DISS. ETH NO. 27690

Robust Robotic Aggregation of Irregularly Shaped Objects

Martin Wermelinger

DISS. ETH NO. 27690

Robust Robotic Aggregation of Irregularly Shaped Objects

A dissertation submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZÜRICH
(Dr. sc. ETH Zürich)

presented by

Martin Wermelinger

M.Sc. ETH Mechanical Engineering
born on 23 August 1990
citizen of Hergiswil LU, Switzerland

accepted on the recommendation of

Prof. Dr. Marco Hutter (ETH Zürich), examiner
Prof. Dr. Matthias Kohler (ETH Zürich), co-examiner
Prof. Dr. Nils Napp (Cornell University), co-examiner
Prof. Dr. Melanie Zeilinger (ETH Zürich), co-examiner

2021

Robotic Systems Lab
Institute for Robotics and Intelligent Systems
ETH Zürich
Switzerland

Abstract

Robotic technology has entered many new fields and applications in the last years and decades, from automated manufacturing, over warehouse logistics, to inspection of industrial sites. While automation and robotics are pushing more and more from industrial to everyday applications, one sector has fallen behind in adapting to those new technologies, the construction sector. Although there is a labor shortage in construction, it still has one of the most considerable unused potentials for automation. Exploiting on-site robotics is not only a great opportunity to resolve the high demand for labor and to increase productivity, but it also has the potential to enable architectural designs that exceed the size and complexity practical with conventional methods. It also offers the opportunity to leverage context-specific, locally sourced materials that are inexpensive, abundant, and low in embodied energy. However, it is still unclear what exact technologies are required and how such robot solutions would potentially look like at this stage.

This thesis addresses the development of manipulation skills for a mobile manipulator to detect and assemble arbitrary solid material in a cluttered and unstructured environment like a construction site. The focus lies on the perception of the environment, modeling object instances, and grasping and assembling objects. We are especially interested in manipulating raw material, like stones and boulders, in the context of robotic landscaping, as it targets applications in dangerous-to-access or remote locations undesired for human operators.

In this work, we investigate the task of executing autonomous missions in unseen environments at different scales and from stationary to mobile applications. Besides localization, obstacle detection, and navigation methods for the mobile base, we introduce a compliant manipulator to perform interaction tasks. The compliance gives impact robustness and the capability of controlling contact wrenches accurately but comes with limited actuator bandwidth causing tracking performance loss. We address this problem by including the actuator dynamics in a receding horizon control formulation improving tracking and disturbance rejection and show that the natural stiffness of the manipulator can be preserved.

The ultimate target of this work is the construction of large-scale structures in real-world outdoor scenarios, outside of well-defined laboratory settings, complicating the handling of previously unseen irregularly shaped objects. We contribute by presenting a perception and grasp pose planning pipeline for autonomous manipulation of objects of interest with a robotic walking excavator. A mapping system incrementally builds a temporally and spatially consistent LiDAR-based map of the robot's surroundings. It provides the ability to register externally reconstructed point clouds of the scene, e.g., from images captured by a drone-borne camera, helping to increase map coverage. Our grasp planning method utilizes point clouds and mesh surface reconstruction of stones to plan grasp configurations with a 2-jaw gripper mounted on the excavator. Besides taking into account the geometry of the stone to sample force closure grasps, the grasp planner considers collision constraints during object pick and place, informed by the LiDAR-based point cloud map. Furthermore, we show an approach to reorient arbitrarily shaped objects that cannot be directly placed at the desired location without violating collision constraints.

Finally, the presented manipulation methods are combined with geometric target pose planning to compute structurally stable object compositions following a target design. We create a novel process that takes the digitized object model in a physics simulation to find settled placement poses and assess stability, alternately with a 3D shape matching to the target geometry. The locations of the placed objects are refined and updated in simulation to have an accurate digital twin of the scene. Our approach has been thoroughly validated on several interdisciplinary collaborations building vertical stone towers in a table-top setup and dry-stone walls composed of over one hundred stones with an average weight above 1000 kg with the autonomous excavator HEAP.

Zusammenfassung

Die Robotertechnologie hat in den letzten Jahrzehnten viele neue Bereiche und Anwendungen erobert, von der automatisierten Fertigung über die Lagerlogistik bis hin zur Inspektion von Industrieanlagen. Während sich Automatisierung und Robotik stetig von industriellen zu alltäglichen Anwendungen hinbewegen, ist ein Sektor bei der Anpassung an diese neuen Technologien ins Hintertreffen geraten: der Bausektor. Obwohl im Baugewerbe ein Arbeitskräftemangel herrscht, gibt es hier noch ein grosses ungenutztes Potenzial zur Automatisierung. Die Nutzung von Robotern in situ ist nicht nur eine Möglichkeit, den hohen Bedarf an Arbeitskräften zu decken und die Produktivität zu steigern, sondern erlaubt auch architektonische Designs, die in Grösse und Komplexität konventionellen Methoden übersteigen. Es ermöglicht kontextspezifische, lokale Materialien zu nutzen, die kostengünstig und reichlich vorhanden sind und einen geringen Anteil an grauer Energie aufweisen. Allerdings ist zum jetzigen Zeitpunkt noch unklar, welche Technologien genau benötigt werden und wie eine solche Roboterlösungen aussehen sollte.

Diese Doktorarbeit befasst sich mit der Entwicklung von Manipulations-Fähigkeiten für einen mobilen Manipulator, um beliebiges festes Material in einer unstrukturierten Umgebung, wie einer Baustelle, zu erkennen und zusammenzufügen. Der Fokus liegt dabei auf der Wahrnehmung der Umgebung, der Modellierung von Objekten und dem Greifen und Platzieren. Wir sind besonders an der Manipulation von rohem Material, wie Steinen und Felsblöcken, im Kontext von Landschaftsgestaltung interessiert, da es Anwendungen an gefährlichen und schwer zugänglichen Orten anvisiert welche die manuelle Bedienung der Maschine erschweren.

In dieser Arbeit stellen wir zunächst einen mobilen Roboter vor, der in der Lage ist, autonome Missionen in unbekannten Umgebungen auszuführen. Neben Methoden zur Lokalisierung, Hinderniserkennung und Navigation für die mobile Basis, präsentieren wir einen nachgiebigen Manipulator, um Interaktions-Aufgaben durchzuführen. Die nachgiebigen Aktuatoren verleihen dem Manipulator Robustheit gegenüber Stössen und die Fähigkeit Kontaktkräfte und -momente genau zu

regeln. Aber die Nachgiebigkeit geht mit einer begrenzten Regelbandbreite einher, was zu einem Spurfehler führt. Wir behandeln dieses Problem, indem wir die Aktuator-Dynamik in eine modellprädiktive Regelung mit einbeziehen. Wir können dadurch den Spurfehler und das Verhalten auf Schläge verbessern, und zeigen, dass die natürliche Steifigkeit des Manipulators gleichzeitig erhalten bleibt.

