A}(t)^T \mathbf{A}(t) + \dot{\mathbf{A}}(t) + \mathbf{A}(t)^2 \right) \bar{\boldsymbol{\theta}}, \quad (5.22)$$

being bounded. This condition holds if $\sigma(t)$, $\mathbf{A}(t)$ and $\dot{\mathbf{A}}(t)$ are bounded, which are satisfied as $\sigma(t) \in [0, 1]$ and $\sigma(t)$ is of C^3 , while $\mathbf{A}(t)$ and $\dot{\mathbf{A}}(t)$ are bounded, because \mathbf{J}_1 , \mathbf{J}_2 consist of analytic functions which are of C^ω and the pseudo inverses are damped. Thus, $\lim_{t \rightarrow \infty} \dot{V}(t, \bar{\boldsymbol{\theta}}) = 0$ holds and therefor also

$$\lim_{t \rightarrow \infty} \bar{\boldsymbol{\theta}} = \mathbf{0}. \quad (5.23)$$

5.3.2 Discrete Systems

An extensive discussion on the stability for discrete systems is presented in [84]. For our simple case, some bounds for the gains $\mathbf{K} = \text{diag}([k_1, k_2, \dots, k_n])$ of the discrete system corresponding to (5.3) can be found via the steps shown below. The discrete system

$$\frac{\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}}{\Delta t} = \mathbf{K} \boldsymbol{\theta}, \quad (5.24)$$

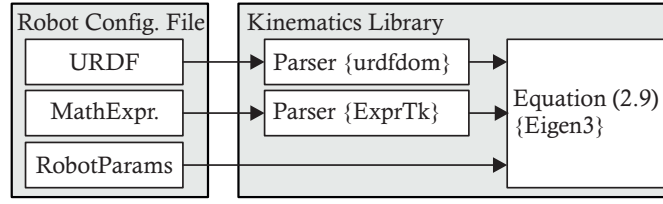


Figure 5.1: The architecture of the Kinematics Library. The robot configuration file holds the geometric description of the mechanical system in either the URDF format or hard coded mathematical expressions, that will be parsed using the *urdfdom* or *ExprTk* library. The block that implements (5.9) uses *Eigen3*. The configuration file also holds all other parameters that are relevant for the motion controller of the CRF.

which can be rewritten as a discrete dynamic system

$$\boldsymbol{\theta}_{k+1} = (\mathbf{I} - \Delta t \mathbf{K}) \boldsymbol{\theta}_k, \quad (5.25)$$

that is asymptotically stable if the condition

$$\|\text{eig}\{\mathbf{I} - \Delta t \mathbf{K}\}\| < 1 \quad (5.26)$$

holds, which leads to lower and upper bounds for the gain \mathbf{K} in the form

$$0 < k_i < \frac{2}{\Delta t}. \quad (5.27)$$

5.4 Test Environment

The presented algorithm is part of a new robot motion controller for the CERN Robotic Framework (CRF) [11]. The kinematics library was developed in C++ and the only external libraries used, are *Eigen3* [85], *urdfdom* [86] and *ExprTk* [87], see Figure 5.1. The code architecture was designed, taking into account expandability for future implementations and improvements, while still ensuring robustness with proper memory handling and a low computational cost during the main robot control loop. The inverse kinematics algorithm was deployed on a real-time system in a main control loop with a loop time of 2 ms. The computational complexity of the algorithm can not be stated in general, since it depends on the specific tasks one chooses to consider. However, in Section 5.5 the computation times for the most complex scenario are presented.

All tests were done using a Ubuntu system with a Real-Time Linux Preempt Kernel patch [88] launching the process with the highest priority in the scheduler. It is worth mentioning that the system features an 11th Gen Intel^R CoreTM i5-1145G7E processor. The algorithm was tested in three scenarios, exploiting different tasks on different robotic systems, described in the following Sections 5.4.1 - 5.4.3.

5.4.1 Train Inspection Monorail - 9 DoF

This test was performed using the Train Inspection Monorail (TIM) [14] [26] in the Large Hadron Collider (LHC) main tunnel at CERN. TIM is a robotic system driving along the LHC tunnel on a single rail that is mounted on the ceiling. The first two joints of the manipulator are rotational

and the third one creates a translational movement. On the third, linear axis, a 6 DoF Kinova Jaco2 arm is mounted resulting in a total of 9 DoF, see Figure 5.2. This robotic arm is used for



Figure 5.2: The Train Inspection Monorail (TIM) during testing in the LHC tunnel.

remote inspections and maintenance activities, like sensor testing and calibration or radiation measurements. For this test, the EE of the robot will track a six-dimensional reference $\mathbf{z}_{ref,1}$, leading to a redundancy with degree three. The tracking of the EE will be encoded as the highest priority task \mathbf{z}_1 in the first term in (5.9). The lower priority tracking task \mathbf{z}_2 will be used to control joint q_5 , while the robot is tracking the desired EE position. The optimization task τ will drive the robot away from the limits of joints q_1 , q_2 , q_3 and q_4 and will be enabled throughout the entire test. The test will show the behavior of the robotic system during the switching process, that enables/disables the tracking task \mathbf{z}_2 (in other words, blocks or unblocks joint q_5) using the switching parameter $\sigma_\theta(t)$.

5.4.2 Universal Robot UR10e - 6 DoF

The second test will be executed with the 6 DoF robot UR10e from Universal Robot. In order to demonstrate the capabilities of the presented algorithm, the redundancy is created by a reduction of the task space dimensions. Specifically, the EE will track a three-dimensional reference $\mathbf{z}_{ref,1}$ (only including the desired position), such that the robot resembles a redundant system with degree three. The tracking of the EE will be encoded as the highest priority task \mathbf{z}_1 in the first term in (5.9). The optimization task τ will be used to block certain joints, while the robot is tracking the desired EE position. The test will also show the behavior of the robotic system during the switching process, that enables/disables the optimization task (in other words, blocks or unblocks certain joints) using the switching parameter $\sigma_\tau(t)$.

5.4.3 Mobile Platform with Arm - 9 DoF

In this scenario, the robotic system is a mobile base with four mecanum wheels and a 6 DoF Kinova Jaco 2 arm mounted on top, resulting in a system with 9 velocity DoF and 10 joints to be controlled, see Figure 5.3. The EE will track a six-dimensional reference, which will be encoded

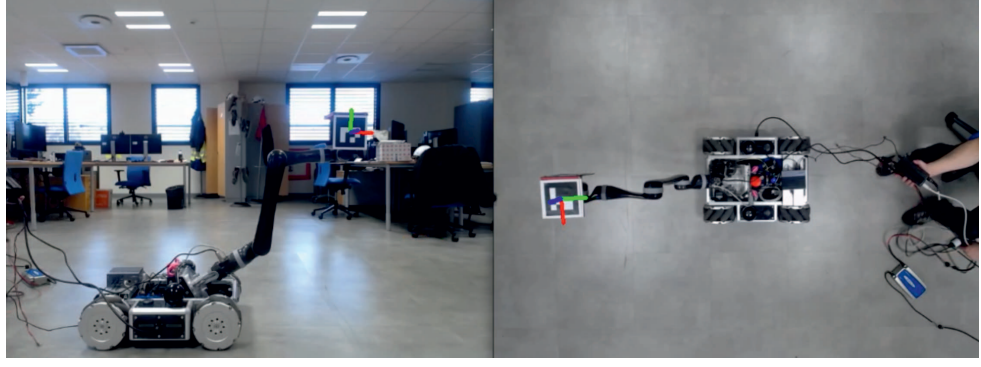


Figure 5.3: Omnidirectional base with a Kinova Jaco2 with spherical wrist

as the highest priority task \mathbf{z}_1 in the first term in (5.9). In this test, no additional tracking or optimization tasks are considered, but instead, the smooth switching of the weighted mobility of the single joints will be demonstrated. This is especially useful when a mobile platform with an attached arm has to be prepositioned to a point of interest (that lies outside the workspace of the attached arm) using the wheels of the robot and once this point is reached, will handle the fine manipulation of an object, using the joints of the robotic arm only. The test will show the behavior of the robotic system while switching the mobility of the joints using the parameter $\sigma_W(t)$. The measurement of the actual task \mathbf{z}_1 was done, using two cameras and two ArUco markers [89], [90], see Figure 5.3. The marker planes were held parallel to the camera sensor plane throughout the entire test. This was done because the precision of the cameras is higher for measurements parallel to the camera sensor as opposed to measurements perpendicular to the camera sensor. The combination of the data from both cameras then allowed to reconstruct the actual task \mathbf{z}_1 .

5.5 Results

The acquired data during the tests, described in the previous Sections 5.4.1 - 5.4.3, are presented and discussed in the following Sections 5.5.1 - 5.5.3 for the Train Inspection Monorail, the Universal Robot UR10e and the mobile platform with arm, respectively. The average computation time of the inverse kinematics algorithm in test scenario 3 was 82.5 μs , with a minimum of 41.3 μs and a maximum of 819.6 μs . As expected, test scenario 3 showed the highest computation time, since it implemented the most complex setup.

5.5.1 Train Inspection Monorail - 9 DoF

The desired EE pose $\mathbf{z}_{ref,1} \in \mathbb{R}^6$ for this test was set constant and represents the reference to the highest priority tracking task. The tracking error $\boldsymbol{\theta}_1$ is shown in Figure 5.4a. The lower priority task will block joint q_5 at a desired position $\mathbf{z}_{ref,2}$ if $\sigma_\theta = 1$ holds. The tracking error $\boldsymbol{\theta}_2$ is shown Figure 5.4b. Additionally, the optimization task

$$\tau = \sum_i -p_i \left(e^{c_i(q_i - q_{max,i})} + e^{-c_i(q_i - q_{min,i})} \right), \quad (5.28)$$

with $i = 1 \dots 4$, will drive the robot away from the limits $q_{max,i}$ and $q_{min,i}$ of joints q_i , if $\sigma_\tau = 1$ holds. The exponential terms in (5.28) allow the robot to freely move within the joint limits and generate a strong, repelling force close to the joint limits. The effect of τ can be tuned with the gains p_i and c_i . The gradient of the optimization task is shown in Figure 5.4e and the switching parameters σ_θ and σ_τ are shown in Figure 5.4f. The parameter σ_τ will be set to a constant value of 1 and thus the optimization task will be active during the entire test in order to avoid collisions. The tracking task, and thus the parameter σ_θ , will be switched on and off twice, blocking the joint q_5 , first at $\mathbf{z}_{ref,2}(t) = \frac{\pi}{2}$ for $t \leq 44$ and then at $\mathbf{z}_{ref,2}(t) = \frac{\pi}{4}$ for $t > 44$. The dashed lines in Figure 5.4 indicate the start and end of the switching sequences.

In Figure 5.4a it is clearly visible that the EE follows the task reference $\mathbf{z}_{ref,1}$ with minor errors, which mainly occur when the secondary task is being enabled. The tracking error θ_2 , see Figure 5.4b, is higher during the switching sequence as long as $\sigma_\theta < 1$, but disappears as soon as $\sigma_\theta = 1$. This is also visible in the plot of the joint angles (result of the inverse kinematics), see Figure 5.4c, for joint q_5 which is first blocked at $\frac{\pi}{2}$ rad and then at $\frac{\pi}{4}$ rad and in Figure 5.4d the joint velocities are shown. Since the robot starts in a relatively compact configuration, the optimization task is driving the robot away from the joint limits at the beginning of the sequence, as it is clearly visible in the plot of the gradient of the optimization task, see Figure 5.4e.

5.5.2 Universal Robot UR10e - 6 DoF

The desired EE trajectory in position $\mathbf{z}_{ref,1} \in \mathbb{R}^3$ is shown in Figure 5.5a. The trajectory resembles a linear movement in the three dimensions x, y and z. In this test, $\mathbf{z}_{ref,1}$ represents the reference for the highest priority tracking task. The tracking error θ_1 is shown in Figure 5.5b. The results of the redundancy resolution in joint space, are shown in Figure 5.5c and 5.5d with joint angles \mathbf{q} and velocities $\dot{\mathbf{q}}$, respectively. The gradient of the optimization task

$$\frac{\partial \tau}{\partial \mathbf{q}} = \mathbf{C}(\mathbf{q}_{des} - \mathbf{q}) \quad (5.29)$$

is shown in Figure 5.5e. $\mathbf{q}_{des} \in \mathbb{R}^{DoF}$ represents the vector with desired joint angles and \mathbf{C} is a diagonal matrix with weighing gains in the main diagonal corresponding to the robot joints. Figure 5.5f shows the switching parameter σ_τ . The red and blue areas visualize the time during which the optimization task (blocking of joints) is being enabled/disabled ($0 < \sigma_\tau \leq 1$). In between the dashed lines in either of the red or blue areas, the optimization task is enabled ($\sigma_\tau = 1$). During the red area the optimization task with the desired joint angles $\mathbf{q}_{des,1} = [\sim, \sim, -\frac{\pi}{2}, \sim, 2.5, \sim]$ will be enabled and during the blue area the desired joint angles are $\mathbf{q}_{des,2} = [\sim, \sim, \sim, \sim, \sim, 0]$. The corresponding joints with \sim entries are free and will not be blocked.

In the red area, it can be observed that at the beginning of the switching process, the joints q_3 and q_5 start to move towards the desired angles as stated in $\mathbf{q}_{des,1}$. Then, in between the dashed lines, when the optimization task is enabled, the joints q_3 and q_5 hold their desired position. At the time the switching parameter starts disabling the task, the joints are free and start to move again according to the highest priority task reference $\mathbf{z}_{ref,1}$ with minimal joint velocities.

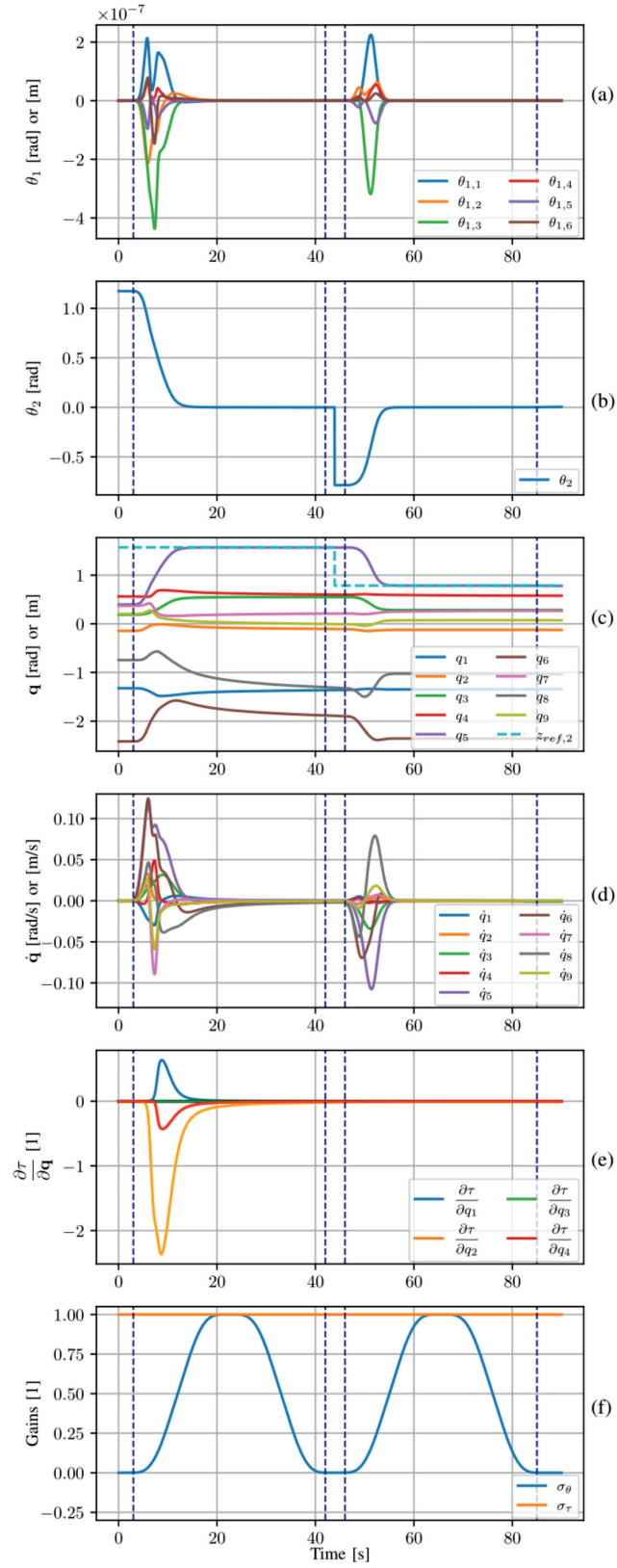


Figure 5.4: Test data recorded with TIM in the LHC tunnel. Showing tracking error $\theta_{1,i}$ for the i -th dimension (a), tracking error θ_2 (b), the inverse kinematics solution in joint angles \mathbf{q} (c) and joint velocities $\dot{\mathbf{q}}$ (d), the gradient of the optimization task τ (e) and the switching parameters σ_θ and σ_τ (f).

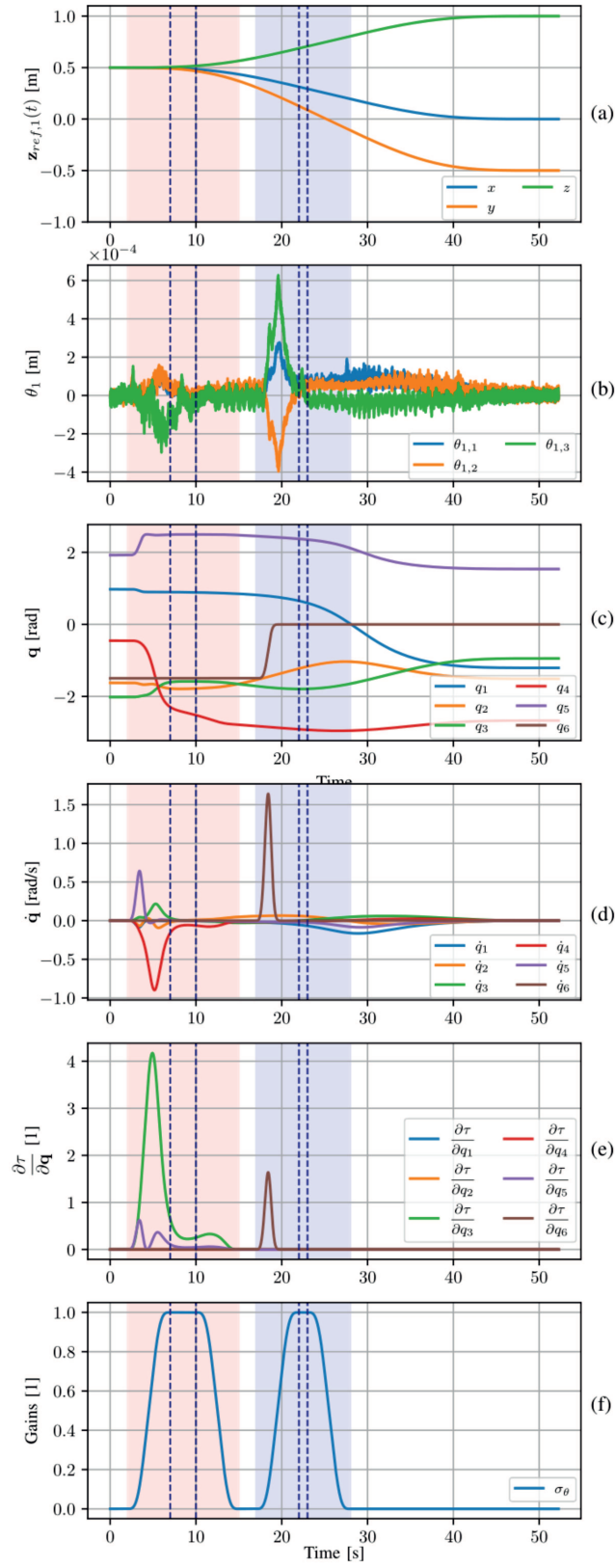


Figure 5.5: Test data recorded with UR10e. Showing the desired task reference $\mathbf{z}_{ref,1}$ (a), tracking error θ_1 (b), the inverse kinematics solution in joint angles \mathbf{q} (c) and joint velocities $\dot{\mathbf{q}}$ (d), the gradient of the optimization task τ (e) and the switching parameters σ_θ and σ_τ (f).

In the blue area, only joint q_6 will be blocked according to $\mathbf{q}_{des,2}$. At the beginning it is clearly visible that the joint is being pulled towards the zero line and is constant during the time, the task τ is enabled. Other than during the first sequence (red area), now the joint q_6 stays constant even after the joint has been unblocked again. This is first of all, because the desired EE trajectory $\mathbf{z}_{ref,1}$ is of dimension three and does not include the orientations and second, because the redundancy resolution is done with minimal joint velocities.

The task error θ_1 is shown in Figure 5.5b. It is clearly visible that the task switching is causing spikes in the position error. This is because the maximum accelerations of the single joints were not high enough to support a sufficiently fast repositioning of the robot into the new configuration. This effect can be avoided by increasing the switching time of the switching parameter σ_τ .

5.5.3 Mobile Platform with Arm - 9 DoF

The desired EE trajectory $\mathbf{z}_{ref,1} \in \mathbb{R}^6$ resembles a circle in the x-y plane with a constant orientation of the EE and lies at a constant height in the z-direction. This trajectory represents the reference for the highest priority tracking task. Figure 5.6a shows the tracking error θ_1 , calculated with the actual EE position \mathbf{z}_1 from the data of two ArUco markers. The noise in these measurements is due to some false positive detections in the testing environment. A video of this experiment is available via the link provided in [80].

The results of the redundancy resolution in joint space are shown in Figures 5.6b and 5.6c for the joint angles of the mobile base $q_1 \dots q_4$ (wheels) and the manipulator $q_5 \dots q_{10}$. In Figures 5.6d and 5.6e the joint velocities of the mobile base and the manipulator are shown. The diagonal elements of matrix \mathbf{W} in (5.5) are shown in Figure 5.6f and the blue area indicates the time during which the switching of the weights is taking place.

At the start of the trajectory the diagonal elements of \mathbf{W} defined by σ_W , see (5.11), are set to $\sigma_{W1..W4} = 1000$ and $\sigma_{W5..W10} = 1$, thus the desired EE trajectory will be followed by the 6 DoF manipulator exclusively. This is clearly visible in the plot of the wheel velocities which are all 0 before the blue area. In the blue area the weights are being switched to $\sigma_{W1..W4} = 1$ and $\sigma_{W5..W10} = 1000$. As shown in the plots of the joint velocities, Figure 5.6d and 5.6e, the wheels and the manipulator's joints are both moving during the switching process and after the blue area only the wheels are now creating the motion to follow the circular trajectory with the EE.