Das Ziel dieser Arbeit ist die Konstruktion von massiven Strukturen in realen Umgebungen, ausserhalb einer wohldefinierten Labor-Umgebungen, was die Handhabung von zuvor unbekannten und unregelmässigen Objekten erschwert. Wir präsentieren einen Ansatz zur Erfassung von Objekten und Planung von Greif-Konfiguration für einen autonomen Schreitbagger. Das Perzeptions-System erstellt inkrementell eine zeitlich und räumlich konsistente LiDAR-basierte Karte der Umgebung des Roboters. Es besteht die Möglichkeit die Abdeckung der Karte zu erhöhen indem extern rekonstruierte Karten mit der Szene abgeglichen werden könne, z. B. von Kamerabildern, die mit einer Drohne aufgenommen wurden. Wir nutzen Punktwolken und Oberflächenmodelle von Steinen, um Griffe für einen am Bagger montierten 2-Backen-Greifer zu planen. Mit der Geometrie des Steins prüfen wir die Kraftschlüssigkeit des Griffs und wir berücksichtigen Kollisionsbeschränkungen während der Objektaufnahme und -platzierung, basierend auf Punktwolken-Karte. Darüber hinaus zeigen wir einen Ansatz, um beliebig geformte Objekte, die nicht direkt an der gewünschten Stelle platziert werden können, neu zu orientieren, ohne Kollisionen zu riskieren.

Schliesslich werden die vorgestellten Methoden zur Manipulation von Objekten mit einer geometrischen Ziel-Posenplanung kombiniert, um strukturell stabile Objekt- Kompositionen entlang einer gewünschten Form zu berechnen. Wir kreieren einen neuartigen Prozess, der digitalisierte Objektmodelle in einer Physik-Simulation platziert, um Kontakte zu finden und die Stabilität zu bewerten, und abwechselnd das Objekt mit einem Formschluss an die 3D Ziel-Geometrie anpasst. Die tatsächliche Positionen der platzierten Objekte wird lokalisiert und in der Simulation aktualisiert, um einen genauen digitalen Zwilling der Szene zu erhalten. Unser Ansatz wurde in mehreren Anwendungen validiert: von kleinen vertikalen Steintürmen bis zu Trockenmauern aus über hundert Felsblöcken mit einem Durchschnittsgewicht von über 1000 kg, gebaut mit dem autonomen Bagger HEAP.

Acknowledgements

Numerous people have contributed to the creation of this doctoral thesis. First and foremost, I would like to thank my supervisor, Marco Hutter, for giving me the chance to pursue this degree at the Robotic Systems Lab (RSL). He was giving me the support, encouragement, trust, and freedom needed to obtain a PhD. With the RSL, he has created an exciting and intellectually inspiring environment that provides all the resources to conduct excellent research.

I'm honored and grateful to have Prof. Mathias Kohler (ETH Zürich), Prof. Nils Napp (Cornell University), and Prof. Melanie Zeilinger (ETH Zürich) to join the committee and review my work as all of them provide profound expertise in their domain.

As my thesis was carried out as part of the NCCR Digital Fabrication, I would like to thank the dFab management for providing an environment that encourages interdisciplinary collaborations. Indeed, I had many close collaborations during my PhD, and a special thank goes to Fadri Furrer, Hironori Yoshida, Jan Carius, Andrea Carron, Ruben Mascaro, Yifang Liu, and Ryan Johns for their contributions, inspiring discussions, and the exciting and intense time we spent together.

I want to thank all present and past members of the Robotic Systems Lab that I had a chance to work with. Many of them provided valuable input, and all of them were great friends contributing to a supportive and cheerful work environment. A special thank goes to my supervisor Farbod Farshidian for his guidance and many fruitful discussions. Without the support of the current and former colleagues of the excavator team Dominic Jud, Philipp Leemann, Gabriel Hottiger, Simon Kerscher, Edo Jelavic, Burak Cizmeci, Pascal Egli, Tom Lankhorst, Koen Kramer, and Julian Nubert it would not have been possible to conduct such exhaustive experiments. Likewise, I want to express my gratitude to Eris Sako and Johannes Pankert for their help in developing a small-scale manipulation platform.

Similarly, I want to thank the RSL staff, Maria Trodella, Bruno Kaufmann, and Konrad Meyer for making our life easier and allowing us to focus on research. I am also deeply grateful to all the students that helped to pursue ideas or explore

alternative research directions, and many of them contributed in some way to this thesis.

My most profound gratitude goes to my parents, Maria and Anton, my sisters, Karin, Yolanda, Anita, and Eveline and my partner Julia for unconditionally supporting me in everything I do.

Zürich, Spring 2021 *Martin Wermelinger*

Funding Acknowledgements

This work was supported in by the Swiss National Science Foundation through the National Centre of Competence in Digital Fabrication (NCCR dfab).

Contents

List of Abbreviations

AscTec	Ascending Technologies
CAN	Controller Area Network
CNN	Convolutional Neural Network
CoM	Center of Mass
CRC	Collective Robotic Construction
DoF	Degrees of Freedom
EKF	Extended Kalman Filter
ETH	Eidgenössische Technische Hochschule
FPFH	Fast Point Feature Histogram
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GUI	Graphical User Interface
HEAP	Hydraulic Excavator for an Autonomous Purpose
HoG	Histogram of oriented Gradients
IBVS	Image-based Visual Servoing
ICP	Iterative Closest Point
IK	Inverse Kinematics
IMU	Inertial Measurement Unit
ISE	Integral Square Error
ISS	Intrinsic Shape Signatures
KNN	k-Nearest Neighbor
LiDAR	Light detection and ranging

LQ	Linear-Quadratic
LQR	Linear-Quadratic Regulator
MBZIRC	Mohamed Bin Zayed International Robotics Challenge
MPC	Model Predictive Control
NLOC	Nonlinear Optimal Control
ODE	Open Dynamics Engine
PBVS	Position-based Visual Servoing
PCA	Principal Component Analysis
PCL	Point Cloud Library
PD	Proportional-Derivative
RANSAC	Random Sample Consensus
RBD	Rigid Body Dynamics
RoPS	Rotational Projection Statistics
ROS	Robot Operating System
RTK	Real Time Kinematic
SEA	Series Elastic Actuator
SfM	Structure from Motion
SHOT	Signature of Histograms of Orientations
SLAM	Simultaneous Localization and Mapping
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UTM	Universal Transverse Mercator
VI	Visual-Inertial

Nomenclature

A	support polygon area
$\boldsymbol{b}$	centrifugal and Coriolis terms
c	cost term
$\mathcal{C}$	camera frame
d	distance
E_{kin}	kinetic energy
$\boldsymbol{e}$	error
$\boldsymbol{e}_{x,y,z}$	unit vector in x/y/z-direction
$\boldsymbol{e}_o$	orientation error
f	frequency / cost function
f_{cam}	focal length
$\boldsymbol{f}$	system dynamics / attraction force
$\mathcal{F}$	local reference frame
$\boldsymbol{g}$	gravity terms
$\mathcal{G}$	gripper frame
$\mathcal{H}$	horizontal LiDAR frame
$\boldsymbol{J}$	jacobian matrix
k_s	spring stiffness
$\boldsymbol{k}$	keypoint
$\boldsymbol{K}$	stiffness matrix
L	intermediate cost
$\boldsymbol{\lambda}_{\text{ee}}$	wrench (force/torque) at end-effector
$\boldsymbol{M}$	mass matrix
$\mathcal{M}$	target map frame
$\boldsymbol{n}$	normal direction vector
$\boldsymbol{\nu}$	input trajectory / Eigenvector / thrust line
o	object
$\mathcal{O}$	object frame
p	probability / point

$\boldsymbol{P}$	(object) point cloud
$\boldsymbol{p}_{\text{ee}}$	end-effector pose
$\boldsymbol{q}$	generalized joint positions
$\dot{\boldsymbol{q}}$	generalized joint velocities
$\bar{\boldsymbol{q}}$	rotation quaternion
$\boldsymbol{Q}$	cost weighting matrix
r	gear ratio / radius / resolution
$\boldsymbol{r}$	position
$\boldsymbol{r}_c$	centroid
$\boldsymbol{R}$	rotation matrix / input cost weighting matrix
S	support polygon
$\boldsymbol{S}$	scan point cloud
$\mathcal{S}$	source map frame
t	time
$\boldsymbol{T}$	transformation
$\mathcal{T}$	tooltip frame
$\boldsymbol{\tau}$	joint torque
$\boldsymbol{u}$	control input
Φ	terminal cost
$\boldsymbol{\Omega}$	information matrix
$\boldsymbol{v}_m$	actuator motor velocities
$\mathcal{V}$	vertical LiDAR frame
w_i	cost function weight
$\boldsymbol{w}_{\text{ee}}$	end-effector twist
$\boldsymbol{x}$	state vector
$\boldsymbol{z}$	measurement vector

In general, scalar values are lower case and normal style. Vectors are lower case and bold style. Matrices are upper case and bold style.

1

Introduction

Over the last decades, significant advances were achieved in robotic manipulation and assembly. Especially pick and place tasks are already widely researched in the robotic community and heavily applied in industry for various goods such as cars, electronic devices, or toys. However, they are employed in controlled environments to objects with well-defined properties in mass and shape. Often, the task is geometrically defined, and accurate trajectory following is the primary competence needed. In contrast, contacts between the manipulator and the environment are avoided, and interaction forces are considered a disturbance. As opposed to the heavily automated fabrication of mass-produced goods, many sectors rely on manual labor for simple assembly tasks because current robotic systems are not adaptive enough to ever-varying tasks and object properties like pose and shape. For example, on a construction site, a robot has to deal with a changing and unstructured environment, uncertainties of object location and geometry, and varying object scale. There will be a need for flexible robotic solutions in the future, as robots could help solve the labor shortage in the construction sector. Jobs in this sector are physically demanding, repetitive, and often have an increased risk for accidents and injuries. Demographic change and fewer young people willing to work in this sector will further aggravate this shortage and increase the demand for construction robots.

Whereas humans are incredibly versatile in grasping and manipulating arbitrary objects with various properties (shape, dimensions, weight) and surface characteristics, this is still a challenging task in robotics to solve. Humans utilize visual and haptic feedback to get a model of a previously unknown object, adapt to its shape and property, generate an appropriate grasp intuitively, and confirm a proper hold. Remarkably, they can perform such tasks solely using haptic feedback, whereas robotic systems heavily rely on vision to identify object instances and create object models. For reliable and robust manipulation, the human hand is an ideal tool, unmatched in robotics. It is adaptive but still versatile with many controlled degrees of freedom and an opposable thumb. Additionally, the haptic sense gives high-quality feedback of the contact situation and allows to place objects stably.

This adaptivity arises from keeping the impedance (force resistance) low during contact, which allows the hand to adjust to the occurring contact forces.

Motivated by allowing robots to interact with their surroundings, there has been a paradigm shift in the last years from stiff and accurately position-controlled to compliant and force-controlled manipulators that can adapt to unstructured environments. This shift is made possible through recent improvements and developments of series elastic actuators and high bandwidth torque controllable actuators that allow low end-effector impedance. They make it possible to be sensitive enough to assess and control the contact situation between objects precisely. The interaction between a gripped object and the environment is no longer a disturbance but the desired entity to sense, control, and shape. Stiff systems, however, will fail if the uncertainty of the interaction becomes significant, as incremental changes in joint position will decide about contact and eventually lead to unacceptable high contact forces. Additionally, a low impedance system is capable to dynamically interact with the environment and quickly react to unexpected impacts without harming itself or the environment. Even if construction robots do not yet reach the versatility, dexterity, and low impedance of humans, they hold the potential to help us in many areas and might prevent the need to expose workers to strenuous or dangerous situations.

1.1 Motivation and Objectives

We are interested in automating construction processes, as it promises more efficient, more sustainable, and safer building operations and enables the implementation of novel types of architectural structures with unprecedented complexity and functional properties. In our work, we investigate the direct utilization of raw or minimally processed solid irregular material. With *irregular*, we refer to a complex surface shape containing concavities, tips, and a varying number of potentially curved faces or edges. Raw stone or rubble material can be naturally present in mountainous terrain or disaster sites, and it is especially suited for landscape construction in dangerous-to-access and remote places. The minimally processed material can come from a local quarry or as recycled concrete rubble. Omitting further processing steps and directly incorporating the irregular material in the robotic construction process reduces the ecological footprint. Locally sourced material and autonomous construction machines have a great potential for landscape building

in hard-to-access areas with complicated logistics or that are hazardous places for human operators. Leveraging locally sourced material is particularly relevant for the task of building utility structures (e.g., retaining walls, noise protection walls, riverbank reinforcements, coastal or avalanche protections). They usually consist of vast quantities of distinctive elements, and the approximate global shape is more important than the exact composition.

The day we will see a construction site that is mainly operated by autonomous mobile robots is still far away. Nevertheless, using (semi-)autonomous agents on the construction site is up-and-coming to reduce strenuous physical work and accurately and continuously sense the current state and feed it back, informing the construction design and adapting it.

This work aims to enable robots to perform assembly tasks with arbitrarily shaped objects in real-world scenarios and focuses on the following aspects:

Perception & Manipulator Control: We present approaches for perceiving the environment, reconstructing object instances, control interaction forces, and grasping and assembling objects.

Structural Planning: We investigate how to bring objects in contact under uncertainty and plan stable assemblies of irregular objects.

Validation & Execution: Ultimately, we want to go beyond human capabilities and manipulate objects exceeding manual carrying capacity using a walking excavator as a large-scale mobile manipulation platform.

All this allows performing automated robotic landscaping with material found at the construction site as applied in dry-stone masonry. However, possible application areas are not restricted to the construction sector solely as robust and stable object placing under uncertainties emerges in many applications with a dynamic environment or by interaction with a human. Such applications can be found in household and service robotics, collaborative assembly tasks with humans, inspection and operation of remote facilities, or search and rescue tasks.

1.2 State of the Art

This section provides insights into the current advances of integrating robotic systems in construction and their applications, focusing on automated assembling.

Each chapter will cover further related work specific to the individual chapter's topic itself.

Utility structures, i.e., structures with a specific function but whose exact shape matters less, are a vital part of our infrastructure. Examples include erosion barriers for changing coastlines, temporary support structures in disaster sites, or containment vessels made for contaminated materials of a nuclear or chemical leak. A construction method that is particularly well-suited for such types of utility structures is dry stone stacking, as it can make use of locally available material. Skilled masons have practiced this method over thousands of years and learned how to compose stable structures, typically forming those according to some heuristics [1]. These ancient methods are still in use today, especially for landscaping, however primarily in niche applications as they are highly labor-intensive.