This feature increased the efficiency of operations during which a large workspace has to be covered and at the same time, fine manipulation tasks have to be performed. The base can then be used as a fast, less precise prepositioning system and the manipulator can later on be used for fine manipulation tasks while the wheels of the robot are blocked. All this can be done without stopping the movement or reconstructing objects in C++. At the time these tests were performed, no odometry has been implemented in the new motion controller yet and the inverse kinematics algorithm was launched without feedback regarding the actual position. Thus, all the accumulated errors like jerky motions of the Kinova at the beginning and end of the trajectory, or uneven floors are visible in the recorded EE position in Figure 5.6a.

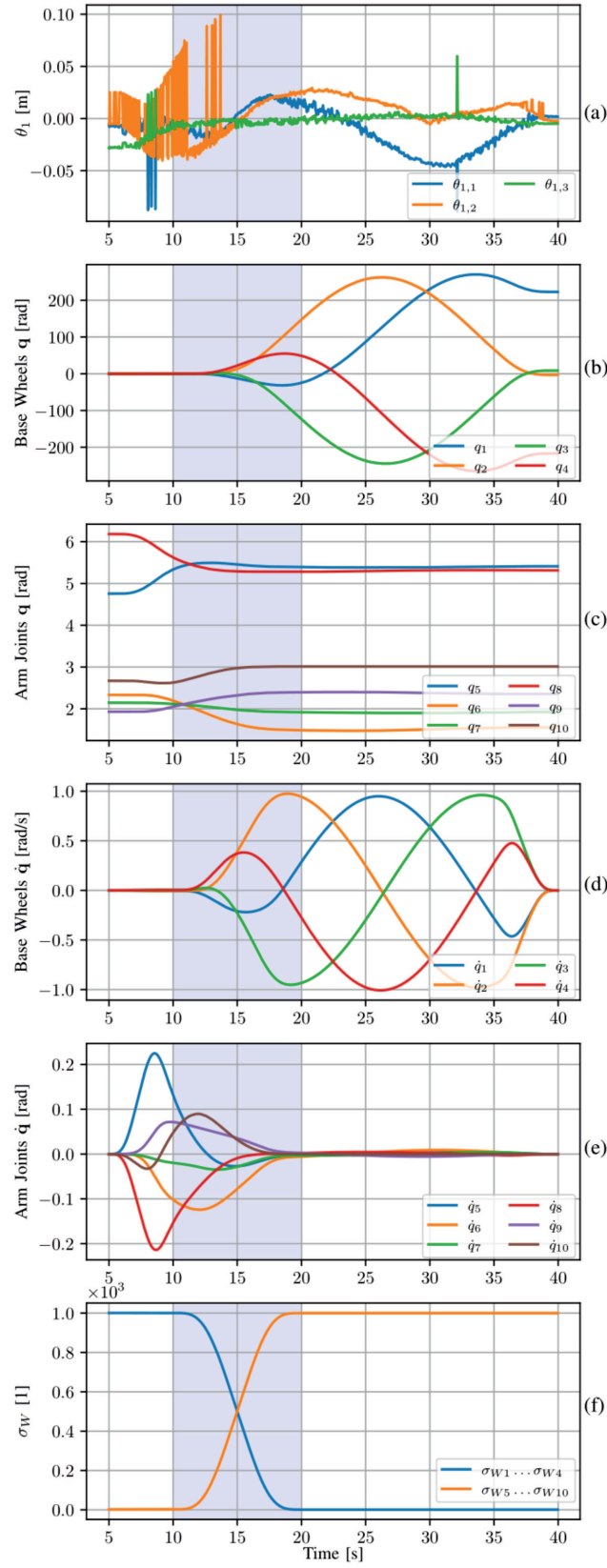


Figure 5.6: Test data recorded with the mobile platform with arm. Showing tracking error θ_1 (a), the inverse kinematics solution in joint angles \mathbf{q} (b) and (c) and joint velocities $\dot{\mathbf{q}}$ (d) and (e) and the weighing gains σ_{W_i} for the i -th dimension (f).

5.6 Conclusion

The presented inverse kinematic approach combines the augmented Jacobian approach with an optimization based approach in order to incorporate tracking and optimization tasks. Furthermore, a smooth adjustment of the weights in the damped, least squares pseudo-inverse was implemented to control mobile robotic platforms, either moving only the base or only the robotic arm. The presented algorithm seems very promising to increase the efficiency and reduce the stress on operators, though quantified bench-mark human-study tests are still missing. The test results of the three experiments with a 6 DoF, and two 9 DoF systems are as expected and showed stable behavior as desired. Nevertheless, there are some shortcomings with the presented algorithm: currently, only two tracking tasks can be taken into account. Disabling the second tracking task will result in this task not being tracked, but the nullspace will not be increased for the optimization task. In order to achieve this, the optimization task can be disabled as well and then the nullspace can be adjusted without problems. The maximum computation time for the most complex system (considering the presented hardware), lies below 1 ms, showing that the algorithm is well suited for real-time applications.

In summary, the presented algorithm is of great value for practical implementations and complex teleoperated interventions. Currently, efforts are being made to publish these developments as a standalone, open-source C++ library. Future work will focus on the formulation of optimization tasks that generate reliable cyclic trajectories and the computational complexity will be investigated in more detail for common use cases.

A Modular Motion Controller for the CERN Robotic Framework

ABSTRACT. This chapter provides a comprehensive introduction and overview of the CERN Robotic Framework (CRF), discussing its architecture, infrastructure, and continuous integration tool chain. It then presents the motion controller library, which has been successfully deployed across CERN's entire robotic fleet. This library is designed with a modular structure, employing well-defined interfaces to achieve both flexibility and robustness. The implementation of the novel inverse kinematics approach will be discussed in detail. Emphasizing the commitment to open-source principles, important details about the kinematics library, including its dependencies, interfaces, and practical usage will be discussed. Finally, the chapter concludes with a discussion on the achieved work and presents potential directions for future developments.

Chapter Content

6.1	The CERN Robotic Framework	99
6.2	The Motion Controller Module	105
6.3	Development of an Open-Source Kinematics Library	107
6.4	Showcase Examples . . .	111
6.5	Conclusion	112

The *Mechatronics, Robotics and Operations* (MRO) section began to perform remote interventions in 2015 with off-the-shelf robotic solutions like Teodor and Telex as presented in Section 2.2 and Figure 2.3(a) and 2.3(b). After numerous successful operations the demand for such interventions in the CERN complex increased continuously together with the complexity of the tasks. Especially the increased complexity of the tasks in terms of workspace, locomotion, specialized tools for manipulation or perception, led to increasing efforts in customized robotic solutions. At the beginning the focus was on customized mechanical designs for e.g. simple grippers, but soon required the integration and implementation of sensor and actuator interfaces on software level. In 2015 robotic frameworks like the *Robot Operating System* (ROS) were not capable of real-time applications and thus the decision was made to kick-off the development of a robotic framework mainly written in C++.

This framework, later called CRF, was designed to cover everything from low-level driver implementations over a middle-ware communication layer to applications like localization and user interfaces for teleoperation. One of the design goals was a modular architecture that will be easily extendable and allows for fast prototyping in case of emergency interventions that require adjustments to the robotic systems. Furthermore, it was essential to ensure real-time capability for safety critical tasks as well as a *radiation hard* software design.

However, during the early design phase of the CRF development, the integration of off-the-shelf solutions of robots, sensors, cameras, etc. and especially the middle-ware, for communication between these systems was prioritized, which was a necessary decision due to the time pressure for certain interventions in the CERN facilities. The consequences of this decision though, were clearly visible 4 years later in 2019, when this work was initiated: Little effort has been put into the development of a holistic concept for controlling customized robotic solutions that do not come with an interface/API from the manufacturer.

Therefore, only very simple control methods had been implemented that did not follow the modular design of the framework and thus, were unstable and rigid implementations. Especially there were not modules for robot kinematics, dynamics or control, which did not allow the MRO section at that time to control customized robot designs, which at the same time got increasingly important due to more and more challenging requests from stakeholders within CERN, that could not anymore be satisfied with off-the-shelf solutions.

After four years of remote interventions, robot operators have gained a considerable amount of experience during hundreds of interventions and often reported inefficient and tedious interventions due to a lack of flexibility in control options, especially for inverse kinematic resolutions. State of the art research showed that none of the existing inverse kinematic methods could account for the requests of the operators and it became clear that only a novel inverse kinematic tailored to their needs would be able to increase the efficiency and robustness of interventions.

The aforementioned lead to the decision to develop a holistic concept for the control of robotic systems following the principles of the CRF, the development of a modular motion controller. This furthermore requires the design of a novel kinematics library that accounts for the requests and experiences from robot operators.

In Section 6.1 a overview of the structure and scope of the CRF is presented, including the framework infrastructure and continuous integration toolchains in Sections 6.1.1 and 6.1.2. The motion controller module with the contained kinematics library will be discussed in Sections 6.2 and 6.3. In the following Section 6.4 some introductory examples using simple robotic

structures will be presented to showcase the capabilities and usage of the developed systems. Finally, conclusions and future developments will be discussed in Section 6.5.

6.1 The CERN Robotic Framework

The CRF is used to control the entire robotic fleet and all prototypes in the MRO section and it consists of modules that are grouped into categories like *Communication*, *Navigation* or *Actuators*. The modules follow well defined interfaces to ensure an easy expandability and can be enabled or disabled to only compile what is necessary for a specific robotic system.

Figure 6.1 provides a comprehensive overview of the architecture of the CRF and showcases the usage of the CRF for a specific example application. In the application scenario the FCC Robot, consisting of a gripper, a lidar, three PMSM motors and a Universal Robot, is being controlled in the LHC from the CERN Control Center (CCC) using the GUI.

The top rectangle in Figure 6.1 visualizes the components of the FCC Robot that need to be controlled using the CRF, representing the physical layer. The bottom rectangle shows the robotic intervention workspace in the CCC with the GUI on the left screen and terminal messages on the right screen, representing the application layer. In between those two rectangles the CRF manages the translation from a user input to an actual movement of the robot.

The middle rectangle in Figure 6.1, representing the CRF, contains smaller rectangles, which indicate the different modules contained in the CRF. For the sake of clarity, only an arbitrary subset of the modules of the CRF have been visualized here. The status of the single modules, for the given application example, is indicated by the small labels on the top left corner of the modules, with **I** or **O** indicating a module that is enabled or disabled, respectively.

Modules that are enabled are necessary to control the specific robotic system and are therefore compiled and deployed on the robots PC. For the showcase example it is necessary to enable the *Gripper* module, the *Laser* module to communicate with the Lidar, the *CANOpenMotor* module to communicate with the motor drivers via the CANopen protocol and subsequently being able to control the PMSM machines and the *UniversalRobot* module to control the UR10e.

Furthermore the *TrajectoryGenerator* is enabled to plan trajectories according to the user input from the *2DGUI*. The *CombinedRobot* module is used to control an arbitrary combination of an arbitrary amount of motors and robots, in this case forming the FCC Robot. The *Kinematics* module is taking care of the necessary kinematics calculations. This module builds on the theoretical developments from Chapter 5 and the implementation will be discussed in more detail in Section 6.3.

The *CANOpenMotor* and *UniversalRobot* are directly passed to the *CombinedRobot*, which itself is directly passed to the *MotionController*, together with the *TrajectoryGenerator* and the *Kinematics* module. The *MotionController* module is one of the new developments mentioned already in the introduction and will be presented in Section 6.2. The *Gripper*, *Laser*, *MotionController* and *2DGUI* do not communicate directly with each other, but via a *Communication* point module, which handles the messages and takes care of the multi thread processing and pipelines for communication.

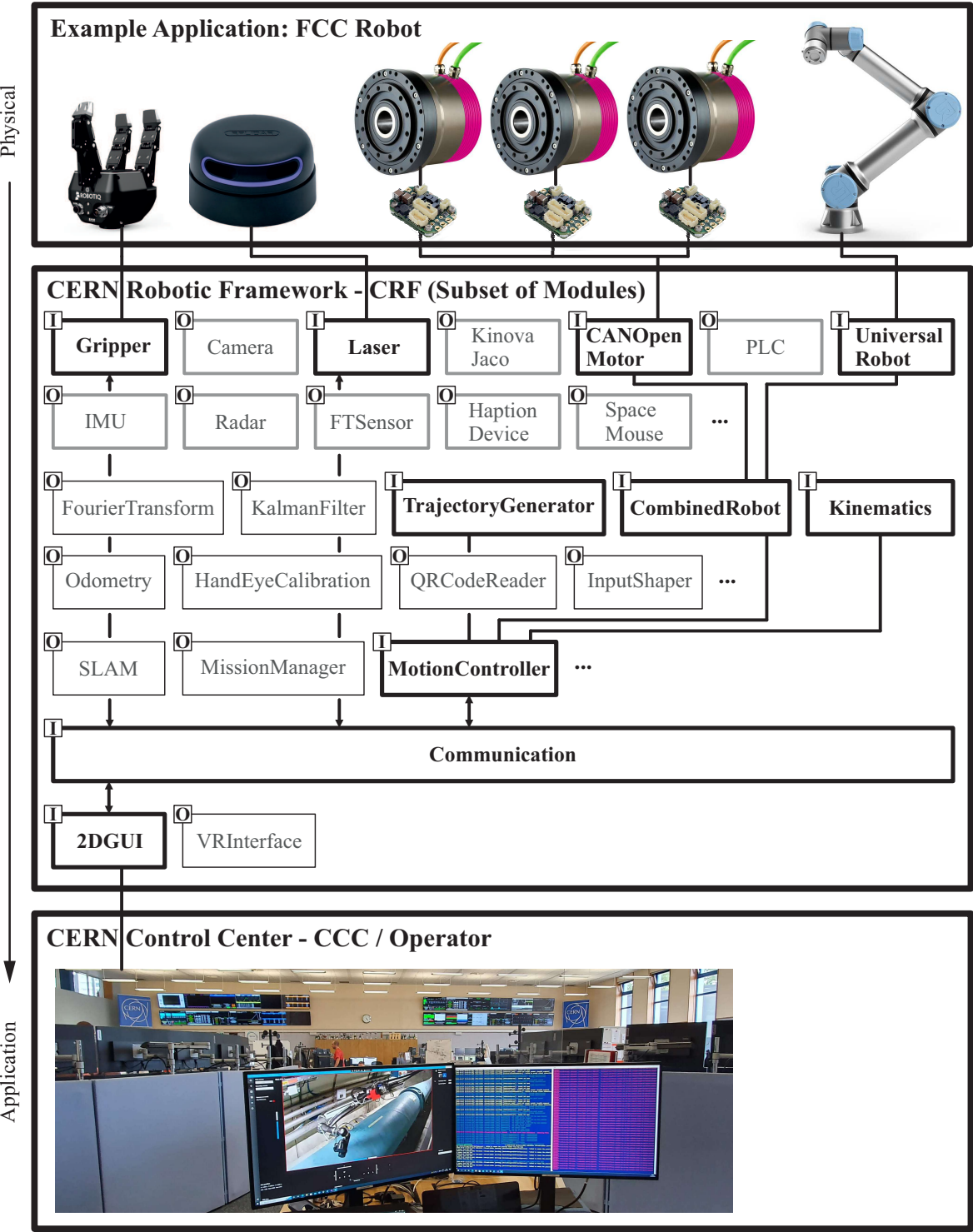


Figure 6.1: The CERN Robotic Framework - CRF. An example showcasing the CRF used to control the FCC Robot.

6.1.1 Framework Infrastructure

Over the years, the CRF became an extensive framework and is still continuously growing. The management of such big frameworks with many contributors improving or expanding the code at the same time becomes a challenge on its own. It is crucial to thoroughly keeping track of changes and save every version of the framework, as well as defining clear rules concerning the code style and testing procedures.

In order to account for the aforementioned points, the CRF heavily relies on common software management tools. The version control of the framework is realized with *git* [91] handling the contributions of roughly 30 developers in about 80 branches.

For build automation, packaging and installation the software packages *CMake* [92] and *Make* [93] are used. *CMake* is a cross-platform software that provides a user friendly way to generate the build files, in this case the *MakeFiles*, which will be used by *Make* to compile the code. These software packages also take care of locating dependencies (in form of internal and external libraries) and linking the libraries, which would otherwise be a very tedious process. Furthermore, *CMake* is used in the CRF to enable or disable the required modules for a specific project, as it is visualized by the **I** or **O** labels on the blocks in Figure 6.1. *Make* on the other hand is able to detect which bits of the code have changed and only compile the missing parts.

The CRF follows Google's coding style guides [94] and Google's *cpplint* [95] is used to check the conformity of the developed code with the best practices defined by Google. Whenever possible, raw pointers or direct memory allocation are avoided. However, every code has to undergo a check for memory leaks using *valgrind* [96].

Finally, new modules have to be thoroughly tested before being merged to the master branch of the framework. This ensures that the master branch is clean and always ready to be deployed to a robotic system without compilation errors or bugs, since this could impact active interventions in the e.g. CERN underground tunnels. These tests are performed using the extensive *GoogleTest* framework. This allows for efficient implementation of test scenarios and provides a method to mock objects on which the subject under testing (SUT) relies on. The mocking of objects is essential when the SUT e.g. communicates with hardware that is not available at the point of testing and thus could not pass any of the test scenarios.

Figure 6.2 provides a comprehensive overview of the directory structure of the CRF. The expanded folders locate the implementation of the Universal Robot as an example. All other modules follow the same directory structure.

The highest level folder *CERNRoboticFramework* contains the folders and files, which will briefly be discussed in the following. The folder *cmake* includes files with instructions for *CMake* on how to find dependencies and general parameters for its set up. The folder *libraries* contains external libraries and *modules* contains all code related to the robotic activities, organized in subfolders like *Actuators*, *Applications* or *Communication*. The folder *scripts* contains bash files for the installation of the CRF, a script to run the *cpplint* for code style checks and several scripts to set up hardware interfaces like CAN ports. The file *CMakeLists.txt* is the main file that *CMake* uses to generate the build instructions for *Make* and includes the information on which modules should be enabled for the specific project. Finally, a *README.md* provides general instructions on the usage of the CRF in the markdown [97] format.

As an example for the directory structure for the modules the *Actuator* module with the *Robot* implementation is expanded. The first folder on *Robot* level is *config* which contains all necessary information to control a specific robot e.g. the robot kinematics, software and hardware limits and communication information. In the next folder *include* *Robot* the header files can be found and *samples* contain minimal examples of the usage of the specific modules. The *src* and *tests* folder contain the source files and different test scenarios, respectively. Furthermore, on this level another *CMakeLists.txt* file can be found that contains information on dependencies for the specific module, which will be located and processed by the main *CMakeLists.txt* file when running CMake.

In the next folder level *UniversalRobot* the header files for the specific robot are stored, which in this case is essentially a wrapper for the Universal Robot API on one hand and the instructions on how to parse the configuration files from *config* on the other hand. The header file *IRobot.hpp* is a pure virtual class used to define an interface for the robot class, from which every robot implementation has to inherit from. Interface definitions are widely used to simplify and limit the amount of callable public functions for an implementation by enforcing certain predefined conventions on the developers. Finally, the file *RobotMock.hpp* is an implementation of the robot interface used during software testing, when the physical robot is not connected to PC which runs the tests.

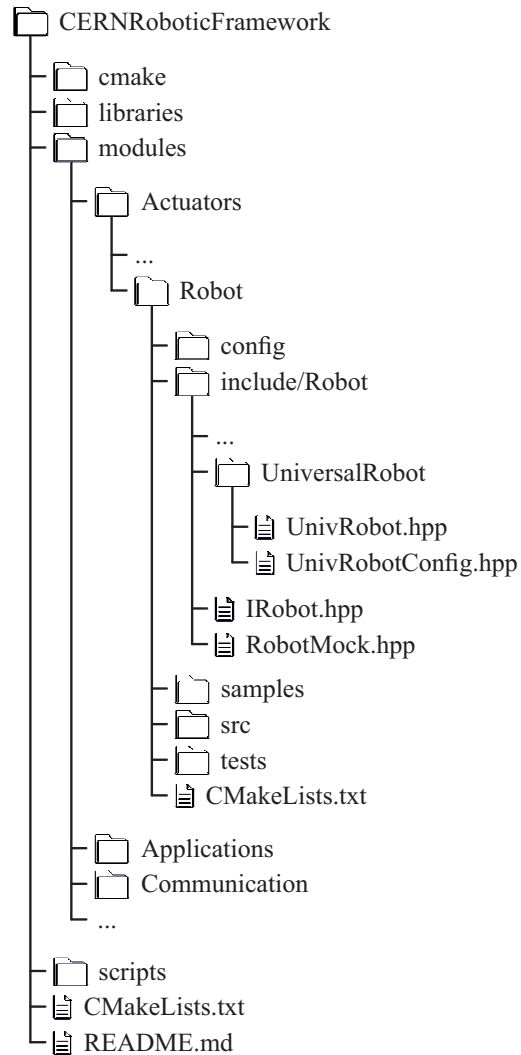


Figure 6.2: The directory structure of the CRF. The expanded directory locates the implementation of the Universal Robot.

6.1.2 Continuous Integration

The processes and checks e.g. code style checks, checks for memory leaks or unit tests, are strictly required before merging new code to the framework to ensure its availability. Furthermore, these tests have to be performed on a machine which is set up in the standard configuration to run the CRF. Performing these tests locally on the personal machines of the developers would be inefficient and unstable due to unavoidably different machine set ups. Thus, a remote server (here GitLab [98]) is used to automatically trigger these tests and checks after a developer pushes changes from his local machine to the remote branch they are working on. This process is usually referred to as Continuous Integration (CI) pipeline.

Figure 6.5 visualizes the work flow for a new implementation in the CRF. First, an issue will be created on the remote server side, which describes the the type of changes (e.g. new functionality,

bug fix, etc.) and other necessary details for the implementation. The issue will be assigned to a developer, who then creates a new branch from master on the remote side, which they will subsequently clone to their local machine, the developer PC. On their local machine the developer will then start to implement the changes and once they are done or want to save the current state of their work, they will push the code from the local machine back to remote branch.