Several projects demonstrated the potential of large-scale autonomous construction with on-site robotics and an overview is presented by Melenbrink, Werfel & Menges [2]. Generally combining mobile platforms with conventional industrial robots, these compound setups have been predominantly demonstrated with standardized building components [3, 4] or industrially produced chemical products [5]. In the automation of excavation work, Jud *et al.* [6] showed impressive results in creating accurate free-form trenches with force-controlled trajectories using the Hydraulic Excavator for an Autonomous Purpose (HEAP). Recent research has demonstrated the potential of robotic systems to construct auxiliary structures in order to achieve and maintain navigability in previously unknown or untraversable terrain. Thangavelu *et al.* [7] used decorative creek and pebble stones to build motion support structures autonomously, however on a small scale. Others did not make use of naturally occurring found construction materials (like stones) but instead used compliant bags [8] or polyurethane foam [9] to homogenize the irregularities in their environment. Foreseeing future applications, robotic construction with in-situ material will become more critical for building extraterrestrial structures. Launching building materials into space is very costly, yet simple structures – such as berms, walls, and shelters – might be readily built from minimally processed but rearranged materials [10, 11].

Petersen *et al.* [12] present an overview of multi-robot systems that can construct autonomously building structures far more extensive than the individual robots themselves and introduce them with the term Collective Robotic Construction (CRC). Those systems are often inspired by how animals cooperate for building nests, pro-

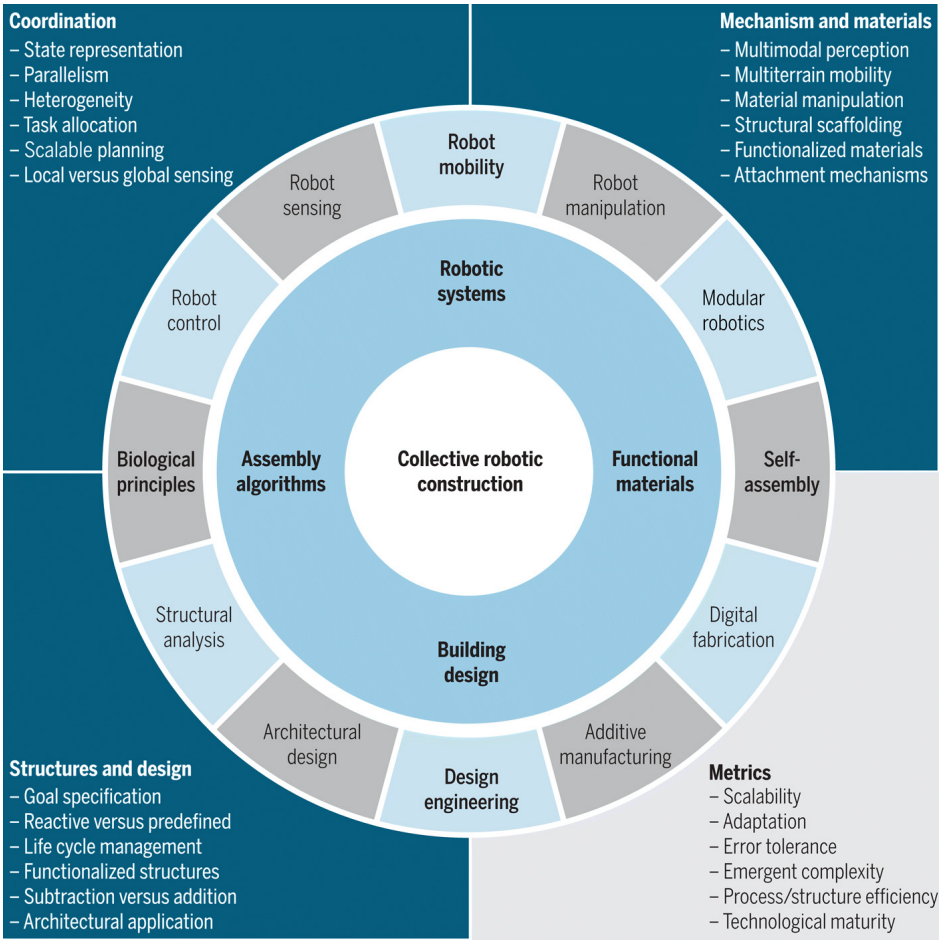


Figure 1.1: The emerging field of Collective Robotic Construction (CRC) is at the intersection of many existing fields like architectural design, construction processes, and robotics [12].

tection barriers, or traps. Although CRC focuses on using multiple agents cooperatively, we share the same goals of tightly integrating architectural design, construction processes, robotic mechanisms, and control to achieve scalability and adaptability (see Figure 1.1). Like CRC, we see our work in the combination of robot control, sensing, and structural design, enabling a digital fabrication process with nonstandard input material.

Re-establishing nonstandard construction material in the domain of architectural digital fabrication is gaining popularity recently [13]. However, it is still a niche, and only a few projects deal with irregular stone-based assembly. De Boer *et al.* [14] used automotive digitization tools to create an inventory of partially-dressed limestone for facade planning, and Clifford, McGee & Muhonen [15] present the idea of reusing construction debris for masonry through robotic stone-cutting methods. Whereas this shows the potential of combining masonry techniques with digital

Figure 1.2: The work presented in this thesis has been mainly evaluated on the mobile manipulation platforms mANYpulator (left) and HEAP (right). Although varying in size considerably, they both are designed for a versatile interaction tasks and include complete onboard sensing and a force-controllable arm.

models, we want to focus on complete usage of the available material without processing.

1.3 Thesis Organization

The content of this thesis is organized into three main parts: Mobile Manipulator Control, Object Handling, and Discrete Assemblies. In the first part about Mobile Manipulator Control (I), we introduce the mobile manipulation platform mANYpulator (see Figure 1.2, left), a versatile robot designed for autonomous mission and interaction with an unstructured environment. We describe navigation and obstacle detection approaches and present a force controllable manipulator composed of series elastic actuators. With this manipulator, we can use low end-effector impedance and obtain precise measurements about the applied joint torques. Together with a precise model of the system, contact points and contact forces can be estimated. As the end-effector load in assembling is unknown and may vary considerably, we present an actuator-aware receding horizon Model Predictive Control (MPC) method that achieves robust high accuracy end-effector tracking while preserving the natural compliance of the manipulator.

In the second part, Object Handling (II), we present a complete framework to map, segment and reconstruct large-scale object instances with laser sensor data

in a real-world environment. Furthermore, we show an approach for localizing the objects in a scene and present a grasp planner that generates configurations for reorienting and placing objects in a cluttered environment. We demonstrate the applicability of our approach with real-world construction on an architectural scale and deploy the previously developed methods on the autonomous excavator HEAP (see Figure 1.2, right).

The third part, Discrete Assemblies (III), discusses how the developed tools in control and object manipulation are employed to create dry-stone structures in real-world applications, from small-scale vertical towers to large-scale retaining walls. We focus on the construction with irregularly shaped stones and present methods to plan and assess structurally stable assemblies in simulation with the digitized object models. Incorporating feedback from the actual scene in the planner, we prove that the planned assemblies are translatable into physical structures composed of over one hundred objects.

Last, we are concluding this work in Chapter 8, summarizing the core contributions of each chapter and providing an outlook about future developments.