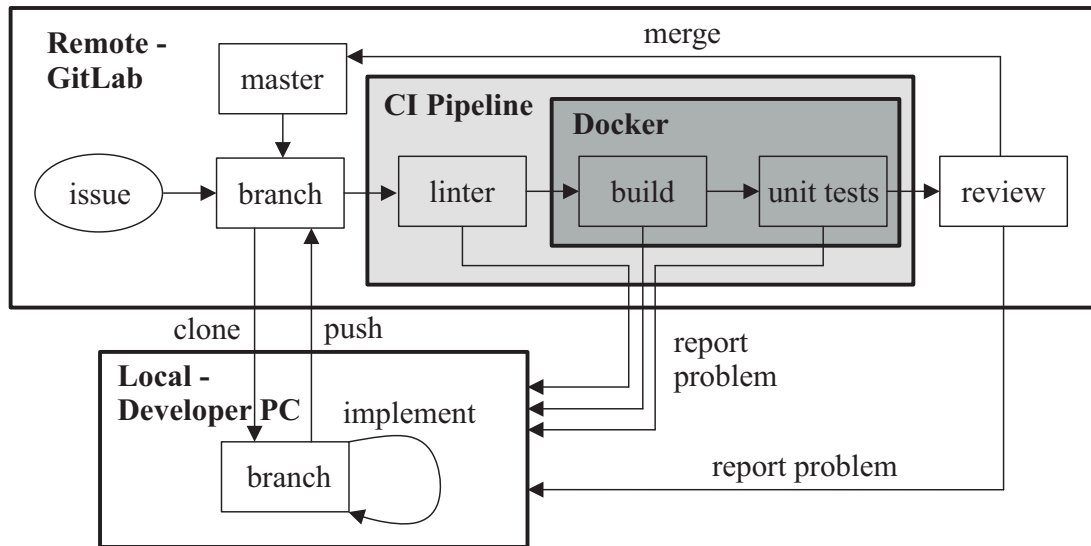


Figure 6.3: Visualization of the work flow for a new implementation in the CRF.

Whenever a push to the remote side is performed, the CI pipeline will be triggered, which will subsequently invoke the previously discussed checks and tests. First, the pipeline launches the linter to check the code style and in case problems are detected the pipeline fails, stops and reports the problems, which the developer has to fix and start over the process by pushing the code to the remote branch.

If the linter checks pass, the pipeline will invoke a virtual machine hosted by the software package Docker [99]. Docker allows to run images of a predefined machine set up in order to ensure that the build process and unit tests always run on the same system, which ensures reproducible reports. In case any problems are detected during the compilation (build process) or the unit tests it will be reported back to the developer, who will apply the fixes on the local machine and start over the process by pushing the code to the remote side. Here the importance of the mocks (e.g. RobotMock.hpp, see Figure 6.2) become very clear, since invoking the unit tests will run all tests in the framework, including tests for all robots, sensors, cameras, etc., which can obviously not all be connected to the virtual machine that runs the tests.

In case the CI pipeline passed successfully, a group of senior developers, that have been assigned to the specific issue, will be notified and perform a final review process of the submitted code. This can either result in a merge of the new code to the master branch of the CRF, which concludes the implementation process and closes the issue, or in case further problems or recommendation are reported, the developer again will implement the requested changes and start over the process by pushing to the remote branch.

6.1.3 Operating System

The development of a motion controller that directly controls the position of the motor of the robots needs to send smooth trajectories in fixed time steps, instead of relying on the controller of the manufacturer of the robotic system or motor driver. This new approach furthermore requires a real-time capable system taking care of the "low level" control of the robot motors.

The CRF builds on the Linux distribution Ubuntu and is currently tested on Ubuntu 20.04 and Ubuntu 22.04 [100]. Before the development of the motion controller module the control of robots and motors relied on the real-time communication of the built-in robot controllers or motor drivers and the CRF was not real-time capable.

In order to achieve real-time capability with Ubuntu 20.04 the standard Linux kernel was replaced with a version of the kernel which is compatible with the real-time patch, both available at kernel.org/pub/linux/kernel and kernel.org/pub/linux/kernel/projects/rt, respectively. After the patch has been applied, the new preempted real-time kernel is available. The preemption of the kernel places an additional layer between user processes and the physical hardware, which takes care of the interrupt handling and assures that the highest priority (real-time) tasks will not be interrupted. Furthermore the RT kernel allows to launch processes with different RT scheduling policies like e.g. round robin. However, not all interrupts can be avoided, meaning that some process which are critical to the OS will still interrupt the first CPU core. These interrupts though are rare and cause delays, which can often be neglected. For very latency sensitive tasks other available CPU cores can be isolated and the process can be assigned to those cores exclusively. Ubuntu 22.04 already comes with a pre-installed RT kernel that can be selected in the *bios* or set up as standard setting using *grub*.

The installation of the preempted RT kernel on Ubuntu 20.04 requires some non-trivial adaptations to the operating system and thus, the exact instructions can be found at [101]. After the installation, the new capabilities of the OS can be verified using the program *cyclicttest* of the *rt-tests* library [102] investigating the latency time of the system. The latency time defines the delay between a process is invoked and the kernel handles it and is a good measure for the RT compatibility of an OS.

Figure 6.4 shows two histograms generated with *cyclicttest* program, in which 1×10^6 processes have been launched while the CPU was stressed. For every process the latency time is measured and plotted in μs on the horizontal axes and on the vertical axes the number of invoked threads with a certain latency time are plotted. The left plot was generated on a standard Ubuntu 20.04 kernel and shows latency times that are relatively widely spread out along the horizontal axis. Especially worrying for RT systems though, would be the max. latency time of 0.574 ms. On the right side a plot of the latency test on a the Ubuntu system with preepted RT kernel is shown and it is clearly visible that the latency times are much lower in avarage and more importantly there are no outliers, as it becomes clear when looking at the max. latency time of 25 μs .

As it is clearly visible in Figure 6.4 the preempted RT patch is performing well, however the providers of the patch do not guarantee hard RT neither can the be hold liable for bugs or errors that might lead to problems during e.g. robotic interventions. Thus, institutions like NASA's Jet Propulsion Laboratory often chose commercial solutions like VxWorks [103], that do guarantee certain properties, but also have a limited list of external libraries that are verified to work on their system. The performance of the VxWorks and RTLinux is comparable though [104] and

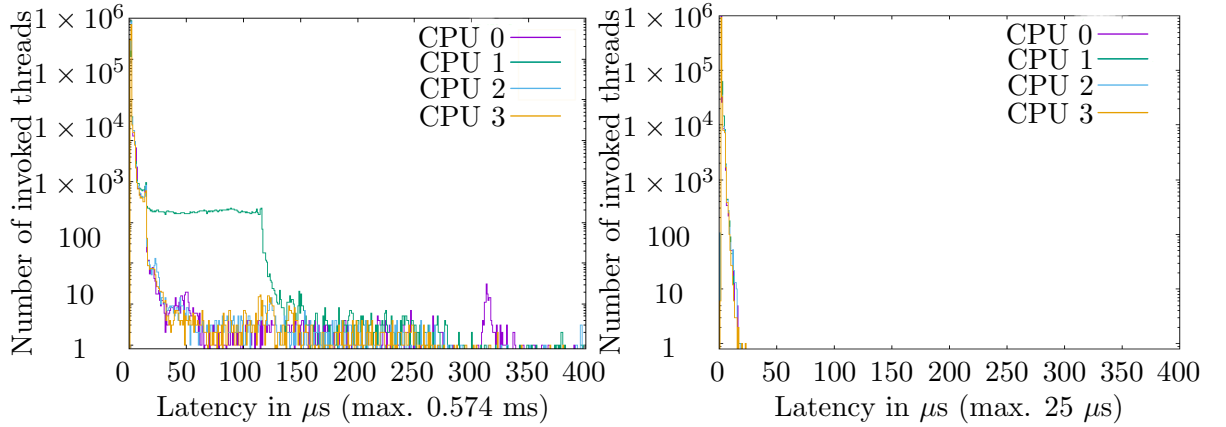


Figure 6.4: Latency times for the standard (left) and the preempted (right) kernel on the same embedded PC with Ubuntu 20.04.

the flexibility in external libraries has higher priority for the CRF developments. However, every robotic system that requires some sort of perception for e.g. mapping and localization will be equipped with two PC's, a low-level RT PC and a powerful perception PC with graphic cards and interfaces for multiple cameras and sensors.

6.2 The Motion Controller Module

As described in the previous sections, it was necessary to develop a motion controller module that builds on clearly defined interfaces and thus allows for fast implementation of new functionality, while ensuring an intuitive and simple architecture. Figure 6.5 shows an overview of the *Navigation* module of the CRF including the *MotionController*. In the following a brief overview of the *navigation* module will be presented, before discussing the *MotionController* in more detail.

On top of Figure 6.5 the *MappingLocalization* module is shown. The input to this module are sensors for perception like lasers or cameras and an optional predefined map if a priori knowledge of the environment is available.

The generated map and estimated position is then input to the *GlobalPathPlanner*, which will plan a global path from the current location to the *Target* input \mathbf{z}_{des} . The *Target* could be an input from an operator, from offline, preplanned targets, from an online, automated target detection or from grasping algorithms.

The results of the *GlobalPathPlanner* will be forwarded to the *LocalPathPlanner* which takes dynamic changes to the map into account and plans a collision free path towards the target.

The generated path from the *LocalPathPlanner* is then input to the *MotionController* module. The *MotionController* contains several submodules with an *I* prefix, point out that the motion controller is designed to accept any modules that follow the interfaces of *ITrajectoryGenerator*, *IKinematics*, *IController* and *IRobot*. The chain of these submodules can optionally be bypassed and a target can directly be sent to the *IRobot* in case a commercial robot is used, that comes with an API and internal motion controller. Depending on the type of the *ITrajectoryGenerator*

the output will either be in task or in joint space and thus, will be forwarded to *IKinematics* or directly to the *IController*.

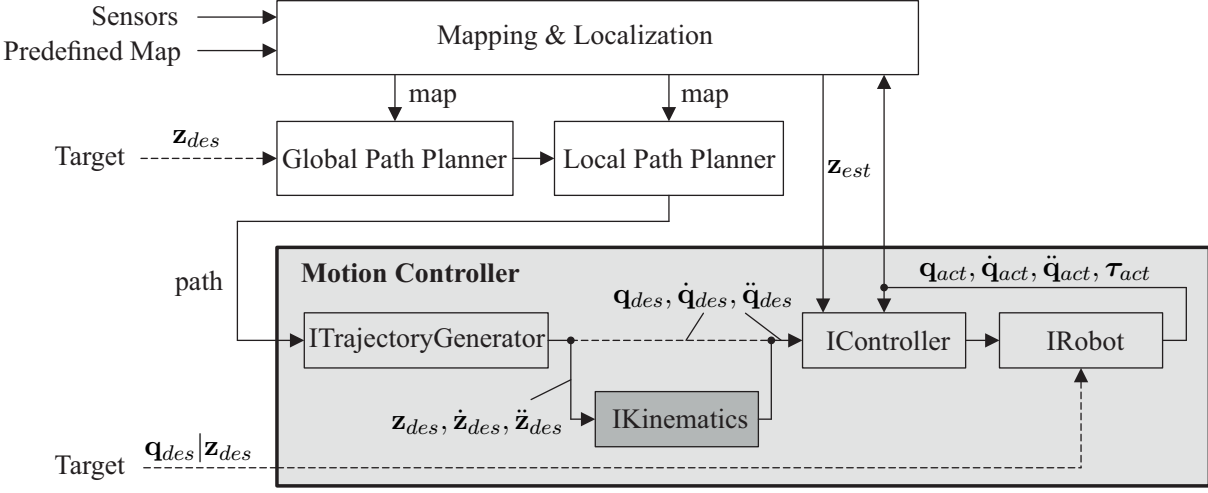


Figure 6.5: Overview of the navigation module of the CRF.

A detailed description of the definitions of the interfaces in used in the motion controller would go beyond the scope of this thesis, however a showcase example of the *IRobot* interface using Unified Modeling Language (UML) diagrams is presented in Figure 6.6 and all relevant other interfaces can be found in Appendix A.1.

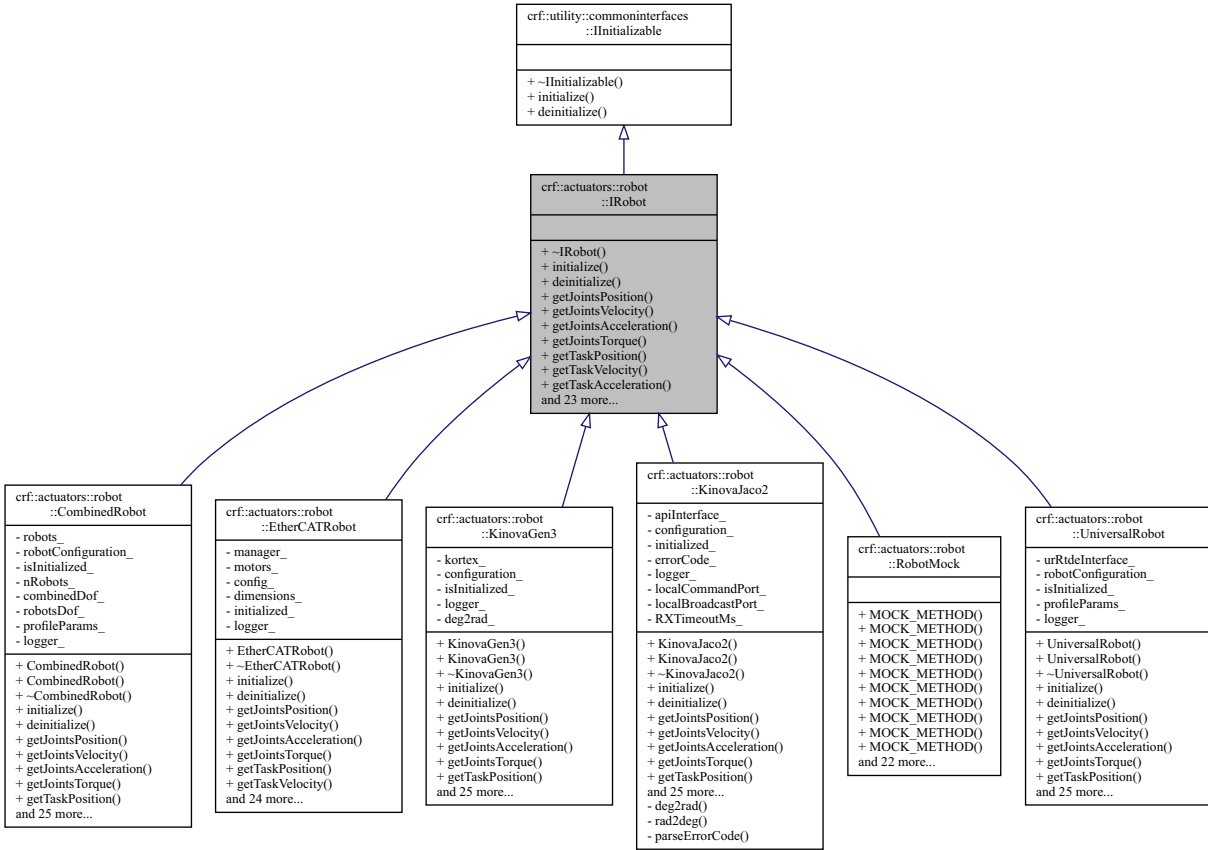


Figure 6.6: UML diagram of the *IRobot* interface and its implementations.

Each of the blocks in the UML diagram represent a class and consist of three sections, of which the top section shows the *namespace* (e.g. *crf::actuators::robot::IRobot*) also indicating where the file is stored in the CRF. The middle section lists member fields/variables of this class and the bottom section provides a list of all member functions in the class. The +/- signs in front of the member functions and fields indicate if the according objects are public (+) or private (-) and the arrows connecting the blocks show the relationship/inheritance between the different classes.

Thus, the UML diagram shows that the *IRobot* interface itself inherits from the *IInitializable* interface, forcing *IRobot* to include the functions *initialize()* and *deinitialize()* as it is shown in the *IRobot* block. As already explained, *IRobot* represents an interface itself and thus, every class inheriting from *IRobot* (shown in the third layer of blocks) is forced to implement all member functions of *IRobot*. The blocks inheriting from *IRobot* are implementations of the interface and are specific to a certain physical robots like *CombinedRobot*, *EtherCATRobot*, *KinovaGen3*, *KinovaJaco2* or *UniversalRobot*. The implementation *RobotMock* is a mock up of the robot class used during unit testing whenever it is not possible to connect to a physical robot, but the SUT relies on a connection to a robot, as explained in detail in Section 6.1.2.

6.3 Development of an Open-Source Kinematics Library

Over the years the MRO section was confronted with increasingly complex remote interventions, that required customized designs of robots to operate in confined work spaces and higher DoF manipulators. This increased complexity, together with requested new functionality from robot operators could not be accounted for with available open-source nor commercial kinematics software. Thus, the development of an open-source kinematics library was initiated.

The developed library is based on the theoretical research presented in Chapter 5. The architecture of the library was designed to be modular and easily expandable for future improvements or additional functionality. Furthermore, it was important to minimize the usage of external libraries and avoid those with restrictive licensing completely.

Figure 6.7 visualizes the modules and external libraries in the so called openRobik (**open**-source **R**edundant **O**ptimization **B**ased **I**nverse **K**inematics) library. The grey blocks in Figure 6.7 are related to the name space or directory in the CRF like *crf::control::* for control related algorithms, *crf::utility::* and *crf::math::* for implementations like helper or mathematical functions that might be used in many other modules or *crf::actuators::* in which robots or other actuators are implemented. The block *external Libraries* lists the dependencies of openRobik library.

The classes in *crf::control::* are visualized using UML diagrams for a more detailed insight in the main concepts, whereas implementations in other name spaces are simply represented by blocks with the name of the class for the sake of clarity. The missing UML diagrams however, can all be found in Appendix A.1. Furthermore all classes have a label next to their name for referencing purposes.

The interface **A** in *crf::control::* defines the callable functions for tasks that are based on measures that are related to the kinematics of the robotic system and implement the functions shown in Table 3.1. The interface defines functions like *getGradient()* or *getTimeDerivative()*,

used to derive the numeric values to control a optimization task or the numeric Jacobian for regulation tasks. The function *enable()* allows for smooth activation or deactivation of a task. The interface **D** defines the callable functions for tasks that are based on measures that are related to the dynamics of the robotic system, but are not implemented in the current release of the library. However, the clear definition of interfaces always allows third parties to extend the according to their needs. Current implementations of the interface **A** are *CollisionAvoidance*, *DesiredJointPosition* and *JointLimits*.

B defines the interface for the CLIK algorithm, which is currently used for all fully automated missions and the interface **C** defines the open loop inverse kinematics (OLIK) algorithm used during teleoperation. This avoids that the solution of the IK algorithm and the actual configuration of the robot drift apart over time. Each of these have one optimization based implementation on velocity level, but again can easily be extended to other IK algorithms on e.g. position or acceleration level.

The interface **E** defines the callable functions for implementations the *IForwardKinematics*. Currently two implementations can be used to generate the forward kinematics of a robotic system. The *KinChainForwardKinematics* gets all the necessary information and thus will be constructed using a Unified Robot Description Format (URDF) of the specific robotic system, whereas the *MathExprForwardKinematics* directly reads the analytic expressions for the forward kinematics from a json file [105], which is called {myRobot}RobotConfig and holds all information about the robotic system required to control it. A brief example of the initialization process for the openRobik library will be given in Section 6.4.

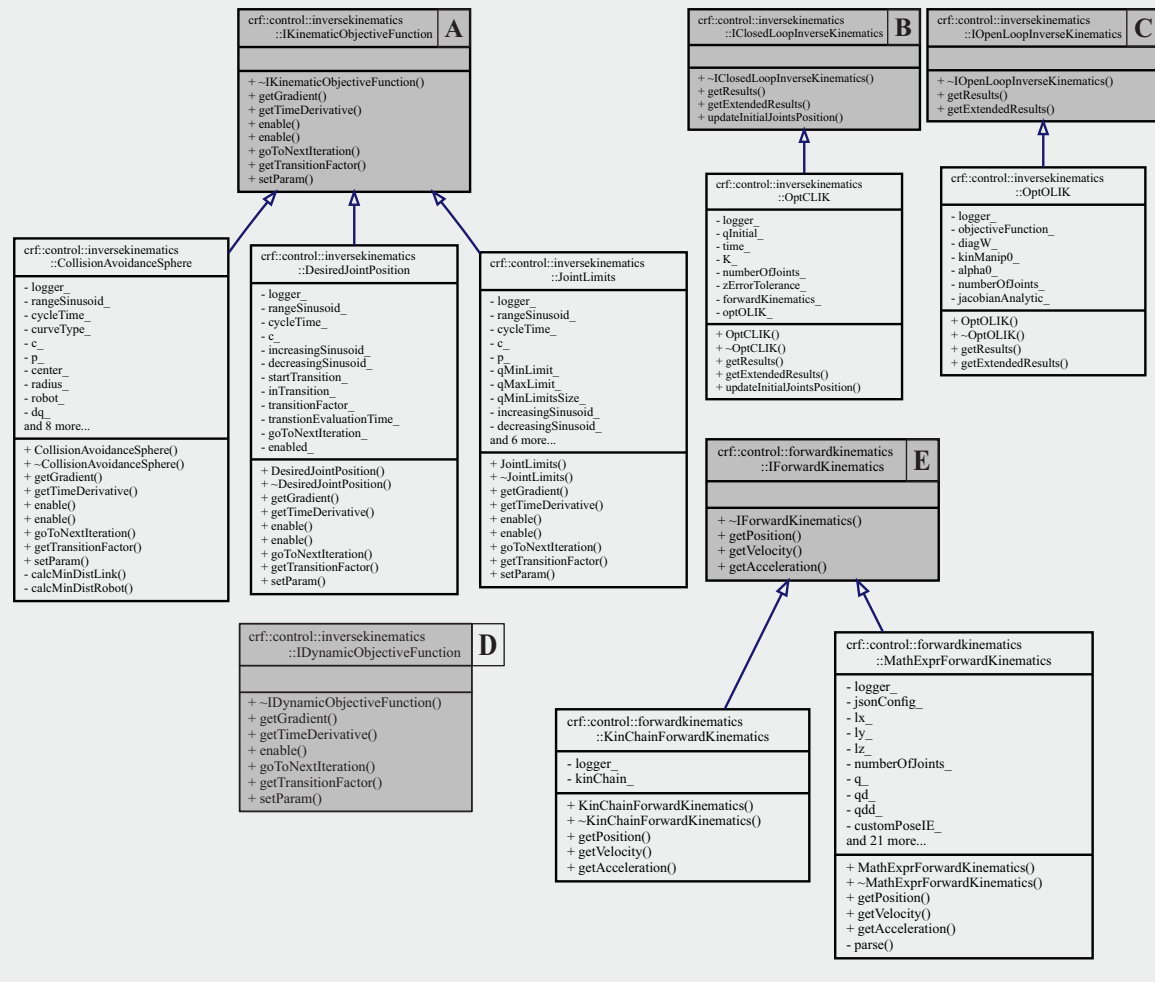
The classes **F** through **L** in the name space *crf::utility::* are used to store joint positions, velocities, etc. according to their name and also include some helper functions e.g. for task positions to convert different orientation representations, overload operators for correct e.g. addition or comparison of the two objects and facilitate efficient printing on the console. The class **M** holds all derivatives of the states of a system and is used to for communication between the control blocks in *crf::control::*. **N** implements a slightly modified version of the *std::expected* class from the C++ standard library *expected* and extends it in a way such that in case of an error also the error code will be returned. Finally, the class **O** holds the Jacobian matrix and provides helper functions for e.g. damped inversion.

The name space *crf::mat::* holds several helper classes for the conversion of orientations discussed in Section 3.1.1 (**P**), the error computation according to 3.1.2 (**Q**), the generation of the kinematic chain presented in 3.3.1 (**R**) and the sinusoid switching parameter according to 5.2.1 (**S**).

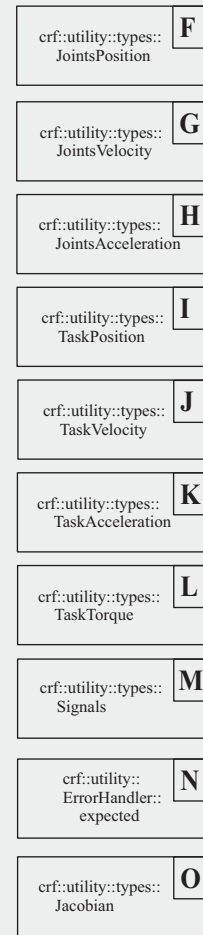
Finally, the *crf::actuators::* name space holds the *RobotiConfiguration* class **T** used to parse the configuration files of specific robots {myRobot}RobotConfig.

openRobik

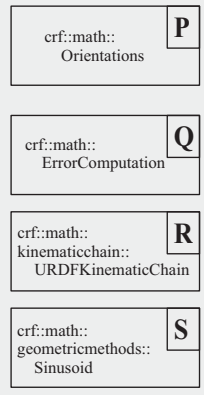
crf::control::



crf::utility::



crf::math::



crf::actuators::



external Libraries



Figure 6.7: Modules and libraries of the Kinematics Library

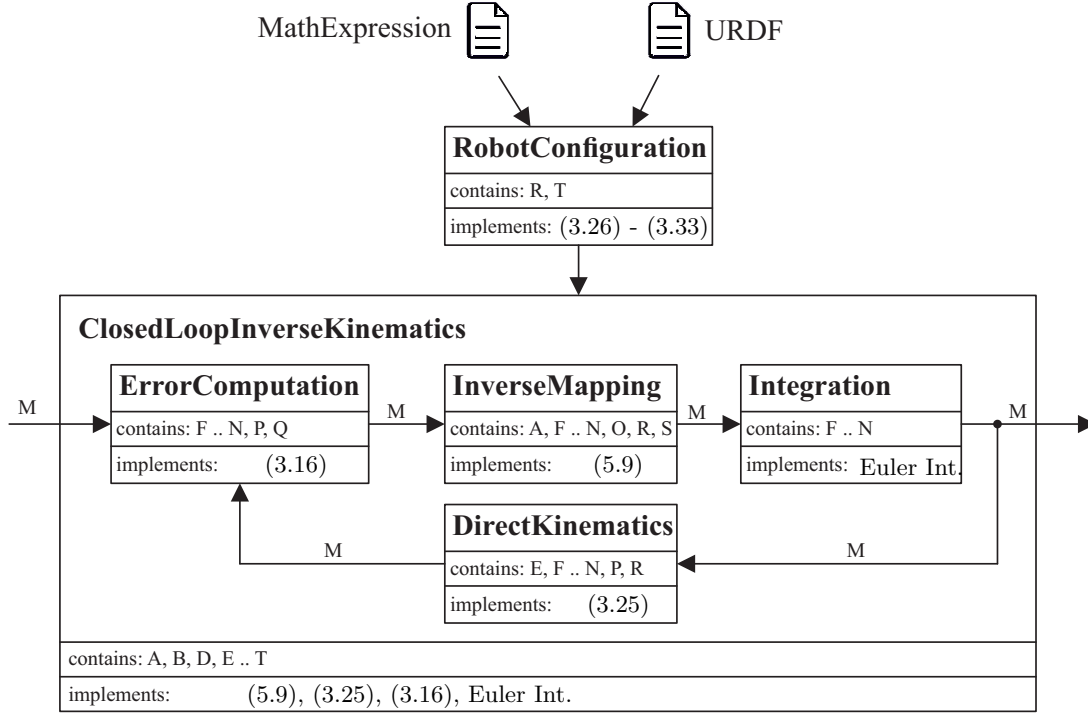


Figure 6.8: Implementation of the inverse kinematics algorithm visualized with the modules from Figure 6.7.

Figure 6.8 visualizes an inverse kinematics algorithm that can be configured with the presented library. The blocks used in this diagram consist of three sections. The top sections describes the main functionality of the block, the middle sections refers to the the contained classes from Figure 6.7 and the bottom section refers to the implemented equations or algorithms that have been presented in previous chapters of this thesis.

On top of Figure 6.8 the two methods to initialize the inverse kinematics algorithm are shown. Either the analytic expressions from the $\{\text{myRobot}\}\text{RobotConfig}$ are parsed with the class **T** or the class **R** will generate the kinematic chain from an URDF file.

The kinematic description of the robot will then be passed to the class **B** which contains several other classes that here make up the CLIK algorithm. The *ErrorComputation* block calculates the error between the input to the class **B** and the feedback, both in the format *Signals* of the class **M**. Then this signal will be passed to the *InverseMapping* block that implements (5.9). The inverse mapping in this example is done on velocity level and thus to output the joint angles, the results will be integrated in the *Integration* block implementing a simple Euler integration. The feedback loop will then be closed via the *DirectKinematics* block that simply implements (4.1) and leading to the block *ClosedLoopInverseKinematics* that implements the algorithm as presented in Section 5.

The library has been rolled out to the entire robotic fleet at CERN and has already proven success and efficiency in automated as well as in teleoperated interventions. It is considered a valuable contribution to the robotics community and therefor the decision was made to publish the library as an open-source tool.

6.3.1 Dependencies & License

In order to publish this library as an open-source tool it was important to ensure that every developer who worked on the library or on parts of the CRF that are used in the library has been employed under a contract that ensures that all developed Intellectual Property (IP) is vested at CERN and not elsewhere. This was confirmed by the knowledge transfer group at CERN. Furthermore, the developed code had to undergo internal review processes to ensure that the code is up to CERN's quality standards and best practices in the open-source community are followed.

To select a license for the openRobik library it was first of all important to identify all licenses of external libraries used in openRobik. The four external libraries and their licenses are listed in Table 6.1.

Table 6.1: List of external libraries and corresponding licenses.

Library	License
Eigen/Dense [106]	Mozilla Public License Version 2.0 [107]
nlohmann/json [108]	MIT License [109]
MathExprTk [87]	MIT License [109]
urdfdom [110]	BSD License [111]

All of the licenses under which the external libraries have been published allow a commercial use and only require that the original license header will either be included in the source files or if binaries are being shared in the documentation of the library.

At CERN most of the open-source publications can be found on a dedicated site on github.com/CERN and usually use the gpl or lgpl licenses [112]. The lpl license also is a very permissive license and allows commercial use of the library but forces developers to contribute to the development of the library in case they change code on the library itself, which will help to further expand the library. Therefor the library of choice for the openRobik library is *LGPL-3.0-or-later*.

6.4 Showcase Examples

In the following a brief example of the usage of the openRobik library for the CamBot robot will be presented. First, the all necessary parameters to control the robot have to be stored in the *CamBotRobotConfig* file in json format. Listing A.1 shows the structure of the config file. In case the kinematics library will be initialized with hard coded expressions in the config file, then this will also be visible in Listing A.1. In case the library will be initialized with a URDF file, then there is also an option to visualize the derived kinematic chain of the robot to facilitate more convenient debugging and prototyping. Figure 6.9 shows the CamBot and the visualized kinematic chain from a URDF file.

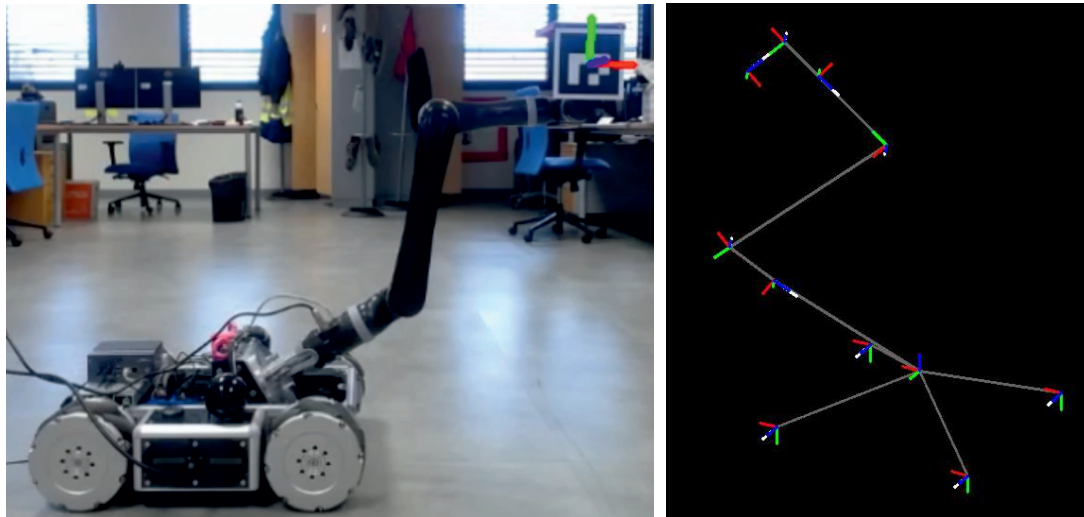


Figure 6.9: CamBot and the derived kinematic chain from the URDF file.

6.5 Conclusion

The newly developed inverse kinematics library represents a significant advancement in robotics, particularly for redundant robots, facilitating an efficient approach to handling complex tasks. The incorporation of smooth switches between tasks has proven success, resulting in enhanced capabilities for the entire robotic fleet at CERN. The positive feedback from robot operators points out the library's practicality and utility in real-world scenarios.

The open source publication of this library marks a milestone, as its potential extends even beyond CERN. The interest shown by external institutions like the European Space Agency (ESA) highlights the broader applicability of the library across diverse industries and settings.

Looking ahead, there are exciting avenues for future work that can further enhance the library's capabilities. One promising direction involves extending the library to perform inverse kinematics at the acceleration level, enabling robots to achieve even more precise and dynamic movements. Additionally, the integration of tasks based on dynamic measures can unlock new possibilities for adaptive and responsive robotic behaviors, leading to improved performance in complex environments.

The expansion of the library to generate cyclic trajectories holds great potential for applications such as repetitive tasks in manufacturing or coordinated movements in swarm robotics. Moreover, the incorporation of quadratic programming techniques to identify collision-free solutions in confined workspaces will be invaluable for ensuring the safety and efficiency of robots operating in constrained environments.

In summary, the newly developed inverse kinematics library represents a milestone, with its impact already resonating within CERN's robotic fleet and beyond. As it continues to evolve, the library's future developments promise to contribute to a new era of advanced robotic control, enabling robots to tackle an even wider array of tasks.

Hardware Design for the FCC Robotic System

ABSTRACT. This chapter delves into the hardware design choices made for the prototype of the FCC robotic system. Building upon the topology and geometry analysis discussed in previous chapters, the hardware design process was initiated. First and foremost, the focus was on determining the optimal mechanical design, taking into account factors such as feasibility and manufacturability. This involved finding a design for the robot links that was both practical and suitable for production. The mechanical design, along with its dynamics and kinematics parameters, served as the foundation for subsequent decisions regarding the robot motors, encoders, and corresponding drivers. It was crucial to ensure a seamless integration of cabling and electronic devices into the overall design. A brief summary will highlighting the challenges encountered, lessons learned, and potential areas for improvement within the robotic system.

Chapter Content

7.1	Mechanical Design . . .	114
7.2	Motor Design	116
7.3	Electrical Design	119
7.4	Installation	123
7.5	Conclusion	125

Related Publications

- | | | |
|-------|--|---|
| [113] | A. Diaz Rosales, Gamper H., M. Di Castro | Control Design Optimizations of Robots for the Maintenance and Inspection of Particle Accelerators - ICALEPCS |
|-------|--|---|
-

The development of a prototype robotic system for the Future Circular Collider (FCC) is a major milestone for upcoming proof-of-concept studies. In these studies, the robotic system will be deployed within a 50 m long mock-up of the FCC main tunnel, including all components envisaged for the actual accelerator tunnel. This setup includes electric cabling, cryo installations, the booster and beam line, section doors, ventilation equipment, communication components, sensors, and transport vehicles.

The primary challenges in the proof-of-concept studies for the robotic system revolve around demonstrating its efficient and robust operation. This includes validating both the mechanical design and the control and automation concepts for various tasks. Additionally, a critical aspect is showcasing an efficient, intuitive, and robust teleoperation mode, which will include the visualization of the robot's environment, leveraging virtual reality systems and incorporating haptic feedback.

The mechanical design of the prototype is based on the recommendations outlined for the robotic system, as presented in Table 2.4. It also aligns with the optimal topology and geometry presented in Figure 4.24 and Table 4.5. This approach ensures that the prototype meets the specified requirements, laying the foundation for the success of the FCC robotic system.

7.1 Mechanical Design

The mechanical design phase was based on a well-defined topology and geometry of the robotic system, as identified during the design optimization presented in Chapter 4. To expedite prototype development, the last six joints of the 11 DoF system were modeled using a light-weight industrial robot, the Universal Robot UR10e. However, the first five joints were entirely designed from scratch to meet the requirements established in the previous chapters:

- Follow the optimal topology and geometry outlined in Chapter 4.
- Implement a light-weight design to minimize the load on the FCC tunnel's ceiling.
- Ensure that the natural frequencies of the robot links lie above those that could be excited with peak motor torque.

Initially, the robot joints were positioned relative to each other, adhering to the specified optimal topology and geometry. By following this optimal design for all links of the robotic system, we arrived at the complete assembly, as presented in Figure 7.1. The CAD models showcase the robotic system in both an extended configuration in Figure 7.1(a) and a folded configuration in Figure 7.1(b). The folded configuration demonstrates how the robot can pass through the fire and section doors of the FCC main tunnel. This flexibility in configuration enables the robotic system to adapt to the specific requirements of the tunnel and efficiently navigate through confined spaces.

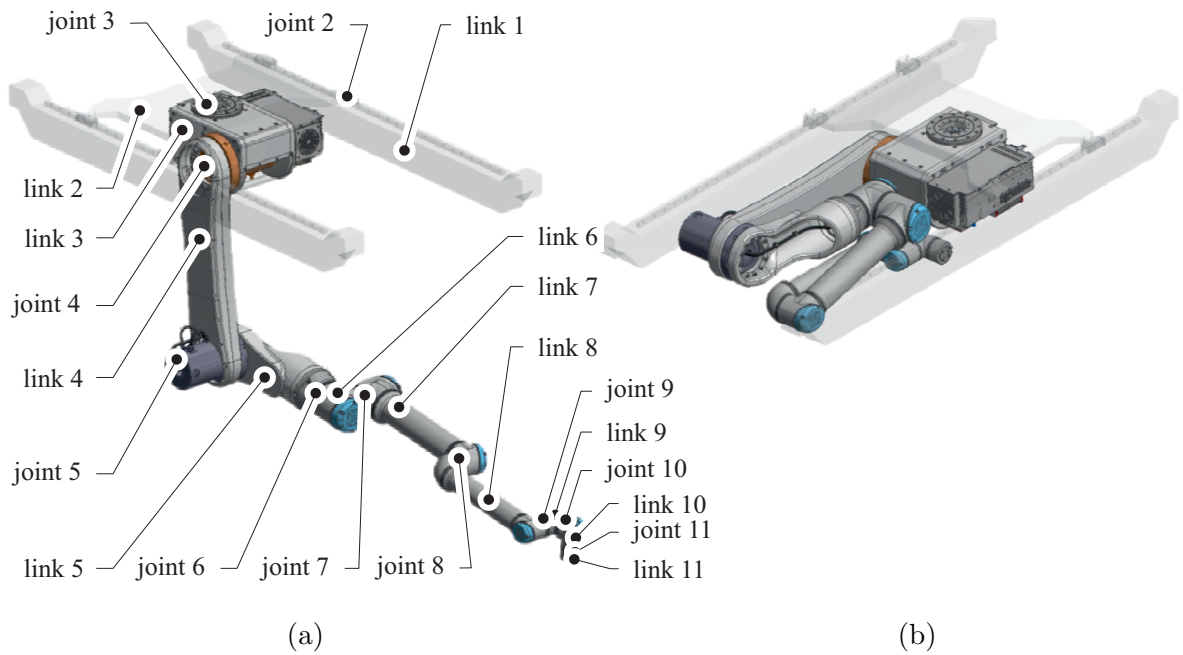


Figure 7.1: CAD model of the full robotic system.

Then, these joints were connected using solid aluminum rods, serving as initial approximations for robot links. While these rods were impractically heavy, they provided a good starting point for subsequent topology analysis.

To reduce the weight of the link to the minimum while maintaining a constant stress throughout the material, topology optimization was performed in two iterations, as depicted in Figure 7.2(a) and 7.2(b). A simplified version of the optimization process was also conducted to ensure efficient manufacturing, illustrated in Figure 7.2(c). This step helped to refine the design and achieve a more practical and effective solution.

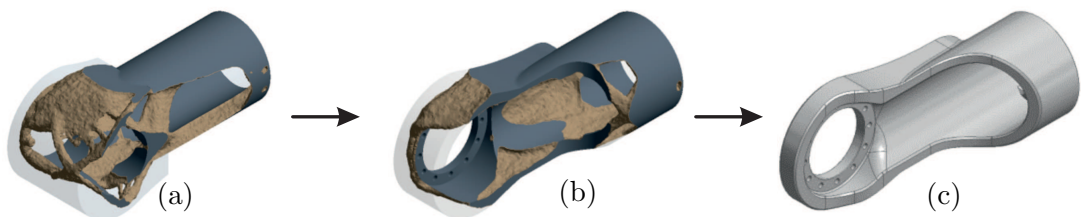


Figure 7.2: Topology optimization of link 5. The weight of the link was reduced from 26.4 kg (a) to 15.3 kg (b) and finally to 9.9 kg (c).

Following the material removal process, a structural stress analysis was conducted to verify the strength of the links. Figure 7.3 displays the stress distribution on the material under full load conditions. Minor adjustments were made at this stage primarily to enhance the ease of mounting the robot links. These modifications ensured a robust and reliable construction while maintaining the desired structural integrity.

Ensuring the design of robot links prevents the excitation of natural frequencies during nominal operation is of paramount importance. Resonances resulting from such excitations could lead

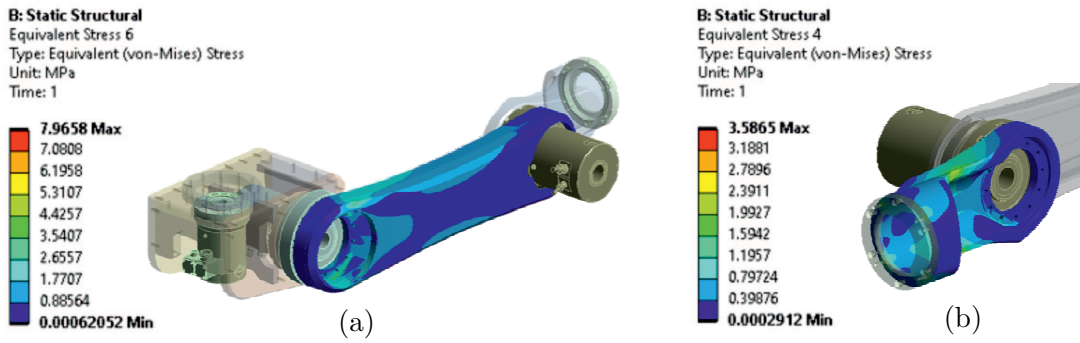


Figure 7.3: Stress analysis of links 4 and 5.

to compromised precision in control objectives or, in worst cases, cause hardware damage. The analysis of natural frequencies for the robotic arm, combining link 4 and 5, is depicted in Figure 7.4. This analysis evaluates the first mode (horizontal oscillation) and the second mode (vertical oscillation). By understanding and addressing potential resonances, the design team can optimize the system's performance and maintain its stability during regular operations.

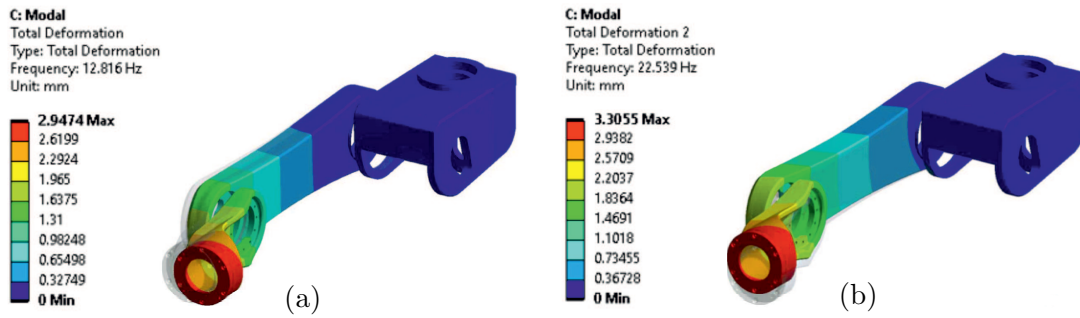


Figure 7.4: Natural frequency analysis of links 4 and 5.