Part I

MOBILE MANIPULATOR CONTROL

2

Mobile Manipulator for Interaction Tasks

This chapter incorporates material from the following publications:

Carius, J., Wermelinger, M., Rajasekaran, B., Holtmann, K. & Hutter, M., Deployment of an autonomous mobile manipulator at MBZIRC. *Journal of Field Robotics*, **35** 8, pp.1342-1357 (2018).

Carius, J., Wermelinger, M., Rajasekaran, B., Holtmann, K. and Hutter, M., *Autonomous Mission with a Mobile Manipulator – A Solution to the MBZIRC* In *Field and Service Robotics*, pp. 559-573, Springer, 2018.

Video: https://youtu.be/iUPJ73Y5yMw

Autonomous manipulation in unstructured environments needs a generic mobile platform designed for various tasks, not specifically tailored to a single application. An excavator, for example, is one of the most versatile machines used for varying tasks. However, for conducting ongoing research in navigation and compliant manipulation, we decided to design a small-scale modular platform suitable for deployment in real-world applications: A robot composed of a commercially available base, a manipulator, and a gripper that are connected through an internal network. We went for a custom manipulator because classical industrial arms are rigidly position-controlled and not well suited to handle the position uncertainty introduced by the mobile platform. Instead, our force-controlled arm, an advancement of [16], can cope with the uncertainties in the real world. The robust mechanical structure, advanced sensing capabilities, and an integrated software stack allow our system to operate in unstructured and unknown environments and to complete navigation and manipulation missions therein. The software structure is optimized to be modular, allows autonomous operation through a state machine, and permits manual intervention if necessary.

In this chapter, we show two core contributions. First, we present a versatile six Degrees of Freedom (DoF) robot arm suited for interaction tasks, called *ANYpulator*, based on torque-controllable Series Elastic Actuators (SEAs). We show how the arm is incorporated into a complete mobile manipulation platform *mANYpulator* suitable for autonomous outdoor missions. In contrast to off-the-shelf solutions, our design does not rely on extreme position accuracy. Instead, it uses force control to be guided by the geometry of the manipulation object. Second, we evaluate our system's capabilities by competing in a robotics contest, the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2017. We describe a set of implemented behaviors concerning perception, navigation, and manipulation skills required and demonstrate how those capabilities are integrated into a single system with a flexible interface for operator interactions. We decompose the complex mission into simpler building blocks that a central task dispatcher coordinates.

The presented concept and system were used to master the second challenge of the MBZIRC. The task was to maneuver an Unmanned Ground Vehicle (UGV) to locate a wrench panel on a 60 x 60 m outdoor field, navigate to it, select and grip a suitable wrench hanging on the panel, and turn a valve stem 360 ° with the tool. This challenge's particular focus lies on entirely autonomous behavior since any intervention by a human operator will directly result in significant score penalties. Furthermore, we employed the system in the grand challenge, where the UGV task is executed simultaneously with an Unmanned Aerial Vehicle (UAV) mission, documented in [17]. The second challenge of MBZIRC was exclusively solved by wheeled mobile robots in combination with a manipulator arm. Some teams of the MBZIRC have officially documented their results for challenge 2, for example [18–22]. Additionally, team websites[1] provide supplementary information on the robotic systems used during the competition, and most teams have published progress videos[2]. A comprehensive overview of the system and its performance during challenge two is shown by the Desert Lion Team of the University of Padova (Italy) [23]. Their system design is, similar to ours, motivated by the idea to have a modular and general platform.

The remainder of this chapter will cover the design of a mobile robot for navigation and manipulation tasks (Section 2.1); a software architecture for autonomous mission execution (Section 2.2); a complete localization and navigation software

1 All qualified teams are listed under https://www.mbzirc.com/qualified-teams/2017 (Accessed: 2021-04-15)

2 See https://mbzirc.com/video/first-progress-report (Accessed: 2021-04-15)

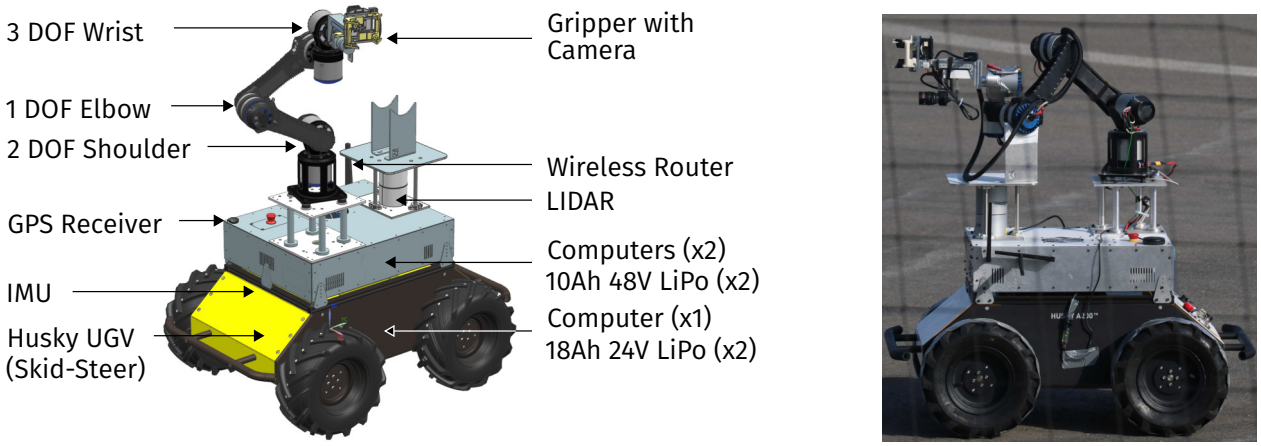


Figure 2.1: Overview of the integrated mobile manipulator platform mANYpulator.

stack (Section 2.3); and manipulation capabilities through visual servoing (Section 2.4) and manipulator control (Section 2.5). A discussion of the system's overall performance in the MBZIRC is given in Section 2.6, and we summarize the chapter with Section 2.7.

2.1 Hardware and System Design

This section presents our mobile manipulation platform's hardware setup and communication architecture, which is shown in Figure 2.1. We paid particular attention to designing a modular system, allowing for exchanging the mobile base, mounting different sensors, attaching the manipulator at different positions, exchanging the end-effector tool, or possibly adding more than one robot arm.

2.1.1 Mobile Base

As a ground vehicle, we use the four-wheeled skid-steer platform Husky designed and manufactured by Clearpath Robotics. We opt for a wheeled platform due to its speed and efficiency in locomotion, simplified state estimation and control, and high payload capabilities.