7.2 Motor Design

Based on first rough simulations it quickly became clear that the required torques for this over 3 m long robotic arm, do not allow for off the shelf motor solutions. Therefore, detailed dynamic simulations of the robotic system have been developed in order to verify the motor torques and facilitate easy adjustments of parameters throughout the iterative development process between mechanical desing, simulation and updating of parameters. Based on certain requirements a pre-selection of two motor types has been performed, for which then dynamic simulations have been performed in order to verify that they fit. The criteria for the pre-selection process were:

- An operation with a voltage of 48 V and a maximum current 75 A provided by available motor drivers.
- A similar base design of the motors to decrease the cost of the customized solutions. Therefore the rotor, stator and housing is the same for both motors. Components that can be adjusted are the coiling, gearbox, encoders and brakes.

- Strain wave gear.

The kinematic and dynamic model was developed following the techniques described in Sections 3.3 and 3.5, respectively. The analytic models were generated in Maple [114] and were then passed to Matlab [79] for numerical simulations. The necessary geometric and dynamic parameters for the part of the robot that was designed from scratch have been exported and iteratively updated from Inventor [115]. The geometric parameters for the UR10e could be found in datasheets, whereas the dynamic parameters were extracted from a safety file, stored on the UR control box, that contains the parameters which were specifically identified from the manufacturer for the specific robotic arm with which the control box was shipped.

Figure 7.5 shows two examples of the performed simulations to identify the required motor parameters. Although the specific trajectories that will be performed with the real system, were not clear at this stage, some generic trajectories that exploit the full workspace and show a total velocity vector of 0.6 m/s, the maximum expected end effector velocity were generated. In order to generate smooth end effector trajectories, NURB splines have been defined to generate a spiral and elliptic movement as shown in the left and right column in Figure 7.5, respectively.

In Figures 7.5(a) and 7.5(f) the trajectories and the robot, using convex collision objects are visualized. Then, the inverse kinematics presented in Section 7.5 was used to obtain the joint angles shown in Figures 7.5(b) and 7.5(g) and the translational end effector velocities are shown in Figures 7.5(c) and 7.5(h). This finally leads to the required motor torques, necessary to follow the given trajectories shown in Figures 7.5(d) and 7.5(i) for the motor in joint 4 and Figures 7.5(e) and 7.5(j) for the motor in joint 5. The dashed lines in these plots indicate the velocity dependent torque limits of the motors, with the *Limit*, *S3* and *S1* curve indicating the maximum torque that can be provided for less than 3 s, less than 1 min and permanently.

The results of the simulation show that the motor for joint 4 is under the *S3* curve when the arm is fully extended, which is acceptable since the arm is not supposed to rest at a fully extended configuration for more than 3 min. The motor for joint 3 on the other hand has more margin to the *S3* curve and in general a slightly weaker motor could be selected, but this would increase the development costs and was therefor neglected.

Thus, the final motor configuration for joint 4 consists of the base layout of the Harmonic Drive AP58 motor and a strain wave gear with a gear reduction of 160, since very low dynamics but high static loads are expected during operation. Furthermore two digital encoders, both single turn encoders with a resolution of 21 bit, will be installed on the motor as well as on the output side. The high resolution on both the motor and output side will allow to use the gear elasticity for direct torque measurements. Additionally, a 10 Nm brake is installed on the motor side, leading to the final motor configuration for joint 4 as Alopex-58A-160-DL-O-MXB-B-SXB-SP.

The final motor configuration for joint 5 consists of the base layout of the Harmonic Drive AP50 motor and a strain wave gear with a gear reduction of 160. This motor is expected to perform rotations more than 360 degrees and thus the encoder configuration shows a multiturn encoder with 20 bit resolution on the output side and a single turn encoder with 21 bit resolution on the motor side. Additionally, a 5 Nm brake is installed on the motor side, leading to the final motor configuration for joint 5 as Alopex-50A-160-DL-O-MXB-B-MXB-SP.

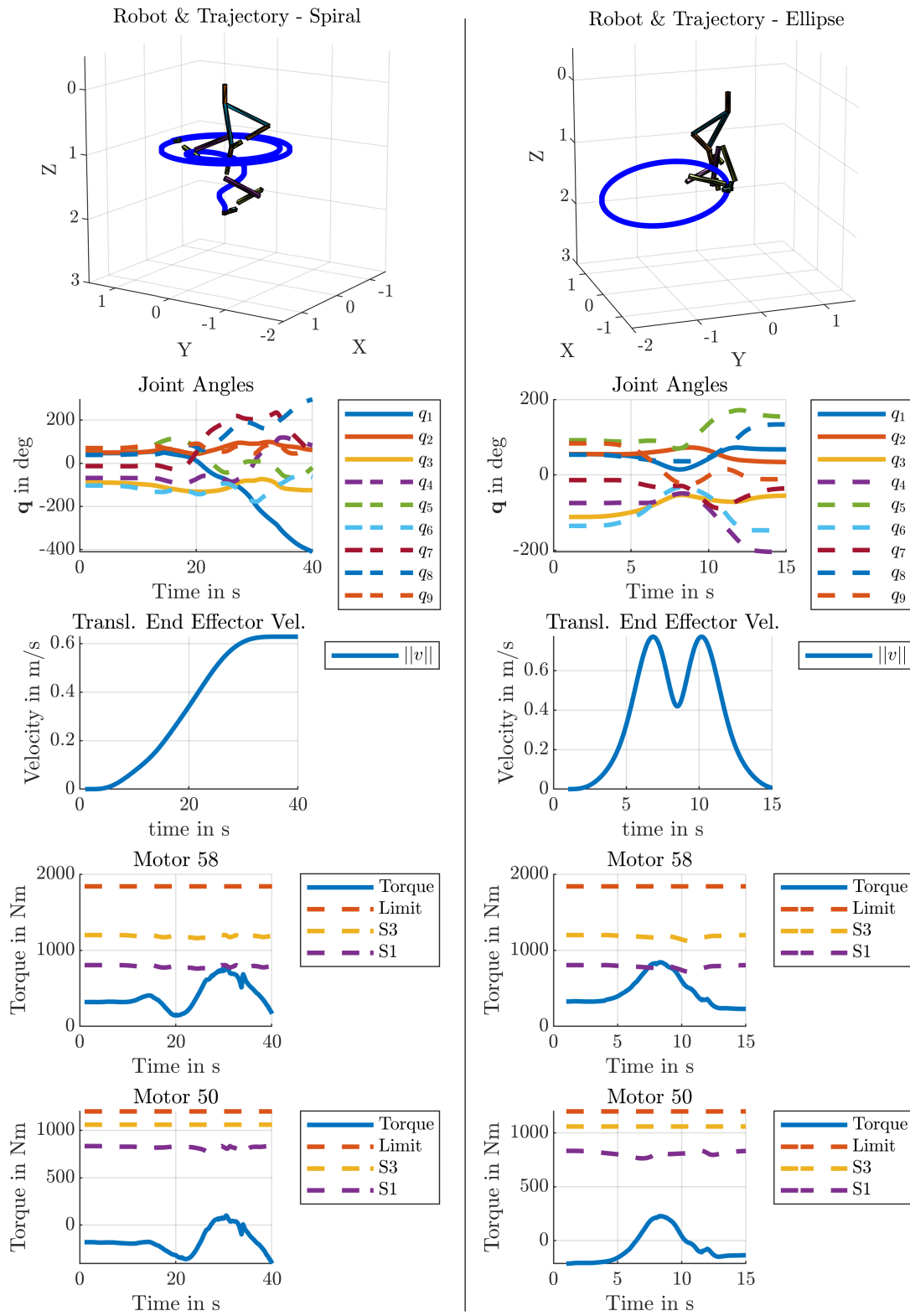


Figure 7.5: Simulation data for motor design.

7.3 Electrical Design

The electrical design plays a crucial role in the overall performance and functionality of the robotic system. It addresses the installation of power supply, safety components, and communication channels, all of which directly affect the system's reliability and maintainability. One area that presented particular challenges was the cabling of robotic systems, as it often involves routing cables through the movable joints of the robot.

Initially, the electrical design requirements were defined, which ensured that all necessary aspects were considered and accounted for. Subsequently, the focus was to find the simplest and most efficient solution, with the aim of minimizing the number of cables and devices involved. This approach not only streamlines the design process but also enhances the overall performance of the robotic system.

The main requirements for this system are:

- Power Supply
 - Powered via battery and fixed grid connection.
 - Output power of 6 kW at a current of 125 A.
 - Stable DC supply at 48 V.
- Safety
 - Emergency stops for human and robot safety.
 - EMC standards fulfilled.
- Communication
 - EtherCAT
 - CANopen

Figure 7.6 provides a simplified electrical schematic of the prototype. In the figure of the real system, labels indicate various integrated elements and the following Table 7.1 provides a more detailed list of all the parts shown in Figure 7.6.

On the left-hand side of the figure, curly brackets indicate the specific components installed in different physical spaces within the robotic system, including the robot links, control room, and end effector. The dashed, black lines signify the separation of these physical spaces by the robot joints. Additionally, two dashed, black lines at the top indicate the division between the robot and the control room.

The power supply is represented by red lines, distinguished by solid, dash-dotted, and dashed patterns, corresponding to 48 V, 24 V, and 12 V, respectively. Two DC/DC converters handle the conversion between these power lines.

The green lines denote EtherCAT communication channels, while the blue lines depict safety-related logic networks (safety torque off - STO) connected to the emergency stop button.

Table 7.1: List of the components of the FCC robotic system.

Part	Description	Reference
A	Various control interfaces	–
B	Emergency stop button	APEM A01-AX1607US
C	3 phase power switch	–
D	On/off console for UR control box	–
E	Power supply	ABB GP100H3M54TEZ
F	HarmonicDrive motor AP32	Alopex-32A-160-DD-M-R00-B-SP
G	Motor driver Elmo	ELMO G-CEL50/100ERS
H	Safety relais	Pilz PZE X4
I	Embedded main PC	ROScube Pico TGL
J	HarmonicDrive motor AP58	Alopex-58A-160-DL-O-MXB-B-SXB-SP
K	Motor driver Ingenia	Ingenia DM1-E-XCR-02
L	Braking chopper	inhouse PCB design, see Figure 7.8
M	UR control box	OEM controller e-series
N	HarmonicDrive motor AP50	Alopex-50A-160-DL-O-MXB-B-MXB-SP
O	Motor driver Ingenia	Ingenia DM1-E-XCR-02
P	Embedded sensor PC	Fitlet 2
Q	Various tools	–
R	Universal Robot	Universal Robot UR10e
S	Operational Amplifier	Microchip TC4426M
T	Resistor (R1)	47 Ω
U	Resistor (R2)	10 k Ω
V	Capacitor (C)	100 nF
W	Schottky Diode (SD)	Diodes SB570
X	Power MOSFET	Hexfet IRLZ44NPbF
Y	Power Resistor (R)	Multicomp MC14735

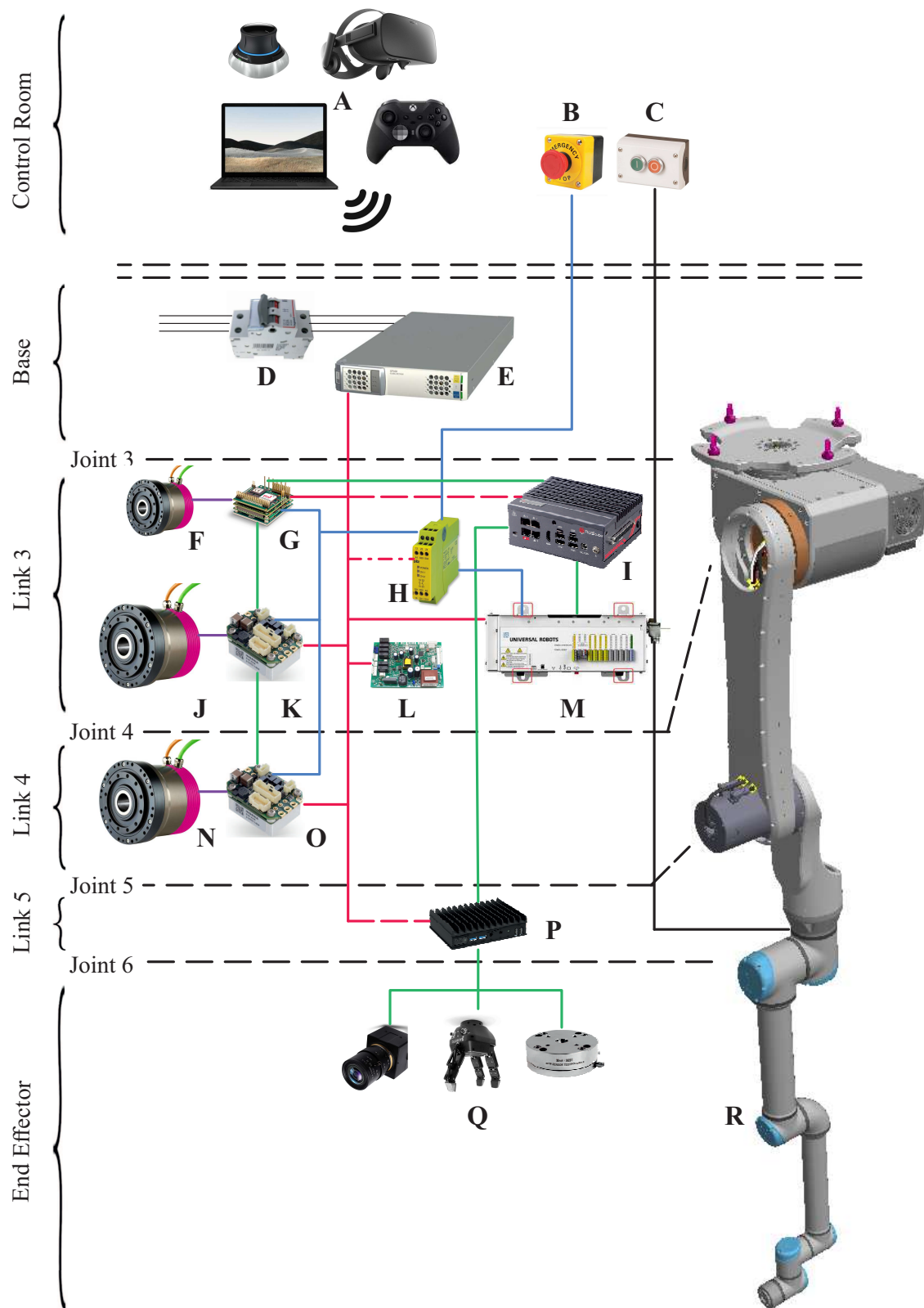


Figure 7.6: Electrical Schematic of the FCC Prototype

A black line represents a logic connection to the Universal Robot's control box, enabling the switching on and off of the UR10e. It also resembles the connection between the control box and the motors of the UR10e, including communication, power, and safety aspects.

The purple lines symbolize the PWM and encoder signals exchanged between the motor drivers and the motors.

Lastly, the number of intersections between the dashed black lines and other lines corresponds to the number of cables that must pass through the associated joint and, consequently, the hollow shaft of the motor within that joint. This information provides insight into the cable routing complexity.

Overall, the schematic offers a visual representation of the prototype's electrical setup, allowing for a better understanding of the system's configuration.

Figure 7.6 provides a clear enumeration of devices installed in the robotic system, with bold letters representing each device. In the control room, there are various control options for teleoperation denoted as **A**, an emergency stop button marked as **B**, and an on/off switch for the UR10e indicated as **C**.

To ensure power supply protection, a fuse labeled **D** protects the power supply **E**, which derives power from a three-phase grid and delivers a voltage of 48 VDC with a maximum power output of 6 kW.

In cases where the system generates more energy (e.g., motors in generator mode) than it consumes (e.g., motors in motor mode), a braking chopper denoted as **L** dissipates excess energy through power resistors. This prevents protective overvoltage shutdowns of the motor drivers.

The motors **F**, **J**, and **N** (as explained in Section 7.1) are controlled using their respective motor drivers **G**, **K**, and **O**. The selection of motor drivers **K** and **O** was based on the parameters of the corresponding motors and the required functionality of two inputs for digital encoders. It is worth noting that the motor/driver combination **F/G** was chosen based on the available material from previous projects, which resulted in the difference in driver brands.

When the emergency stop button **B** is activated, it triggers the safety relay **H**, which then switches a direct channel and a delayed channel. The direct channel is connected to the digital inputs of the driver, initiating a QuickStop procedure that gradually decreases the motor velocity with maximum acceleration. Only after this process, the delayed channel is activated, transmitting the actual safety torque off (STO) signal. This sequential operation is important because the immediate activation of the STO signal would abruptly set the motor torque to zero and engage the brakes, causing excessive wear, which would destroy the brakes within five to ten cycles.

The embedded PC **I** serves as the primary control unit, responsible for solving kinematics and dynamics, as well as running the position control loop. This PC operates on a Linux operating system with a preempted real-time kernel. Communication with the motor drivers **G**, **K**, and **O** is accomplished through CANopen over EtherCAT, while communication with the UR control box **M** utilizes the Real-Time Data Exchange (RTDE) protocol over Ethernet.

Additionally, a secondary embedded PC **P**, with lower processing power, is dedicated to controlling various end effectors and sensors **Q** that may be installed on the last joint of the robot. In such cases, only one Ethernet cable is routed to the main PC **I** for further data processing or control purposes, such as managing grippers and tools.

In order to limit the work load of developing this prototype, the last 6 DoF were realized using a Universal Robot UR10e **R**.

The simplified schematic depicted in Figure 7.6 served as the foundation for generating detailed electrical drawings prior to the assembly of the initial prototype. However, the detailed technical drawings fall outside the scope of this thesis and thus are not provided here.

To ensure the feasibility of cable routing, particularly within the robot's joints where all cables must pass through the hollow shafts of motors **F**, **J**, and **N**, the design concept underwent validation using CAD software. This step was crucial in assessing the viability of the design before proceeding to the production phase.

Figure 7.7 presents a comparison between the design simulated in the CAD software and the challenges encountered during the actual implementation. This visual representation provides an impression of the challenges that emerged and aims to support further design iterations and improvements.

The braking chopper was developed specifically for the FCC robotic system. It will be triggered by the digital outs of the Ingenia driver that allow for a observation of the voltage in the main circuit and give a logic 1 as an output if the voltage in the main circuit raises above a threshold of 55 V. This signal is then used to turn on the energy dissipation.

7.4 Installation

Figure 7.9 shows snippets of the three installation phases: First, shown in Figure 7.9a, the robot was assembled on a testing bench and first checks on the electrical circuit, motors, encoders, brakes and emergency systems were performed. While most of the systems worked as expected, sever problems were encountered when turning on and controlling the motors in velocity mode and at first these problems seemed to occur randomly. Only after inspecting the bit streams of the encoder cables using oscilloscopes uncovered increased noise levels on the encoder signals. After many hours of searching for the error, it finally became clear that this behavior was due to EMC (electromagnetic compatibility) problems and the mistake happened during the mechanical design of the robot links: The aluminum robot links have been anodized. Usually, the links of industrial robots as well as the purchased motor's housing are finished with a specific alloy *SurTec 650* that shows a very low electrical resistance. This ensures that potentially induced currents, from e.g. the PWM signals in the cables between the motor drivers and the coils of the motor, find a low resistance path from the motor housing via the robot link, back to the motor driver - the source (blue path in Figure 7.7). In this case however, the anodization of the robot links created a high electrical resistance on the surface and thus the electrical path from motor housing to robot link has been insulated. In addition, the shielding of the encoder cable was connected to ground on both ends, the motor housing and the motor driver housing. Thus, a more attractive electrical path with lower resistance was created from the motor housing via

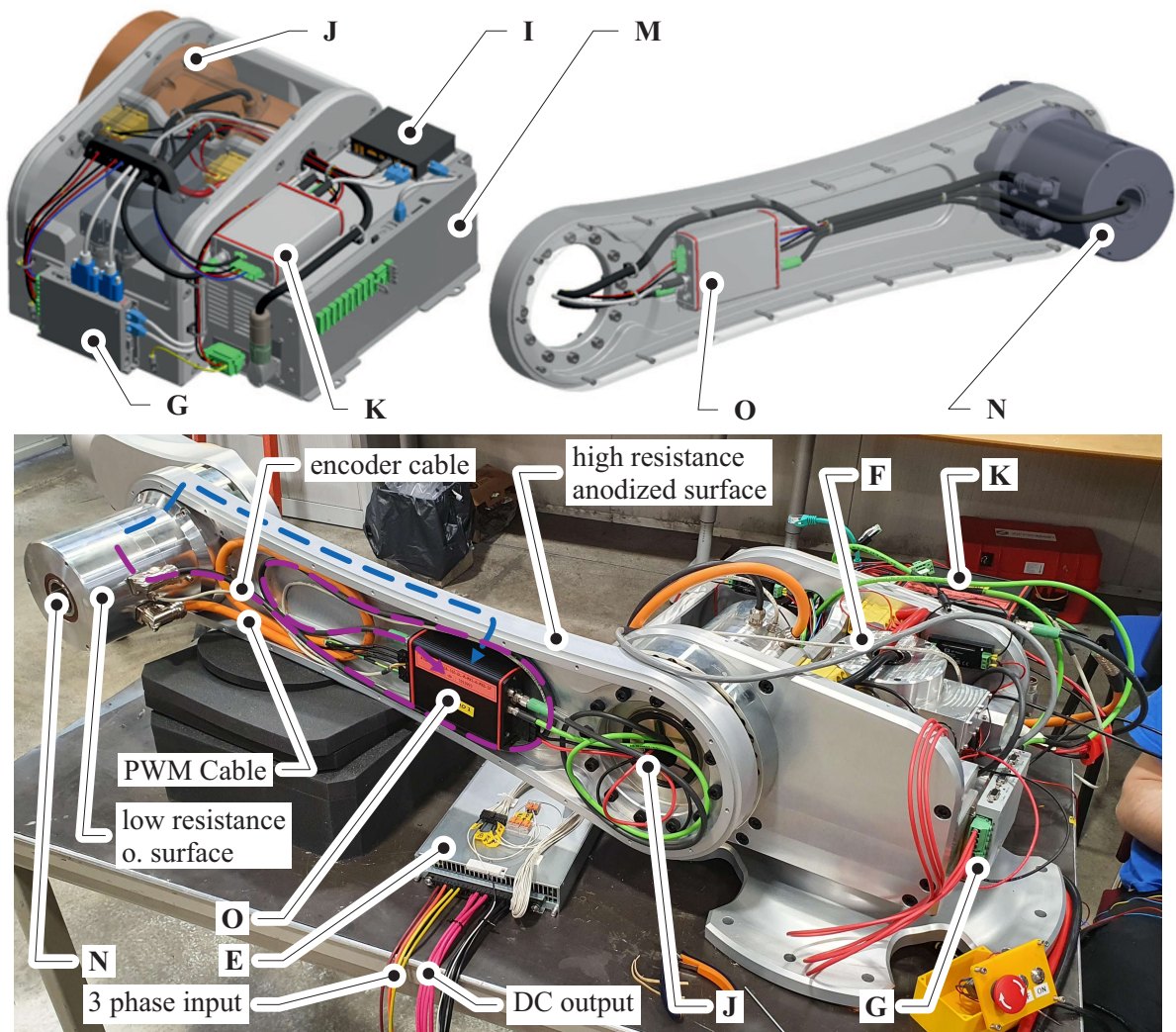


Figure 7.7: The CAD design of the robot base including the cabling (top) and the cabling work ongoing on the prototype (bottom).

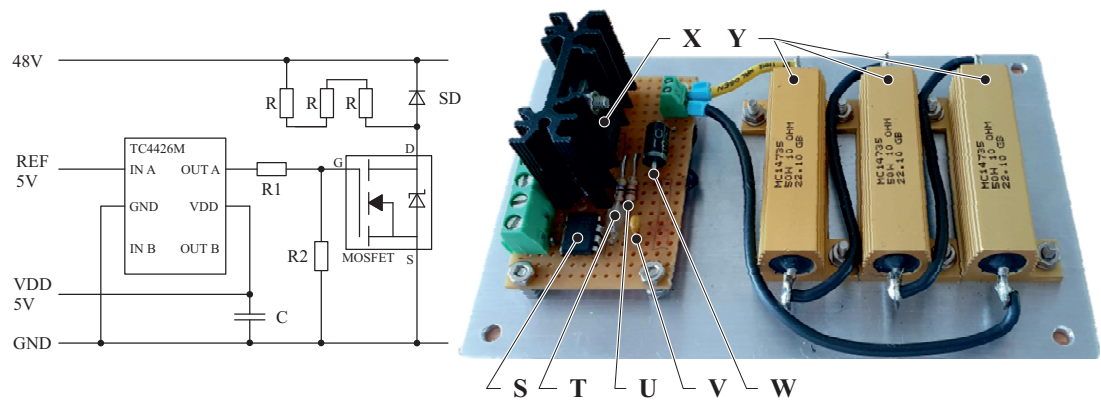


Figure 7.8: The designed circuit monitors the voltage in the main circuit. In case of an voltage increase, energy will be dissipated via the power resistors.

the encoder cable, back to the motor driver - the source (purple path in Figure 7.7). These currents in the encoder cable's shielding were the cause for the noise in the velocity signals that made it impossible to control the motors in first place. Once the problem was understood, the solution was simple: The encoder cable's shielding was cut in the middle, such that this path would no longer be the one with lowest resistance. The bit streams then were clean and the motors running smoothly. After this first phase of installation the robot was brought back to the laboratory, disassembled and the electrical cabling was upgraded, especially in terms of EMC.

During the second phase of installation the robot was lifted with an overhead crane into a mock-up of the LHC tunnel and mounted on the ceiling, that was previously especially reinforced, see Figure 7.9(b). Figure 7.9(c) shows the final installation phase with a collimator installed under the FCC robot for proof of concept studies.

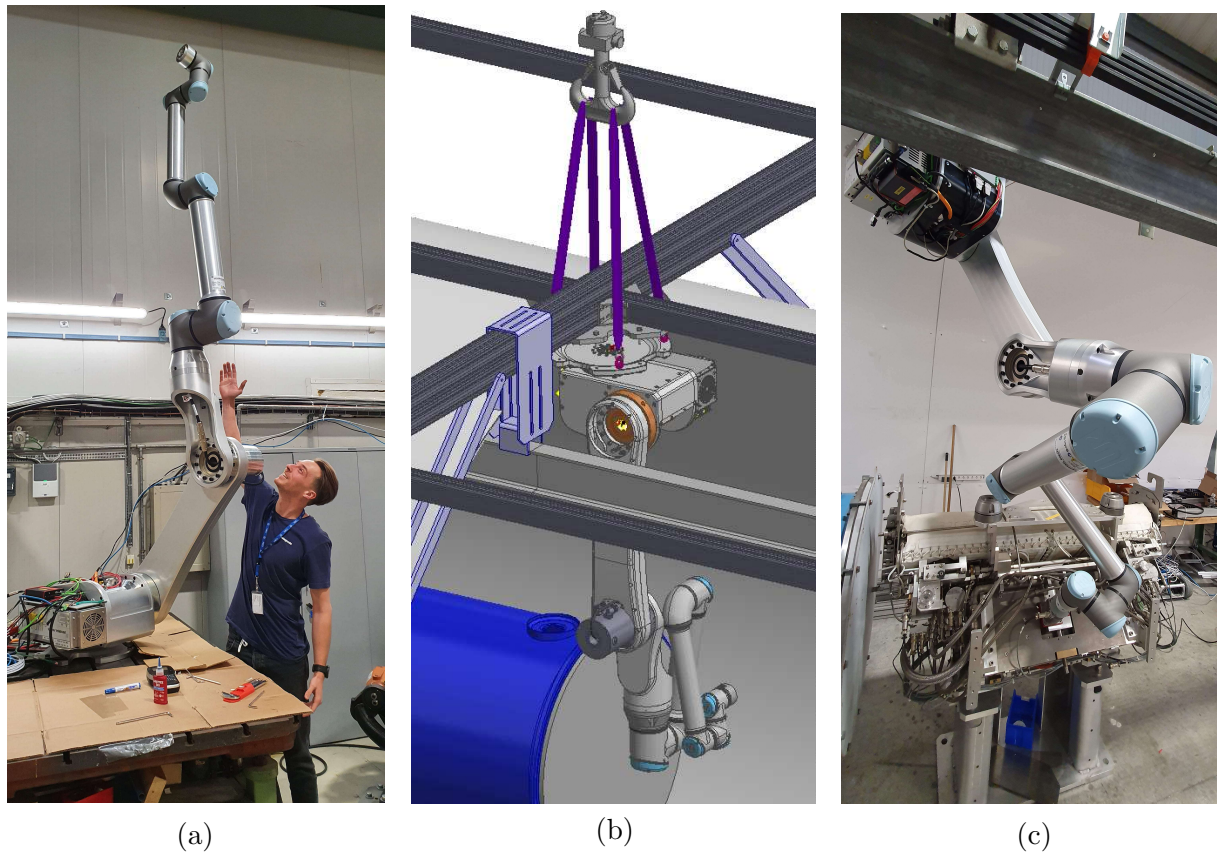


Figure 7.9: Installation phases of the FCC robot

7.5 Conclusion

The focus of the design process was on the development of joints and links 3-5 of the robotic system, while joints 6-11 have been realized with a commercial solution from Universal Robots to speed up the construction of the first version of the prototype. A drawback is the deviation from the optimal geometry identified during design optimization in Chapter 4. However, this trade-off enables fast prototyping and facilitates the evaluation of software and sensors. Lessons learned during this phase will be important for future developments, especially concerning the development of joints 1 and 2 and the joints 6 - 11 in a fully customized manner.

Throughout the design, assembly, and installation of the prototype, numerous valuable insights were gained, crucial for the next version and extension of the robotic system. The simulation results, combining dynamic modeling for motor torque, stress analysis, elasticity, and Eigen modes and frequencies using ANSYS and Matlab, proved highly accurate, validating the employed tool chain. However, the EMC effects of the 80 A PWM signals were initially underestimated, necessitating substantial updates for smooth motor and encoder operation. Future versions must address this during material and coating selection.

Due to supply shortages, the motor driver selection resulted in the installation of two different driver types by Elmo and Ingenia, leading to significant software challenges due to varying communication interfaces. Future versions should opt for a single motor driver manufacturer. The cabling complexity for in-house-designed links, acceptable for larger motors, may require slip rings for the smaller motors of the last six joints to ensure smooth operation and allow for multiple revolutions. The energy introduced by motors in generator mode was not always consumed by active motors, leading to over voltage shutdowns. To mitigate this, a PCB module, the braking chopper, was designed to dissipate excess energy.

Despite these challenges, the overall design, assembly, and installation were successful, providing valuable lessons crucial for progressing toward a fully customized robotic solution for the FCC.

Conclusion & Outlook

ABSTRACT. This thesis provides the foundation for a holistic robotic system for FCC in terms of requirements, restrictions, logistics, robot hardware and software. First, an extensive survey about the requirements and restrictions on a robotic system has been conducted. Considering these findings, a novel design optimization algorithm was developed in order to identify the optimal topology and geometry of the robotic system for FCC, which converged at a system with 11 DoF. The control of such a highly redundant system with sufficient precision required the development of a novel inverse kinematics algorithm. This development furthermore led to a new motion controller architecture for the CRF, following a generic and modular approach that facilitates fast prototyping and reusability. In Chapter 7 of this thesis the lessons learned during the construction and assembly phase of the prototype are presented. This thesis marks significant progress towards achieving the goal of an holistic robotic system for FCC.

The conclusion of this Ph.D. thesis marks the culmination of an exhaustive research journey dedicated to advancing the development of a robotic system tailored specifically for the FCC project. In this concluding chapter, the primary discoveries made throughout this thesis and future developments will be summarized.

The central aim of the first phase of the thesis has been to lay a robust foundation for the development of a robotic system adapted to meet the unique requirements and constraints imposed by the FCC, see Chapter 2. This undertaking started with a thorough examination of the multifaceted restrictions and demands inherent in a project of this scale and complexity. This first phase was concluded with clearly defined recommendations on a robotic system and the official integration of a robotic system in the base line layout of the FCC main tunnel.

Departing from these high-level recommendations for a robotic system a novel design optimization algorithm has been developed to identify the optimal topology and geometry of the robotic system. The novelty of the algorithm lies in the specific type of cost function called pruning function, that allows for simultaneous optimization of the topology and geometry. The algorithm proved to be a valuable tool for manipulator optimization in general and was subsequently employed to identify optimal designs for other projects in the MRO section, like the RF cavity inspection. The design optimization process for the FCC robot has converged at a 11 DoF robotic structure.

State of the art inverse kinematics algorithms were unable to satisfy the recommendations presented Chapter 2 and at the same time perform the redundancy resolution for a highly redundant robotic structure as described in Chapter 4. Therefor a novel inverse kinematics algorithm has been developed, which not only effectively addresses the challenge of controlling redundancies but also introduces the possibility to enable and disable tasks during runtime, which significantly improves the efficiency and user experience during teleoperation. Furthermore it holds great promise for the broader scientific community, as it is scheduled for release as an open-source tool. The European Space Agency (ESA) recently reached out and communicated their interest in the developed library. Currently a demonstration of the capabilities of the inverse kinematics library is being prepared on the example of the robotic manipulator (Canada Arm) of the International Space Station (ISS).

In pursuit of extending the applicability of inverse kinematics to the entirety of the robotic fleet, a project has been undertaken to create a modular motion controller tailored for the CRF. This innovative motion controller encompasses several enhancements, including the incorporation of task pose objects, establishment of URDF definitions for all robotic entities, automated generation of both inverse and direct kinematics from URDF files, a dedicated visualizer for URDF files, and the provision of well-defined interfaces compatible with various control architectures. The finalized motion controller has undergone deployment across the entire robotic fleet at CERN. Its implementation not only streamlines the availability of inverse kinematics but also fosters efficient and seamless prototyping and testing of new platforms within the research environment. This initiative stands as a significant stride towards standardizing and enhancing the control capabilities of the robotic systems at CERN.

The development phases, including mechanical design, manufacturing, assembly, and installation of the initial prototype, yielded invaluable insights essential for advancing towards a fully tailored robotic solution for the FCC. This marked a pioneering moment as the MRO section successfully introduced a wholly in-house designed robotic arm. Notably, we navigated key challenges related to electromagnetic compatibility (EMC), cable management, and the integration of motor drivers

from diverse manufacturers. In the initial testing phase, the robot demonstrated the effectiveness of our toolchain. The results obtained from simulation exercises, particularly in areas such as elasticity, motor torques, and oscillations, proved remarkably accurate, affirming the robustness of our design and development approach.

In summary, this Ph.D. thesis represents a significant leap toward the goal of crafting an optimized robotic system tailored specifically for the Future Circular Collider. The initial study as well as the novel design optimization and kinematic advancements showcased, provide an insight into the immense potential of robotics for particle accelerators in terms of machine maintenance, availability and downtime. As this research continues to evolve, it stands ready to make profound contributions to the success of the FCC project, pushing the boundaries of what can be achieved with current approaches as applied to the LHC.

Continuing on, the ongoing challenges and future work are outlined in the work package detailed in Appendix A.3. In brief, the initial survey phase is now being elevated to a higher level, with a search for laboratory space underway at CERN. This space will serve as the testing ground for building and integrating the system into a mockup of the FCC tunnel, facilitating proof-of-concept studies in collaboration with all involved sections.

Additionally, the design optimization algorithm will be improved for further enhancement to include the capability to prune joint axes that do not align with the system's requirements. The inverse kinematics algorithm will undergo refinement to handle cyclic trajectories and will be extended through optimization solvers like quadratic programming. This extension will incorporate inequality constraints, enabling precise robot positioning within confined workspaces. The motion controller is heavily used for the entire robotic fleet and undergoes constant upgrades with additional implemented features. All future advancements will be shared with the broader scientific community as an open-source resource.

The operator interface will see improvements, incorporating augmented reality (AR) interfaces and undergoing rigorous testing within the tunnel environment, including enhancements in adaptive stiffness control for haptic feedback control.

Moreover, the robot will be extended by adding joints 1 and 2, along with incorporating a tool changer system and gripper for improved versatility. Additionally, the integration of various sensors and cameras is underway, allowing to seamlessly integrate the robot into an AR interface. This expansion will enable the robot to satisfy all requirements found in previous chapters and will therefor be installed in the 50 m long mock up of the FCC tunnel for proof of concept studies.

Collaborations established during this thesis with Johannes Kepler University, École polytechnique fédérale de Lausanne, Italian Institute of Technology, Università di Napoli Federico II, Universidad Jaume I and Politécnica de Madrid will be ongoing with several students set to join the project, contributing their expertise to the ongoing development of the FCC robotic system. This collaborative effort reflects the enduring commitment to advancing the boundaries of robotics and particle collider technology, ensuring a promising future for the FCC project and its associated innovations.

Bibliography

- [1] A. C. Rolfe. Remote Handling JET Experience, 1999. ISSN 0140-4067.
- [2] D. McGarrigle, E. Flynn, C. Kennedy, A. Loving, and S. Budden. Analysis of Existing and Proposed Maintenance Deployment Systems Toward DEMO MPD Development. *IEEE Transactions on Plasma Science*, 11 2022. ISSN 19399375. doi: 10.1109/TPS.2022.3163466.
- [3] B. Graves and M. Dayton. Core Vessel Insert Handling Robot for the Spallation Neutron Source. 1 2011. URL <https://www.osti.gov/biblio/1042830>.
- [4] E. Boman and L. Smisovsky. Remote Handling within the Active Cells Facility at the European Spallation Source, Using Digital Reality Techniques, 2016. lunduniversity.lu.se/lup/publication/8872915.
- [5] The European Organization for Nuclear Research. URL <https://www.home.cern/>.
- [6] K. Hanke. Past and Present Operation of the CERN PS Booster. *International Journal of Modern Physics A*, 28(13):1330019, 2013. doi: 10.1142/S0217751X13300196. URL <https://doi.org/10.1142/S0217751X13300196>.
- [7] B. de Raad. The CERN SPS Proton-Antiproton Collider. *IEEE Transactions on Nuclear Science*, 32(5):1650–1652, 1985. doi: 10.1109/TNS.1985.4333680.
- [8] S. Myers. The Large Hadron Collider 2008–2013. *International Journal of Modern Physics A*, 28(25):1330035, 2013. doi: 10.1142/S0217751X13300354. URL <https://doi.org/10.1142/S0217751X13300354>.
- [9] CDR. FCC-hh: The Hadron Collider. *The European Physical Journal Special Topics*, 228(4):755–1107, Jul 2019. ISSN 1951-6401. doi: 10.1140/epjst/e2019-900087-0. URL <https://doi.org/10.1140/epjst/e2019-900087-0>.
- [10] J. Heron, L. Felsberger, C. Hernalsteens, C. Wiesner, J. Uythoven D. Wollmann, and F. Rodriguez Mateos. Machine Protection and Availability in the FCC-ee. *Journal of Operations Management*, 2023. URL <https://cds.cern.ch/record/2880188>.
- [11] M. Di Castro, M. Ferre, and A. Masi. CERNTAURO: A modular architecture for robotic inspection and telemanipulation in harsh and Semi-Structured environments. *IEEE Access*, 6:37506–37522, 6 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2849572.

- [12] L. Attard, C. J. Debono, G. Valentino, and M. Di Castro. Tunnel inspection using photogrammetric techniques and image processing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing* 144, 2018.
- [13] Eva Barbara Holzer, Bernd Dehning, Ewald Effinger, Jonathan Emery, Viatcheslav Grishin, Csaba Hajdu, Stephen Jackson, Christoph Kurfuerst, Aurelien Marsili, Marek Miśowiec, Markus Nagel, Eduardo Nebot Del Busto, Annika Nordt, Chris Roderick, Mariusz Sapinski, and Christos Zamantzas. Beam Loss Monitoring for LHC Machine Protection. volume 37, pages 2055–2062. Elsevier B.V., 2012. doi: 10.1016/j.phpro.2012.04.110.
- [14] M. Di Castro, M. L. Baiguera Tambutti, M. Ferre, R. Losito, G. Lunghi, and A. Masi. I-TIM: A Robotic System for Safety, Measurements, Inspection and Maintenance in Harsh Environments. Institute of Electrical and Electronics Engineers Inc., 9 2018. ISBN 9781538655726. doi: 10.1109/SSRR.2018.8468661.
- [15] K. A. Szczurek, R. M. Prades, E. Matheson, J. Rodriguez-Nogueira, and M. Di Castro. Multimodal Multi-User Mixed Reality Human–Robot Interface for Remote Operations in Hazardous Environments. *IEEE Access* 11, 2023.
- [16] CDR. FCC-ee: The Lepton Collider. *The European Physical Journal Special Topics*, 228(2):261–623, Jun 2019. ISSN 1951-6401. doi: 10.1140/epjst/e2019-900045-4. URL <https://doi.org/10.1140/epjst/e2019-900045-4>.
- [17] European Strategy Group. 2020 Update of the European Strategy for Particle Physics. *CERN-ESU-015 (2020)*. cds.cern.ch/record/2721370.
- [18] I. Agapov, M. Benedikt, A. Blondel, M. Boscolo, O. Brunner, M. Chamizo Llatas, T. Charles, D. Denisov, W. Fischer, E. Gianfelice-Wendt, J. Gutleber, P. Janot, M. Koratzinos, R. Losito, S. Nagaitsev, K. Oide, T. Raubenheimer, R. Rimmer, J. Seeman, D. Shatilov, V. Shiltsev, M. Sullivan, U. Wienands, and F. Zimmermann. Future Circular Lepton Collider FCC-ee: Overview and Status. 2022. cds.cern.ch/record/2805006.
- [19] Austrian Doctoral Student Program. URL <https://careers.cern/special-programmes>.
- [20] FCC Study. URL <https://fcc.web.cern.ch/overview>.
- [21] Ecole Polytechnique Federale de Lausanne. URL <https://www.epfl.ch/en/>.
- [22] Universidad Politecnica de Madrid. URL <https://www.upm.es/internacional>.
- [23] H. Gamper, H. Gattringer, A. Müller, and M. Di Castro. Design Optimization of a Manipulator for CERN’s Future Circular Collider (FCC). In *Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO*, pages 320–329. INSTICC, SciTePress, 2021. ISBN 978-989-758-522-7. doi: 10.5220/0010601803200329.
- [24] H. Gamper, A. Mueller, and M. Di Castro. A Robotic Concept for Automated Tunnel Maintenance in the Future Circular Collider. *IEEE Robotics and Automation Magazine*, 2021. (under review).
- [25] H. Gamper, M. Di Castro, H. Gattringer, and A. Mueller. A Robotic System for Remote Interventions in the FCC Complex. 2021. URL <https://indico.cern.ch/event/995850/contributions/4405716/>.

- [26] H. Gamper, D. Forkel, A. Díaz Rosales, J. Playán Garai, C. Veiga Almagro, L. Rosario Buonocore, E. Matheson, and M. Di Castro. Adaptive Radiation Survey Using an Autonomous Robot Executing LiDAR Scans in the Large Hadron Collider. In *Robotics Research*, pages 295–303. Springer Nature Switzerland, 2023. ISBN 978-3-031-25555-7.
- [27] L. Gargiulo, P. Bayetti, V. Bruno, J. C. Hatchressian, C. Hernandez, M. Houry, D. Keller, J. P. Martins, Y. Measson, Y. Perrot, and F. Samaille. Operation of an ITER relevant inspection robot on Tore Supra tokamak. *Fusion Engineering and Design*, 84:220–223, 6 2009. ISSN 09203796. doi: 10.1016/j.fusengdes.2008.11.043.
- [28] H. Wang, W. Chen, Y. Lai, and T. He. Trajectory planning of tokamak flexible in-vessel inspection robot. *Fusion Engineering and Design*, 98-99:1678–1682, 10 2015. ISSN 09203796. doi: 10.1016/j.fusengdes.2015.05.028.
- [29] M. Nakahira, Y. Matsumoto, S. Kakudate, N. Takeda, K. Shibamura, and A. Tesini. Design progress of the ITER blanket remote handling equipment. *Fusion Engineering and Design*, 84:1394–1398, 6 2009. ISSN 09203796. doi: 10.1016/j.fusengdes.2009.03.006.
- [30] B. Humann. FCC-ee Radiation Challenges and Mitigation Measures. *CERN Document Server - CERN-THESIS-2023-236*, 2023. URL <https://cds.cern.ch/record/2879400?ln=de>.
- [31] L. Attard, C. J. Debono, G. Valentino, and M. Di Castro. Vision-Based Tunnel Lining Health Monitoring via Bi-Temporal Image Comparison and Decision-Level Fusion of Change Maps. *Sensors (Basel, Switzerland)*, 21, 6 2021. ISSN 14248220. doi: 10.3390/s21124040.
- [32] W. Niewiem, D. Mergelkuhl, and J.-C. Gayde. Geodetic deformation measurement and analysis of the ATLAS experimental cavern at CERN. *Journal of Instrumentation*, 17(11):P11041, nov 2022. doi: 10.1088/1748-0221/17/11/P11041. URL <https://dx.doi.org/10.1088/1748-0221/17/11/P11041>.
- [33] J.M. Jimenez. LHC: The world’s largest vacuum systems being operated at CERN. *Vacuum*, 84(1):2–7, 2009. ISSN 0042-207X. doi: <https://doi.org/10.1016/j.vacuum.2009.05.015>. URL <https://www.sciencedirect.com/science/article/pii/S0042207X09002887>.
- [34] M. Bajko, F. Bertinelli, N. C. Lasheras, S. Claudet, P. Cruikshank, K. Dahlerup-Petersen, R. Schmidt, A. Siemko, P. Strubin, L. Taviani, H. Thiesen, J. Tock, E. Todesco, R. Veness, A. Verweij, L. Walckiers, R. van Weelden, and R. Wolf. Report of the Task Force on the Incident of 19 September 2008 at the LHC. *LHC Project Report 1168*, Mar 2009.
- [35] P. Bestmann. The Control of the LHC Alignment using a Robot. cds.cern.ch/record/1119514.
- [36] D. Missiaen, J. P. Quesnel, and R. J. Steinhagen. The Alignment of the LHC. *The 10th International Workshop on Accelerator Alignment*, 2008. slac.stanford.edu/econf/C0802113/.
- [37] G. Valentino, R. Aßmann, R. Bruce, S. Redaelli, A. Rossi, N. Sammut, and D. Wollmann. Semiautomatic beam-based LHC collimator alignment. *Physical Review Special Topics - Accelerators and Beams*, 15, 5 2012. ISSN 10984402. doi: 10.1103/PhysRevSTAB.15.051002.