The UGV is equipped with several sensors that allow it to navigate in unknown environments autonomously. The mobile robot perceives its state and environment through wheel encoders, a Microstrain 3DM-GX4-15 Inertial Measurement Unit (IMU), a Velodyne HDL-32E Light detection and ranging (LiDAR) sensor, and op-

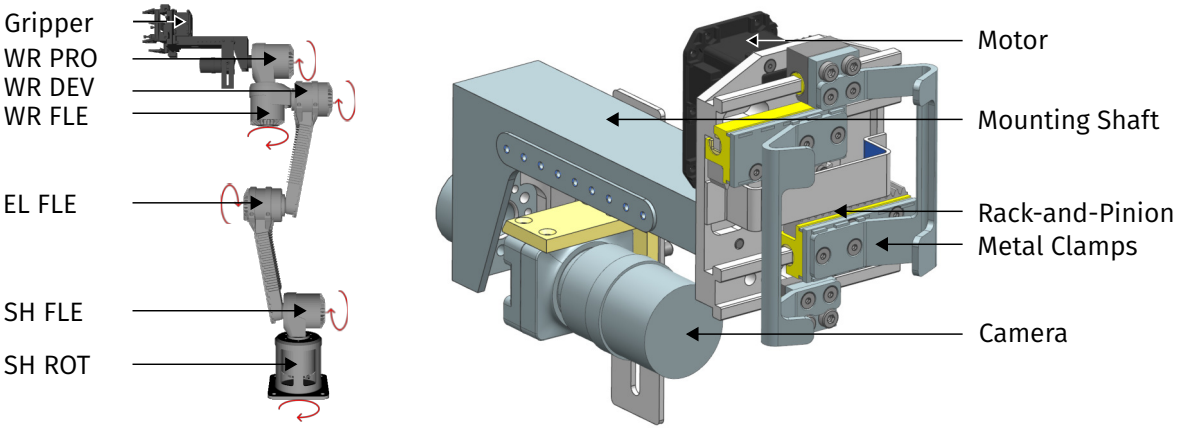


Figure 2.2: **Left:** Structure of the manipulator with labeled joint topology of shoulder (SH), elbow (EL), and wrist (WR). **Right:** Gripper design for wrench manipulation with clamps to envelop wrench shafts and an attached camera for eye-in-hand visual servoing.

tionally a Garmin GPS-18x Global Positioning System (GPS) receiver. An overview of the mounted sensors and their uses can be found in Table 2.1.

2.1.2 Manipulator

We built a six DoF manipulator arm, which is mounted on the mobile platform. This arm is similar to previous work [16], but now three DoF are allocated at the wrist (see Figure 2.2, left). This configuration increases the manipulation capabilities and allows to perform various tasks with the tool but comes with the downside of higher weight and inertia near the end-effector. The same high-performance SEA units actuate all joints of the arm. These actuators, called ANYdrives [24], are composed of a brushless high-torque motor, harmonic drive gear, a torsional spring, and integrated control electronics and sensors (including absolute encoders). The spring adds inherent compliance to the system, making it suitable for interaction tasks such as valve manipulation. Several drives can be chained together along a Controller Area Network (CAN) and DC power bus. To keep the inertia low, the arm's link and base components consist of hollow aluminum parts. In combination with the hollow driveshafts, cables can be routed through the interior of the robot arm. Attached to the final motor is the end-effector (see Figure 2.2, right). The gripper's mechanical design is optimized to provide a lightweight but effective tool for holding the shafts of standard wrenches. A single Dynamixel MX-64 motor drives two sheet metal clamps through a rack and pinion mechanism. The clamp mech-

Table 2.1: Sensors mounted on mANYpulator and their respective use.

Sensor	Location	Purpose
Wheel encoders	UGV wheels	UGV state estimation, localization
IMU	Attached to UGV frame	UGV state estimation, localization
LiDAR	On top of UGV	Localization, general perception tasks
GPS	On top of UGV	Navigation, localization
SEA	Arm joints	Joint position, speed, and torque measurement
Camera	Attached to end-effector	Visual servoing, general perception tasks

anism ensures that the closing position is centered laterally with the end-effector reference frame. Additionally, the clamps are designed to ensure that the head of a gripped wrench coincides with the rotation axis of the last joint. This design facilitates the pure rotational motion of the tool, e.g., for turning the valve stem. Sensing current and position at the Dynamixel allows determining whether or not the gripper successfully holds an object. For visual servoing, a calibrated monocular camera (PointGrey Chameleon 3) attaches to the aluminum shaft below the clamps, such that a grasped wrench does not obstruct the view of the camera.

2.1.3 Computation, Communication, and Power Supply

The storage space inside the UGV, together with a custom assembly on top (metallic box in Figure 2.1), provides space for additional computational units, power supply, and communication devices. Our mobile robotic system contains three onboard computers connected to an operator computer through a wireless network. Figure 2.3 shows the setup of our computational infrastructure:

Arm control: One computer executes control algorithms for the robot arm and manages the communication with six SEAs. Bandwidth limitations of the CAN bus allow a control loop frequency of 200 Hz, which requires approx. 40 % of the processing power of an Intel i7 3.1 GHz processor.