- [38] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer Verlag, 2004.
- [39] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics - Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing series. Springer Verlag, 2009.
- [40] D. Khudher and R. Powell. Quadratic programming for inverse kinematics control of a hexapod robot with inequality constraints. Institute of Electrical and Electronics Engineers Inc., 4 2017. ISBN 9781509033423. doi: 10.1109/RCTFC.2016.7893402.
- [41] W. Suleiman. Regular Paper On Inverse Kinematics with Inequality Constraints: New Insights Into Minimum Jerk Trajectory Generation. *Advanced Robotics*, pages 1–15, 2016.
- [42] A. Rocchi, E. M. Hoffman, D. G. Caldwell, and N. G. Tsagarakis. OpenSoT: A whole-body control library for the compliant humanoid robot COMAN. volume 2015-June, pages 6248–6253. Institute of Electrical and Electronics Engineers Inc., 6 2015. ISBN 9781479969234. doi: 10.1109/ICRA.2015.7140076.
- [43] B. Siciliano and J.-J.E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. pages 1211–1216 vol.2. Institute of Electrical and Electronics Engineers (IEEE), 12 2002. doi: 10.1109/icar.1991.240390.
- [44] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano. Closed-Loop Inverse Kinematics Schemes for Constrained Redundant Manipulators with Task Space Augmentation and Task Priority Strategy. *The International Journal of Robotics Research*, 10:410–425, 1991. ISSN 17413176. doi: 10.1177/027836499101000409.
- [45] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf. Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results. *Frontiers Robotics AI*, 3, 4 2016. ISSN 22969144. doi: 10.3389/frobt.2016.00016.
- [46] M. H. Arbo and J. T. Gravdahl. Stability of the Tracking Problem with Task-Priority Inverse Kinematics. volume 51, pages 121–125. Elsevier B.V., 1 2018. doi: 10.1016/j.ifacol.2018.11.528.
- [47] B. Siciliano and O. Khatib. *Handbook of Robotics*. Springer Handbook of Robotics. Springer Berlin Heidelberg, 2008. ISBN 9783540239574. URL <https://books.google.ch/books?id=Xpgi5gSuBxsC>.
- [48] B. Kenwright. Inverse Kinematics – Cyclic Coordinate Descent (CCD). *Journal of Graphics Tools*, 16(4):177–217, 2012. doi: 10.1080/2165347X.2013.823362. URL <https://doi.org/10.1080/2165347X.2013.823362>.
- [49] P. Beeson and B. Ames. TRAC-IK: An open-source library for improved solving of generic inverse kinematics. volume 2015-December, pages 928–935. IEEE Computer Society, 12 2015. ISBN 9781479968855. doi: 10.1109/HUMANOIDS.2015.7363472.
- [50] L. Unzueta, M. Peinado, R. Boulic, and A. Suescun. Full-body performance animation with Sequential Inverse Kinematics. *Graphical Models*, 70(5):87–104, 2008. ISSN 1524-0703. doi: <https://doi.org/10.1016/j.gmod.2008.03.002>. URL <https://www.sciencedirect.com/science/article/pii/S1524070308000040>.

- [51] A. Aristidou and J. Lasenby. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models*, 73:243–260, 9 2011. ISSN 15240703. doi: 10.1016/j.gmod.2011.05.003.
- [52] R. W. Sumner, M. Zwicker, C. Gotsman, and J. Popović. Mesh-Based Inverse Kinematics. *ACM Trans. Graph.*, 24(3):488–495, jul 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073218. URL <https://doi.org/10.1145/1073204.1073218>.
- [53] N. Courty and É. Arnaud. Sequential Monte Carlo Inverse Kinematics. Research Report RR-6426, INRIA, December 2007. URL <https://inria.hal.science/inria-00194947>.
- [54] J. Demby, Y. Gao, and G. N. DeSouza. A Study on Solving the Inverse Kinematics of Serial Robots using Artificial Neural Network and Fuzzy Neural Network. In *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, 2019. doi: 10.1109/FUZZ-IEEE.2019.8858872.
- [55] A. N. Pechev. Inverse Kinematics without matrix inversion. In *2008 IEEE International Conference on Robotics and Automation*, pages 2005–2012, 2008. doi: 10.1109/ROBOT.2008.4543501.
- [56] A. C. Nearchou. Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm. *Mechanism and Machine Theory*, 33(3):273–292, 1998. ISSN 0094-114X. doi: [https://doi.org/10.1016/S0094-114X\(97\)00034-7](https://doi.org/10.1016/S0094-114X(97)00034-7). URL <https://www.sciencedirect.com/science/article/pii/S0094114X97000347>.
- [57] M. Ayyıldız and K. Çetinkaya. Comparison of four different heuristic optimization algorithms for the inverse kinematics solution of a real 4-DOF serial robot manipulator. *Neural Computing and Applications*, 27:825–836, 5 2016. ISSN 09410643. doi: 10.1007/s00521-015-1898-8.
- [58] A. Colome and C. Torras. Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements. volume 20, pages 944–955. Institute of Electrical and Electronics Engineers Inc., 4 2015. doi: 10.1109/TMECH.2014.2326304.
- [59] H. Gatringer. *Starr-elastische Robotersysteme: Theorie und Anwendungen*. Springer Verlag, 2011.
- [60] H. Bremer. *Elastic Multibody Dynamics: A Direct Ritz Approach*. Springer Verlag, Heidelberg, 2008.
- [61] H. Gamper, A. Luthi, H. Gatringer, A. Mueller, and M. Di Castro. Kinematic Model Pruning: A Design Optimization Technique for Simultaneous Optimization of Topology and Geometry. *Robotics*, 11(2), 2022. ISSN 2218-6581. doi: 10.3390/robotics11020031. URL <https://www.mdpi.com/2218-6581/11/2/31>.
- [62] H. Gamper, A. Luthi, H. Gatringer, A. Mueller, and M. Di Castro. Design Optimization of Quality Inspection Robots for Particle Accelerator Components. ECCOMAS, 2021. ISBN 978-963-421-869-2. URL <https://eccomasmultibody2021.mm.bme.hu/papers.html>.
- [63] A. Macpherson, L. R. Buonocore, M. Di Castro, H. Gamper, and A. Luthi. Camera Placement in a Short Working Distance Optical Inspection System for RF Cavities. *JACoW, SRF2021*:503–507, 2022. doi: 10.18429/JACoW-SRF2021-TUPTEV010.

- [64] Z. M. Bi and W. J. Zhang. Concurrent optimal design of modular robotic configuration. *Journal of Robotic Systems*, 18(2):77–87, 2001. doi: [https://doi.org/10.1002/1097-4563\(200102\)18:2<77::AID-ROB1007>3.0.CO;2-A](https://doi.org/10.1002/1097-4563(200102)18:2<77::AID-ROB1007>3.0.CO;2-A).
- [65] E.J. Van Henten, D.A. Van’t Slot, C.W.J. Hol, and L.G. Van Willigenburg. Optimal manipulator design for a cucumber harvesting robot. *Computers and Electronics in Agriculture*, 65(2):247–257, 2009. ISSN 0168-1699. doi: <https://doi.org/10.1016/j.compag.2008.11.004>. URL <https://www.sciencedirect.com/science/article/pii/S0168169908002238>.
- [66] M. Schappeler and T. Ortmaier. Dimensional Synthesis of Parallel Robots: Unified Kinematics and Dynamics using Full Kinematic Constraints. 2020. doi: 10.17185/dupublico/71211.
- [67] D. A. Rodriguez. Automatic Generation of Task-specific Serial Mechanisms using Combined Structural and Dimensional Synthesis. *Gottfried Wilhelm Leibniz University Hannover*, 2018. doi: 10.15488/4571.
- [68] CERN. Accelerating: Radiofrequency cavities. home.cern/science/..., 2021-12-2.
- [69] K. Watanabe. Review of optical inspection methods. *Insight: Non-Destructive Testing and Condition Monitoring*, 38(4):123–128, 2009. ISSN 13542575.
- [70] M. Wenskat. Automated optical inspection and image analysis of superconducting radio-frequency cavities. *Journal of Instrumentation*, 12(5), 2017. ISSN 17480221. doi: 10.1088/1748-0221/12/05/P05016.
- [71] M. Lemke, E. Elsen, S. Aderhold, U. Cornett, G. Falley, S. Karstensen, T. Külper, A. Navitski, J. Schaffran, F. Schlender, L. Steder, and M. Wenskat. OBACHT - Optical Bench for Automated Cavity Inspection with High Resolution on Short Time Scales. pages ILC–HiGrade–Report–2013–001, DESY–2014–01355, 2013.
- [72] I. Yoshihisa, T. Yujiro, and H. Hitoshi. Development of High Resolution Camera for Observations of Superconducting Cavities. 2008. URL <https://www.ros.org>.
- [73] A. Luthi, A. Macpherson, L. Buonocore, H. Gamper, and M. Di Castro. Camera Placement in a Short Working Distance Optical Inspection System for RF Cavities. International Conference on RF Superconductivity, 2021. URL <https://indico.frib.msu.edu/event/38/>.
- [74] K. Nait-Chabane, P. Hoppenot, and E. Colle. Directional Manipulability for Motion Coordination of an Assistive Mobile Arm. In *ICINCO*, Angers, France, May 2007. URL <https://hal.archives-ouvertes.fr/hal-00341319>.
- [75] R. H. Byrd, M. E. Hribar, and J. Nocedal. An Interior Point Algorithm for Large-Scale Nonlinear Programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.
- [76] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Marti. Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization. *INFORMS Journal on Computing*, 19:328–340, 08 2007. doi: 10.2139/ssrn.886559.
- [77] H. Gamper, H. Gattringer, A. Müller, and M. Di Castro. Design Optimization of a Manipulator for CERN’s Future Circular Collider (FCC). In *Proc. of the 18th Int. Conf. on Informatics in Control, Automation and Robotics - ICINCO*, pages 320–329. INSTICC, SciTePress, 2021. ISBN 978-989-758-522-7. doi: 10.5220/0010601803200329.

- [78] CERN. Future Circular Collider (FCC). home.cern/science/..., 2021-12-2.
- [79] The MathWorks Inc. MATLAB. 2019. URL <https://www.matlab.org>.
- [80] H. Gamper, L. Rodrigo Pérez, A. Mueller, and A. Díaz Rosales. An Inverse Kinematics Algorithm with Smooth Task Switching for Redundant Robots. *IEEE Robotics and Automation Letters*, 2024.
- [81] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir. Inverse Kinematics Techniques in Computer Graphics: A Survey. *Computer Graphics Forum*, 37:35–58, 9 2018. ISSN 14678659. doi: 10.1111/cgf.13310.
- [82] R. Singh, V. Kukshal, and V. Yadav. *A Review on Forward and Inverse Kinematics of Classical Serial Manipulators*, pages 417–428. 02 2021. ISBN 978-981-33-4017-6. doi: 10.1007/978-981-33-4018-3_39.
- [83] G. Antonelli. Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. *IEEE Transactions on Robotics*, 25:985–994, 2009. ISSN 15523098. doi: 10.1109/TRO.2009.2017135.
- [84] P. Falco and C. Natale. On the stability of closed-loop inverse kinematics algorithms for redundant robots. *IEEE Transactions on Robotics*, 27:780–784, 8 2011. ISSN 15523098. doi: 10.1109/TRO.2011.2135210.
- [85] G. Guennebaud and B. Jacob. Eigen v3. 2010. URL <http://eigen.tuxfamily.org>.
- [86] I. Sucas and T. Moulard. URDF DOM Development Files. URL <https://github.com/ros/urdfdom>.
- [87] A. Partow. C++ Mathematical Expression Toolkit Library (ExprTk). URL <https://www.partow.net/programming/exprtk/index.html>.
- [88] The Linux Foundation. Real-Time Linux Preempt RT. URL <https://wiki.linuxfoundation.org/realtime/start>.
- [89] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76:38–47, 2018. ISSN 0262-8856. doi: <https://doi.org/10.1016/j.imavis.2018.05.004>. URL <https://www.sciencedirect.com/science/article/pii/S0262885618300799>.
- [90] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and R. Medina-Carnicer. Generation of fiducial marker dictionaries using Mixed Integer Linear Programming. *Pattern Recognition*, 51:481–491, 2016. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2015.09.023>. URL <https://www.sciencedirect.com/science/article/pii/S0031320315003544>.
- [91] S. Chacon and B. Straub. *Pro Git*. Apress, USA, 2nd edition, 2014. ISBN 1484200772.
- [92] K. Martin and B. Hoffman. Mastering CMake: A Cross-Platform Build System. Apress, 2014. URL <https://cmake.org/>.
- [93] GNU make. *GNU*, 2014. URL <https://www.gnu.org/>.
- [94] Google C++ Style Guide. *Google Inc.*, 2017. URL <https://google.github.io/styleguide/cppguide.html>.

- [95] Google's C++ Linter. *Google Inc.*, 2023. URL <https://github.com/cpplint/cpplint>.
- [96] Valgrind. *Valgrind Developers*, 2023. URL <https://valgrind.org/>.
- [97] Markdown Guide. *J. Gruber and A. Swartz*, 2023. URL <https://www.markdownguide.org/>.
- [98] GitLab. *GitLab B.V.*, 2023. URL <https://about.gitlab.com/>.
- [99] Docker. *Docker Inc.*, 2023. URL <https://www.docker.com/>.
- [100] Ubuntu Canonical. *Canonical Ltd. Ubuntu*, 2023. URL <https://ubuntu.com/>.
- [101] Installation of a Linux Kernel and corresponding RT Patch. *Stackoverflow*, 2023. stackoverflow.com/questions/51669724/...
- [102] RT-Test Test Suite. *The Linux Foundation*, 2023. wiki.linuxfoundation.org/realtime/...
- [103] VxWorks. *Wind River Systems, Inc.*, 2023. URL <https://www.windriver.com/>.
- [104] A. Barbalace, A. Luchetta, G. Manduchi, M. Moro, A. Soppelsa, and C. Taliencio. Performance Comparison of VxWorks, Linux, RTAI and Xenomai in a Hard Real-time Application. In *2007 15th IEEE-NPSS Real-Time Conference*, pages 1–5, 2007. doi: 10.1109/RTC.2007.4382787.
- [105] JSON. *ISO/IEC 21778:2017*, 2023. URL <https://www.json.org/json-en.html>.
- [106] Eigen Library. *tuxfamily.org*, 2023. URL <https://gitlab.com/libeigen/eigen>.
- [107] Mozilla Public License. *Mozilla Foundation*, 2023. URL <https://www.mozilla.org/en-US/MPL/>.
- [108] json Library. *github.com*, 2023. URL <https://github.com/nlohmann/json>.
- [109] MIT License. *Amazon Web Services*, 2023. URL <https://github.com/aws/mit-0>.
- [110] urdfdom Library. *github.com*, 2023. URL <https://github.com/ros/urdfdom>.
- [111] BSD License. *Willow Garage In.*, 2023. URL <https://www.linfo.org/bsdlicense.html>.
- [112] GNU Lesser General Public License. *GNU*, 2023. URL <https://www.gnu.org/licenses/lgpl-3.0.de.html>.
- [113] A. Díaz Rosales, M. Di Castro, and H. Gamper. Control Design Optimisations of Robots for the Maintenance and Inspection of Particle Accelerators. In *Proc. 19th Int. Conf. Accel. Large Exp. Phys. Control Syst. (ICALEPCS'23)*, number 19 in International Conference on Accelerator and Large Experimental Physics Control Systems, pages 153–159. JACoW Publishing, Geneva, Switzerland, 2024. ISBN 978-3-95450-238-7. doi: 10.18429/JACoW-ICALEPCS2023-MO3AO07. URL <https://jacow.org/icalepcs2023/papers/mo3ao07.pdf>.
- [114] Waterloo Maple Inc. Maple. 2022. URL <https://maplesoft.com>.
- [115] Autodesk Inc. Inventor. 2022. URL <https://www.autodesk.de/products/inventor>.

Appendix

A.1 Selected UML Diagrams of the CRF

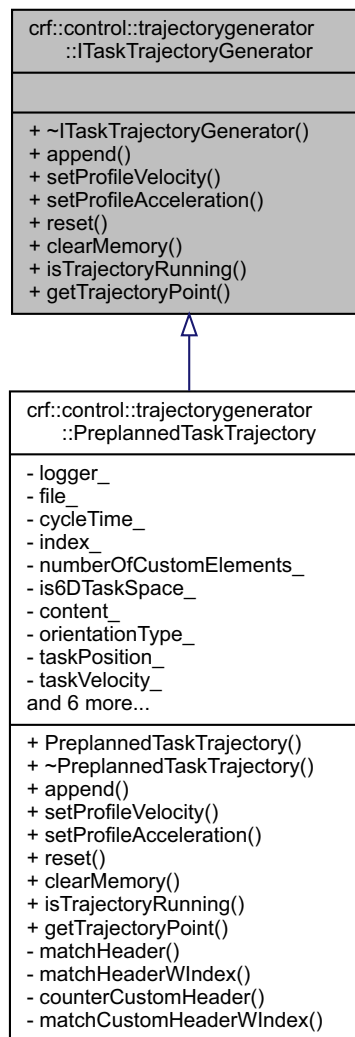


Figure A.1: UML diagram of the ITaskTrajectoryGenerator interface.

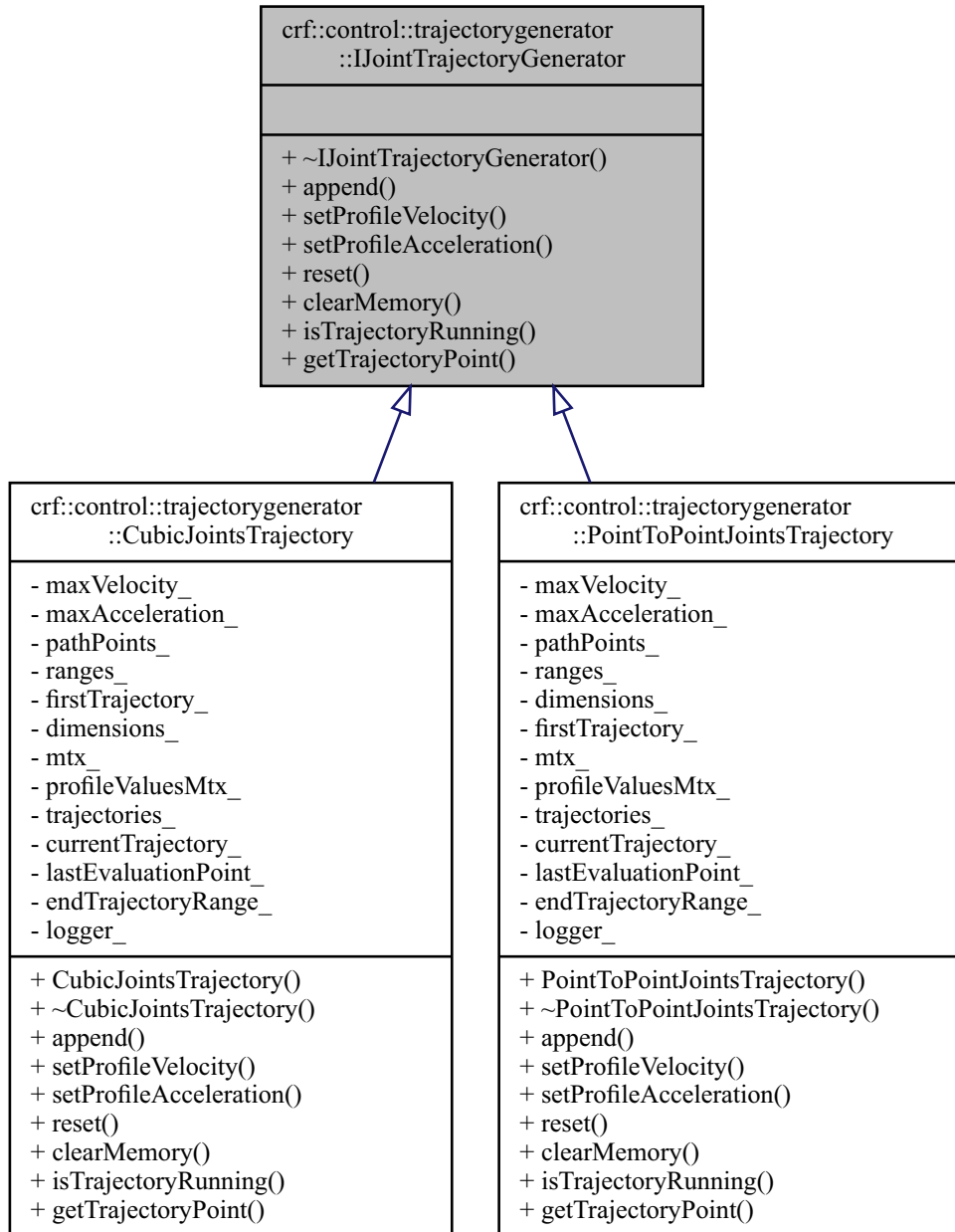


Figure A.2: UML diagram of the `IJointTrajectoryGenerator` interface.