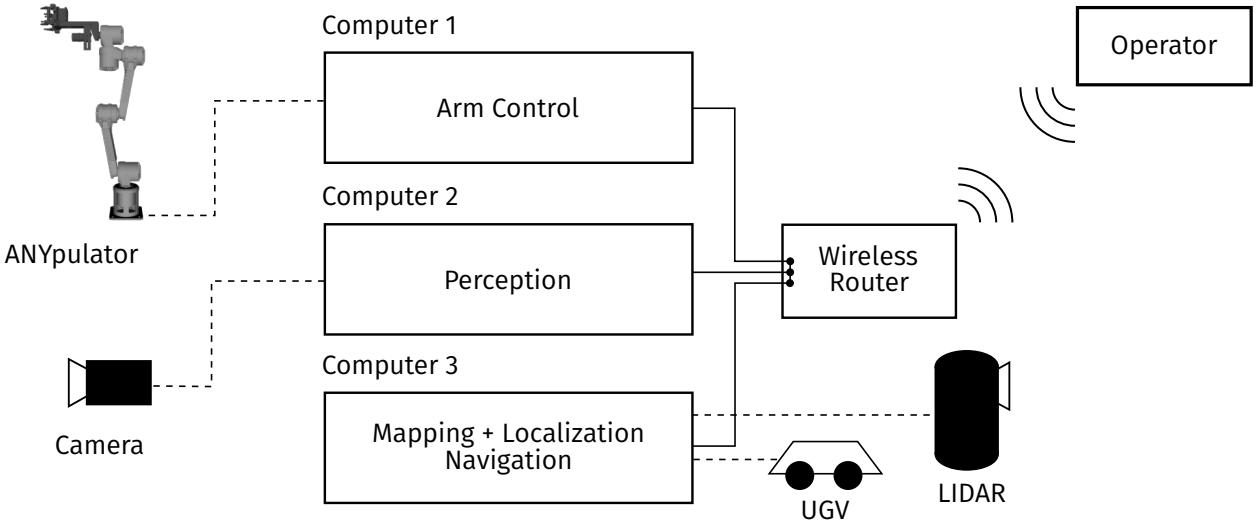


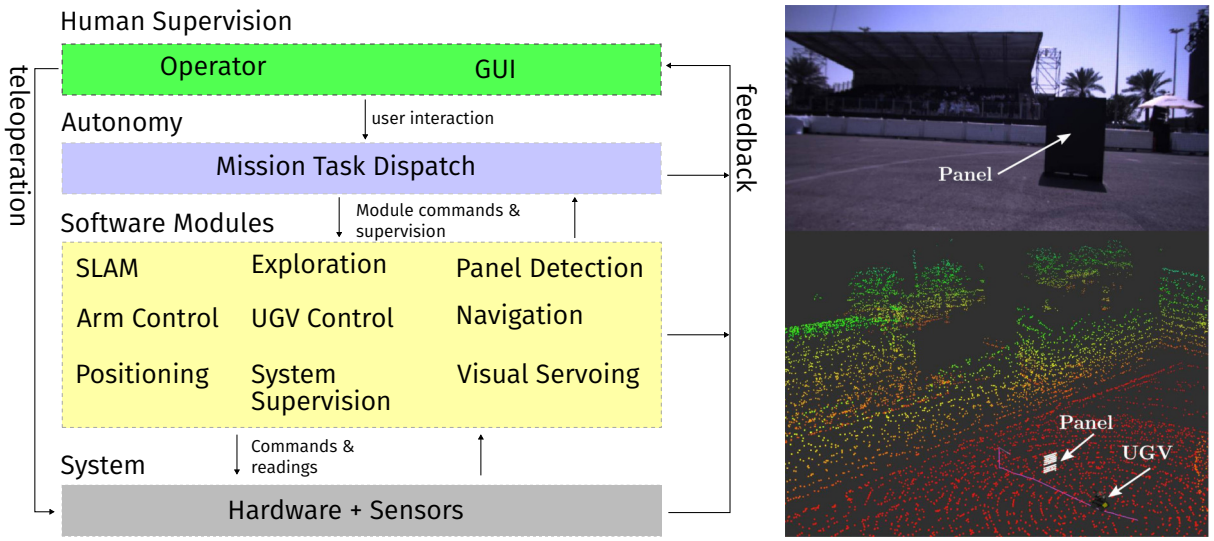
Figure 2.3: Tasks related to autonomous operation of the robot are distributed among three computers on the mobile system. All internal communication on the robot is transmitted through wired ethernet. The operator may supervise the system over a wireless connection.

Perception: A second processing unit with the same specifications is dedicated to perception tasks involving processing the camera feed or point cloud data but can also be allocated to a specific user application.

Navigation: Finally, a pre-installed[3] computer inside the UGV's body is responsible for the navigation stack. It communicates with the UGV motor drivers and processes measurements from GPS, IMU, and LiDAR for the complete localization, mapping, and navigation solution. The computationally intense localization algorithm utilizes the full computational capacity of this computer's Intel i5 2.7 GHz processor.

The Robot Operating System (ROS) [25] framework handles the communication between processes on the same machine and between different computers. ROS enables the distribution of computational power by running resource-demanding applications on separate machines. Finally, onboard batteries provide sufficient energy for powering the system during missions of several hours.

3 Due to the special form factor and mounting by the manufacturer of the UGV, this computer was not replaced.



(a) (b)

Figure 2.4: **Left:** Hierarchical software architecture and abstraction layers. The operator can interact with the mission task dispatcher or teleoperate the system directly. The autonomy software activates different functional modules and monitors them. **Right:** The operator's view of the gripper camera feed and the robot's position in the constructed map. The detected panel is displayed in white in the map.

2.2 Software Architecture and Mission

We expect our robot to fulfill a typical mission involving several sequential steps such as exploration, detection of the manipulation site, navigation, positioning, and manipulation. Tasks of this complexity calls for a modular mission design since each sub-task may individually fail and needs to be repeated. We devise a hierarchical software architecture depicted in Figure 2.4a. At its core, individual modules such as localization, path planning, or arm control are tailored for specific functions. The abstraction into individual modules with standardized interfaces facilitates the redesign of single elements when adapting to new environments, fulfilling new requirements, or testing new algorithms.

The mission task dispatcher introduces autonomy into the system by activating the required set of software modules at any given time and monitoring their outcome or performance. In Figure 2.5, we display our mission state flow diagram for the MBZIRC. Each state is responsible for a specific sub-routine and is accompanied by a supervision module (not shown in the figure) that decides if the task was successful or not. In case of failure, specific recovery behaviors may be executed, or the state machine merely transitions back to the previous task and starts another

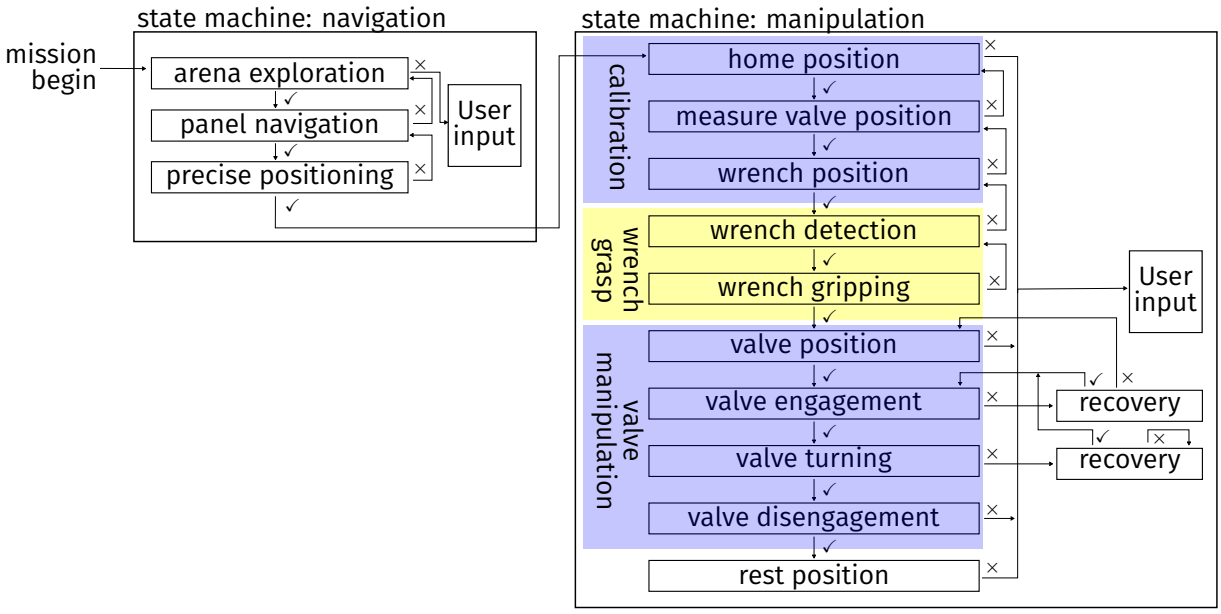


Figure 2.5: State flow during the MBZIRC mission. Each state may succeed (✓) or fail (×). On failure, the robot will execute the previous task again, initiate a recovery action, or request user input.

attempt. This architecture also allows for interruption of the current task on predefined conditions, e.g., when the battery runs low, or the operator requests the mission to stop.

Said operator can monitor the robot through an additional operator computer connected to the robot's wireless network (Figure 2.3). A Graphical User Interface (GUI) displays information from all relevant software modules and diagnostic data such as battery charge state and temperatures. Additionally, a visualization of the constructed environment map and the image stream from the camera are displayed to the operator (Figure 2.4b). At any time, the operator may interrupt the autonomous mission and provide manual commands through a joystick or the GUI. The GUI empowers a user to issue specific commands such as closing or opening the gripper, switching to a specific mission state, or steering the manipulator or mobile platform.