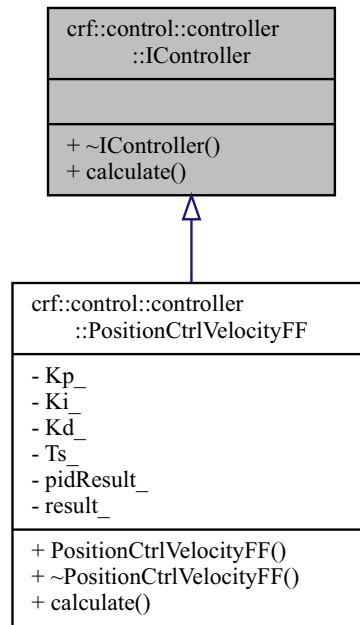


Figure A.3: UML diagram of the IController interface.

A.2 Code Snippets

Listing A.1: Robot configuration file example

```

1 {
2     "IPAddress" : "192.168.1.50",
3     "JointSpaceDegreeOfFreedom" : 6,
4     "TaskSpaceDegreeOfFreedom" : 6,
5     "ControllerLoopTimeMs" : 2,
6     "LookAheadTime" : 0.03,
7     "Gain" : 1200,
8     "JointLimits" : {
9         "MaxPosition" : [6.28, 6.28, 6.28, 6.28, 6.28, 6.28],
10        "MinPosition" : [-6.28, -6.28, -6.28, -6.28, -6.28, -6.28],
11        "MaxVelocity" : [2.09, 2.09, 3.14, 3.14, 3.14, 3.14],
12        "MaxAcceleration" : [3, 3, 3, 3, 3, 3],
13        "MaxTorque" : [1, 1, 1, 1, 1, 1]
14    },
15    "TaskLimits" : {
16        "MaxVelocity" : [0.1, 0.1, 0.1, 0.1, 0.1, 0.1],
17        "MaxAcceleration" : [1, 1, 1, 1, 1, 1]
18    },
19    "ProfileParameters" : {
20        "JointsVelocity" : [1, 1, 1, 1, 1, 1],
21        "JointsAcceleration" : [1.4, 1.4, 1.4, 1.4, 1.4, 1.4],
22        "TaskVelocity" : [0.2, 0.2, 0.2, 0.1, 0.1, 0.1],
23        "TaskAcceleration" : [1, 1, 1, 0.8, 0.8, 0.8]
24    },

```

```

25 "Kinematics" : {
26   "Type" : "MathExpressions",
27   "DH" : {
28     "D" : [0.2755, 0.0, -0.0098, -0.3111, 0.0, 0.2495],
29     "Theta" : [0.0, 1.5708, -1.5708, 0.0, 0.0, 3.1416],
30     "A" : [0.0, 0.4100, 0.0, 0.0, 0.0, 0.0],
31     "Alpha" : [1.5708, 3.1416, 1.5708, 1.5708, 1.5708, 0.
32       0]
33   },
34   "MathExpressions" : {
35     "Lengths" : {
36       "lx" : [0.0, 0.0, 0.6127, 0.57155, 0.0, 0.0],
37       "ly" : [0.0, 0.0, 0.0, 0.0, 0.0, 0.11655],
38       "lz" : [0.0, 0.1807, 0.0, 0.0, 0.17415, 0.11985]
39     },
40     "ForwardKinematics" : {
41       "RIE" : {
42         "Row0" :
43         {
44           "Col0" : "...",
45           "Col1" : "...",
46           "Col2" : "..."
47         },
48       "Row1" :
49       ...
50     },
51     "IrIE" : {
52       "0" : "...",
53       "1" : "...",
54       "2" : "..."
55     },
56     "IwIE" : {
57       "0" : "...",
58       "1" : "...",
59       "2" : "..."
60     },
61     "IvIE" : {
62       "0" : "...",
63       "1" : "...",
64       "2" : "..."
65     },
66     "IalphaIE" : {
67       "0" : "...",
68       "1" : "...",
69       "2" : "..."
70     },
71     "IaIE" : {
72       "0" : "...",
73       "1" : "...",
74       "2" : "..."

```

```
74         }
75     },
76     "Jacobian" : {
77         "Row0" : {
78             "Col0" : "...",
79             "Col1" : "...",
80             "Col2" : "...",
81             "Col3" : "...",
82             "Col4" : "...",
83             "Col5" : "...",
84         },
85         "Row1" :
86             ...
87     }
88 }
89 }
90 }
```

A.3 FCC Robotics Work Package



FCC Project Document No. FCC-XXX-YYYY
CERN Div./Group or Supplier/Contractor Document No. XXX/YYY
EDMS Document No.

<p>Work Package Description</p> <p>FUTURE CIRCULAR COLLIDER</p> <p>TECHNOLOGIES AND R&D WP</p> <p><i>Abstract</i></p> <p>This document describes the FCC Technologies and R&D Work Package X. The following document describes the justification, the objectives, the resources involved and the timeline, including an ORAMS analysis of the impact of the concerned equipment/system on the operation of the FCC-ee machine.</p>		
<p><i>Prepared by :</i></p> <p>Mario Di Castro mario.di.castro@cern.ch Hannes Gamper hannes.gamper@cern.ch Alessandro Masi alessandro.masi@cern.ch Principal Investigator</p>	<p><i>Checked by :</i></p> <ul style="list-style-type: none">• Roberto Losito (CERN)• Group Leaders in charge and/or Responsible of the institute• Leader of the CERN group in charge of reception/operation of the system• Project Safety Officer	<p><i>Approval Leader :</i></p> <p>Michael Benedikt Michael.Benedikt@cern.ch Frank Zimmermann Frank.Zimmermann@cern.ch Tor Raubenheimer Tor@slac.stanford.edu</p>
<p><i>Approval List :</i></p>		

Table of Contents

INTRODUCTION4

WORKPACKAGE DESCRIPTION4

1.1 SHORT WORKPACKAGE DESCRIPTION 4

MANPOWER & BUDGET11

SUMMARY TABLE INSTITUTE A..... 11

SUPPORT, INTEGRATION AND INTERFACES.....12

MANAGEMENT ASPECTS13

COLLABORATIONS 13

EXTERNAL FUNDING..... 13

LABORATORIES AND INFRASTRUCTURE..... 13

RISKS 13

INTRODUCTION

The perception, cognition and control of robotic systems was advancing rapidly in the last decades and is even speeding up, benefiting from increasing computation power and technologies like artificial intelligence. These developments enabled robots to handle more complex tasks with a higher degree of automation. Extrapolating the state of robotic intelligence by about two decades, the time when the FCC Robot would operate in the FCC tunnels, such systems will be able to handle most of the planned interventions that are currently, mainly conducted manually and many of the unplanned or emergency situations. Additionally, most of the manually performed interventions in the LHC complex cannot be applied to the three times longer FCC tunnel, without either increasing efforts in workforce and money or accepting longer machine down times. Thus, a robotic system will decrease machine down time, reduce the radiation exposure of workers and will locate, guide and possibly rescue workers in case of emergencies.

The harsh environment of the FCC tunnel gives rise to very specific requirements for a remote maintenance system and thus do not allow the usage of common industrial robots. This means that customized robotic systems, which are capable of handling the most diverse automation problems under hard restrictions, have to be developed. This furthermore requires R&D in many of the core disciplines in robotics like perception, cognition, localization and mapping, locomotion or motion control, which are shown in detail in the following work package description.

Existing results from a preliminary study concerning a robotic system for the FCC complex ("A Robotic System for Remote Interventions in the FCC Complex" - <https://indico.cern.ch/event/995850/contributions/4405716/>) and a remote maintenance code of practice (EDMS Nr. 2263542) were presented during the FCC Week 2021.

This work package proposes an approach and outline for the development of a robust and fast remote maintenance system (in hardware and software) for inspection and maintenance tasks in CERN's Future Circular Collider (FCC).

WORKPACKAGE DESCRIPTION

1.1 SHORT WORKPACKAGE DESCRIPTION		
WP leader	??	
Collaboration Institutes	JKU, UPM, UJI, EPFL, UNINA, IIT,	
Linkpersons at institutes	H. Gamper	CERN/JKU
	M. Ferre	UPM
	R. Marin	UJI
	V. Lippiello	UNINA
	D. Caldwell	IIT
Scope of Work (description of the tasks to be done)		
1. PROCEDURES AND CODE OF PRACTICE		
1.1. Defining required tasks and intervention procedures		
1.2. Defining Remote Maintenance Code of Practice		
2. INTEGRATION		
2.1. Define rail placement in regular tunnel cross section and parking locations		
2.2. Design radiation safe spaces for parking and robot maintenance		
2.3. Tool storage		
2.4. Design of automated hatches in fire doors		
3. R&D		
3.1. Perception		
3.2. Localization & Mapping		
3.3. Cognition		

- 3.4. Locomotion - Accessibility in Confined Areas
- 3.5. Motion Control
- 3.6. Energy Management
- 3.7. Tool Changer
- 3.8. Operator Interfaces and Training
- 3.9. Proprioception and Haptics
- 3.10. Human-Robot Collaborations
- 3.11. Recovery Scenarios and Emergency Interventions
- 4. PROOF OF CONCEPT ON THE MOCK-UP
 - 4.1. Inspection and Measurements in Full Autonomous Mode
 - 4.2. Teleoperation Mode
 - 4.3. Collaborative Mode
 - 4.4. Dipole Alignment
 - 4.5. Vacuum Leak Detection
 - 4.6. Reconnaissance
- 5. REPORTS & DOCUMENTATION
 - 5.1. Summary of Achievements and Relation with Initial Goals

Objectives (describe for the SoW above how you will measure success)

1. Definition of required tasks, intervention procedures and Remote Maintenance Code of Practice in collaboration with involved sections/parties.
2. Document on integration decisions with all involved sections/parties
3. Increased autonomy of remote maintenance system (measured in increase of efficient high level commands from operator)
4. More intuitive teleoperation interfaces (standardized benchmarks) and thus faster handling of interventions (compared to currently conducted interventions)
5. Increased availability of remote maintenance system itself, when adjusting robots to different tasks (compared to current situation)
6. Increased dexterity to efficiently handle unplanned interventions (compared to current situation)
7. More intuitive human robot collaboration for interventions with human support (standardized benchmark test)
8. Faster availability of in situ robotic support (compared to current situation and estimation for FCC machine)

Objectives of Remote Maintenance in General

1. Increased Availability of FCC
2. Reduced Radiation Exposure of Workers
3. Reduced Logistic Effort for Tunnel Access
4. Increased Safety for Workers in Tunnel
5. Easier Transport of Material
6. Increased Safety in Case of Emergency (Faster Intervention, Faster assessment of the situation, Reduced risk for FireFighters)

Description of work and tasks

This work package covers the development of a remote maintenance system for FCC in terms of software, hardware and control. The developed techniques will be demonstrated with a prototype in laboratory conditions. The laboratories, machines and experiments at CERN require a robotic system to satisfy very specific and hard requirements, which doesn't allow the use of industrial of the shelf solutions. Thus this work package focuses on the development of software, hardware and control algorithms, which in combination with available industrial solutions will allow CERN to deploy a complete, versatile and robust remote handling system. The necessary steps are described below in detail:

TASK 1. PROCEDURES AND CODE OF PRACTICE**Subtask 1.1: Defining required tasks and intervention procedures (BE-CEM)**

Define requirements, restrictions and desired intervention procedures in collaboration with all involved sections at CERN, for the future remote maintenance system.

Subtask 1.2: Defining Remote Maintenance Code of Practice (BE-CEM, EN-MME)

Define a Remote Maintenance Code of Practice in collaboration with all involved sections at CERN, to guarantee efficient interventions. The Code of Practice should contain guidelines for the design process of equipment, intervention procedures, tool definitions and proposals for interfaces, connectors and placement of infrastructure.

TASK 2. INTEGRATION**Subtask 2.1: Define rail placement in regular tunnel cross section and parking locations (BE-CEM, SCE-DOD, EN-ACE)**

Possible placement for the rails of the robotic system in the regular tunnel and the parking locations should be defined and the required space reserved.

Subtask 2.2: Design radiation safe spaces for parking and robot maintenance (BE-CEM, SCE-DOD, EN-ACE)

Radiations safe parking for the robotic units will be defined to protect the system from radiation and to allow maintenance on the robotic system itself, even during operation of the FCC. This is crucial for immediate availability of the remote maintenance system, once the machine is shut down, or in case of emergencies.

Subtask 2.3: Tool Storage (BE-CEM, SCE-DOD, EN-ACE)

Space for tool storage with the most common tools that can be changed automatically should be reserved.

Subtask 2.4: Design of automated hatches in fire doors (BE-CEM, HSE, SCE-DOD, EN-ACE)

For safe and fast interventions automated hatches in the fire doors close to the ceiling need to be installed. The doors should open when the remote maintenance system approaches in a way, such that the system doesn't have to slow down. Afterwards the doors need to be closed again for obvious safety reasons.

TASK 3: R&D**Subtask 3.1: Perception (BE-CEM, UPM)**

In order to facilitate efficient and safe interventions, the robotic system requires a sophisticated system of cameras, Lidars, infrared cameras, radiations sensors, etc. to be aware of its environment at all times and avoid collisions or provide best insight for operators. This requires development in software (sensor fusion, data acquisition, ...) and selection of appropriate hardware.

Subtask 3.2: Localization & Mapping (BE-CEM, UPM)

Localization and mapping are crucial points to know the precise position of the robotic system at any time and map unknown obstacles for collision avoidance. Common methods use inputs like camera images, Lidar data, motor data and other encoders. This requires development in software (odometry, SLAM, ...) and hardware (e.g. encoder system along the tunnel axis,...).

Subtask 3.3: Cognition (BE-CEM, UNINA)

Robotic cognition is currently one of the fastest developing techniques. Classical methods can be used for obstacle avoidance and advanced artificial intelligence algorithms will be used for tunnel health monitoring or localization of people in hazardous situations. These requires

software development for classical methods and training/tuning of networks in field conditions at CERN.

Subtask 3.4: Locomotion - Accessibility in Confined Areas (BE-CEM, IIT)

A rail guided remote maintenance system will be the most efficient and robust version, that can handle most of the required tasks. Nevertheless there will be certain sections of the tunnel (clystron gallery) or situations when a rail guided system cannot operate. It will be necessary to investigate other forms of locomotion, like quadrupeds, humanoids, snakes, etc. to access these complicated zones. Drones might be used to carry out tasks like inspection and support to teleoperation giving different point-of-views of the scenes to the operators and/or to artificial intelligence algorithms.

Subtask 3.5: Motion Control (BE-CEM, JKU)

The hard requirements for a robotic system in terms of dexterity, payload, allowed robot weight, allowed robot space, desired work space are at the limit or beyond technical state of the art. Thus, more sophisticated control algorithms are required to control e.g. light weight structure that are not rigid enough to neglect elasticity and thus the most common non-linear flatness based control methods are not sufficient. In order to provide precise interventions more sophisticated control methods have to be implemented and tested. The various and highly diverse tasks require a manipulator with highest dexterity. This is achieved with highly redundant mechanical structures (above ~10 degree of freedom or "snake" like robots) that often show very sophisticated mechanical designs. The development of the mechanical structure (topology optimization, actuation and motor driver placement, cabling) and the control algorithms (robust redundant inverse kinematics resolution) will be the main challenges that have to be solved.

Subtask 3.6: Energy management (BE-CEM, EPFL?)

Most efficient actuation systems with the highest power densities have to be chosen and evaluated in terms of power consumption and the operation on low voltage battery systems (in case of emergency). A power supply via the rails or embedded in the rails should be investigated to provide enough power for long duration interventions. Furthermore a battery powered operation should be developed (battery management system) to ensure the success of emergency interventions.

Subtask 3.7: Tool Changer (BE-CEM, IIT)

The end effector the robotic manipulator will not be able to hold many different tools, so that small space can still be accessed. For time efficient interventions though, it is crucial to carry the most common tools and provide a way of quick tool changing. The main challenges for a quick tool changing system will lie in the mechanical design.

Subtask 3.8: Operator Interfaces and training (BE-CEM, UJI)

The efficiency and success of interventions that require a human operator will strongly depend on intuitive and robust interfaces that rely on control panels, virtual reality systems and haptic devices for force feedback, in case the robot will physically manipulate targets and not only inspect. The development efforts will be focused on the perception of the robot environment, how to present this information via VR systems, robust teleoperation via haptic devices possibly via 5G and other novel human robot interfaces. In addition, VR systems provide a great way of training operators for future interventions in realistic scenarios.

Subtask 3.9: Proprioception and Haptics (BE-CEM, IIT)

Teleoperation requires proprioception and haptics to give feedbacks to the operators to increase environment awareness reducing mental stress. These features are fundamental for safe and robust teleoperation missions.

Subtask 3.10: Human Robot Collaboration (BE-CEM, JKU)

Human robot collaborations are especially useful when the robot is performing tasks that are impossible for workers like transport of heavy material or precise placement of tools, whereas workers can focus on tasks that require high creativity, physical understanding or dexterity. The main challenges will include the development of emergency shut down techniques and the certification procedures for these algorithms, in addition to safety specifications in human-robots coexistence / cooperation.

Subtask 3.11: Recovery scenarios and emergency interventions (BE-CEM, HSE, UPM)

Procedures to recuperate safely deployed robots must be produced in order to cope with eventual failure scenarios of the robotic systems deployed.

TASK 4: PROOF OF CONCEPT ON THE MOCK-UP

Milestone 4.1: Inspection and Measurements in Full Autonomous Mode (BE-CEM)

In case of e.g. a failure of some kind of infrastructure in the fcc tunnel, a operator should be able to send the robot fully automated to a arbitrary point of interest in the tunnel complex and then take possibly over in teleoperation mode to inspect the problem. This requires development in terms of motion control, trajectory planning, collision avoidance, inverse kinematics, etc.. For this milestone, a robot has to reach an area of the mock-up in an autonomous way reporting environmental measurement (e.g. oxygen %, temperatures, humidity, radiation, tunnel structure status, etc.) to the operator and approaching with a robotic end-effector a beam instrumentation sensor (e.g. BLM). This subtask will need the above R&D developments 3.1, 3.2, 3.3, 3.5, 3.11.

Milestone 4.2: Teleoperation Mode (BE-CEM)

A robust teleoperation mode will be implemented, such that an operator can handle unplanned or chaotic emergency situations. The main challenges will be the human robot interface, in particular haptic control with force feedback either over 5G or with significant network latency times. For this milestone, the developed human-robot interface will be benchmarked with industrial robots interface to measure the efficiency and advancement with respect to the industrial state-of-art robotic interfaces. This subtask will mainly need the above R&D developments 3.3, 3.5, 3.9, 3.10, 3.11.

Milestone 4.3: Collaborative Mode (BE-CEM)

The human robot collaboration will be tested and benchmarked in different scenarios with different complexity. Tests will include transport of material, support for workers with heavy tools, precise placement of devices and emergency shutdown scenarios. This subtask will mainly need the above R&D developments 3.1, 3.2, 3.3, 3.5, 3.7, 3.10, 3.11.

Milestone 4.4: Dipole Alignment (BE-CEM)

For this milestone, 3 robotic end effector integrating on autonomous robots will act simultaneously on the 3 dipoles jack to move the dipoles to a desired position (e.g. vertical motion, yaw and tilt angle). This subtask will mainly need the above R&D developments 3.1, 3.2, 3.3, 3.5, 3.6, 3.9, 3.10, 3.11.

Milestone 4.5: Vacuum Leak Detection (BE-CEM)

For this milestone, a robotic end effector integrated on an autonomous robot will spray helium around vacuum chambers to find possible vacuum leak using also different tools. This subtask will mainly need the above R&D developments 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.9, 3.10, 3.11.

Milestone 4.5: Reconnaissance (BE-CEM)

For this milestone, an autonomous robot will be able to locate and guide people in the mockup, monitor the people health, detect and extinguish fire. This subtask will mainly need the above R&D developments 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11.

TASK 5: REPORTS & DOCUMENTATION

Milestone 5.1: Report on Summary of Achievements and Relation with Initial Goals (BE-CEM,)

The final report will contain a summary of the achievements and a relation with the initial goals. In specific, benchmark tests and test scenarios as well as the results of these tests will be documented. Based on these results the expected improvement of the main objectives (listed under *Objectives*) for FCC will be outlined. Furthermore a state of the art ORAMS analysis for this class of equipment will be performed and included in the final report.

Deliverables	Task	Year
D1: Report on Procedures and Code of Practice	TASK 1	End of Y1
D2: Report on Integration	TASK 2	Mid of Y2
D3: Fully Operational Prototype	TASK 3	Mid of Y3
D4: Report on Validation Proof of Concept on the Mock-Up	TASK 4	End of Y4
D5: Report & Documentation: Summary of Achievements and Relation with Initial Goals	TASK 5	End of Y4



CURRICULUM VITAE

DIPL. ING. HANNES GAMPER BSc

Personal Data

Email hgamper@gmx.at

Phone 0043 660 5641644

Address 60 Rue Blaise-Pascal, 01630 Saint Genis-Pouilly, France

Education

2019 - 2023 ROBOTICS - PhD Studies at Johannes Kepler University, Linz - Austria and CERN - the European Organization for Nuclear Research - Switzerland

2015 - 2019 ROBOTICS - MSc Studies at Johannes Kepler University, Linz - Austria

2015 COMPUTER SCIENCE - MSc Studies at California State Univ. - USA

2011 - 2015 CONTROL DESIGN - BSc Studies at Johannes Kepler Univ., Linz - Austria

2006 - 2011 MECHATRONICS - HTL-Jenbach, Jenbach - Austria

Work Experience

2017 ROBOTICS - Institute of Robotics - Johannes Kepler University, Linz - Austria

2016 SYSTEMS ENGINEERING - NASA's Jet Propulsion Laboratory - USA

2015 SIGNAL PROCESSING - Institute for Design and Control of Mechatronical Systems, JKU - Linz

2010 MECHANICAL ENGINEERING - Bernard Engineers, Hall - Austria

Languages

German First Language, **English** Fluent, **French** Beginner