In general, the system is designed to operate autonomously and safely without any operator. Accordingly, all mission-critical software is running on the robot itself and is independent of the quality of the wireless network. Therefore, even under limited network connectivity, the robot maintains its autonomous behavior. However, safety requirements during the MBZIRC required us to make the robot pause operation if there is a connection time-out to the operator's joystick. In the

following chapters, the essential software components will be presented in more detail and evaluated in the context of the challenge.

2.3 Localization and Navigation

This chapter addresses the necessary building blocks that enable our mobile robot to identify the manipulation site and navigate there on a collision-free path. We first introduce our Simultaneous Localization and Mapping (SLAM) solution and describe how our robot moves through an unknown environment. Subsequently, we explain how the wrench panel is identified and how the UGV positions itself relative to the panel.

2.3.1 Simultaneous Localization and Mapping

SLAM is a procedure to build up and maintain a map of the robot's environment and localize with respect to this map. Our implementation builds upon existing algorithms by Pomerleau and Krüsi et al. [26–28] for Iterative Closest Point (ICP) based localization and mapping. The exact details of the algorithms exceed the scope of this work, so only a short overview is given and illustrated in Figure 2.6a.

An Extended Kalman Filter (EKF) fuses IMU and wheel odometry measurements in order to provide a transformation between the robot's odometry and base frames. The incremental nature of encoder and IMU readings ensures a continuous transformation over time but is also susceptible to drift due to wheel slippage and modeling inaccuracies. Therefore, an ICP matching algorithm provides an additional transformation between the odometry frame and an absolute world-fixed frame by aligning the current LiDAR measurement with points in the constructed map. Optionally, GPS measurements can be fed into a batch optimization that estimates the transformation between the Universal Transverse Mercator (UTM) coordinate system and our world frame. This transformation may be used to convert user-specified GPS waypoints into map frame coordinates. The constructed map has its zero position at the starting point of the robot, although for most applications the absolute position of the map is not relevant.

In this work, we contribute some enhancements to the existing solution that make it more stable for deployment in large outdoor fields. First, we observe that the

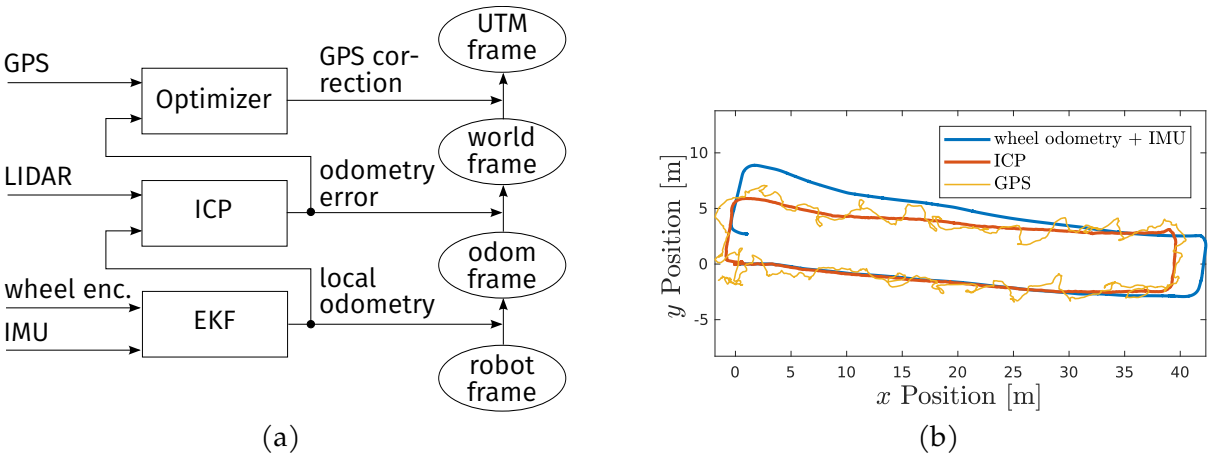


Figure 2.6: **Left:** Localization framework for the UGV. The local odometry is guaranteed to be continuous but may drift over time. Global localization performed by an ICP matching algorithm supplies a (discontinuous) transformation between the world and the odometry frame. **Right:** Time evolution of the robot's position, estimated by IMU and wheel odometry EKF only (blue), with the help of ICP localization (red), and GPS measurements for comparison (orange).

accuracy of the ICP localization suffers from long computation times during the point-matching phase. By the time a pose update is available, it is already outdated if the robot continues to move. Therefore, we down-sample both the incoming scans and the map to reduce computation time. The input scan keeps every point with a probability of 0.1, and a maximum point density of 30 points per m^3 is enforced. The map is restricted to 5 points per m^3, resulting in an update rate of 0.77 Hz on average with a standard deviation of 0.29 Hz. Furthermore, the down-sampling allows us to incorporate laser measurements as far as 100 m (close to the physical capabilities of the sensor) for localization on vast outdoor fields (we verify this on test sites with up to 150 m diameter) with little significant structure or landmarks. Since the accuracy decreases at large measurement ranges, we introduce a dynamic thresholding mechanism to reject mismatched LiDAR scans: After each ICP matching cycle, the magnitude of the proposed translation correction is compared to a dynamic threshold value. If the correction is larger than the current threshold, this update will be rejected, and the threshold increases by a constant factor. Conversely, if the correction is below the threshold, we accept the update and decrease the threshold for the next iteration.

2.3.2 Path Planning and Following

The path planning and following schemes applied to our system are based on the 2D ROS navigation stack[4]. It operates on a 2D grid map (in the world frame), where obstacles correspond to exceedingly high-cost values and free space to zero cost. This grid map is produced and updated by a custom obstacle detection algorithm that operates on 3D LiDAR data [28, 29]: The rotational motion of the LiDAR sensor delivers sequential scan segments in azimuth angle direction. For each segment, all points are sorted in ascending order according to their elevation angle compared to the segment's plane. Each consecutive point pair is then analyzed regarding the inclination between them and their interjacent step height. If these two values exceed given thresholds, we classify the points as obstacles. A cost map is populated and used for path planning by iterating this algorithm for every laser scan segment.

A global trajectory from the current robot pose to the desired goal pose is created by applying Dijkstra's algorithm [30] regarding the cost map as mentioned above. This path will then be followed by a local planner, the so-called Dynamic Windows Approach (DWA) [31]. This approach forward simulates a discrete set of velocities $\Delta\dot{x}$ (translational) and $\Delta\dot{\varphi}$ (rotational) of the robot's control space from the robot's current state over a short period. After evaluating each resulting trajectory by proximity to obstacles, proximity to the goal and path, and speed, invalid trajectories (i.e., those that lead to a collision with obstacles) are discarded, and we pick the best-scoring one. The corresponding velocity commands are then sent to the mobile base. We repeat the local planner at a control frequency of 5 Hz and terminate once the robot has reached its goal position and orientation. The goal-reaching threshold was deliberately set at a substantial value of 0.3 m to avoid time-consuming maneuvering with the non-holonomic robot. Instead, we developed a different strategy for exact positioning (Section 2.3.5).

2.3.3 Field Exploration

The first step in the mission involves finding the panel on the field. Therefore, the exploration module creates a complete coverage path in a predefined rectilinear search space defined by the user via the input of GPS or map frame corner coordinates. This coverage path starts at the center of the search space and works its

4 See `http://wiki.ros.org/navigation` (Accessed: 2021-04-15)

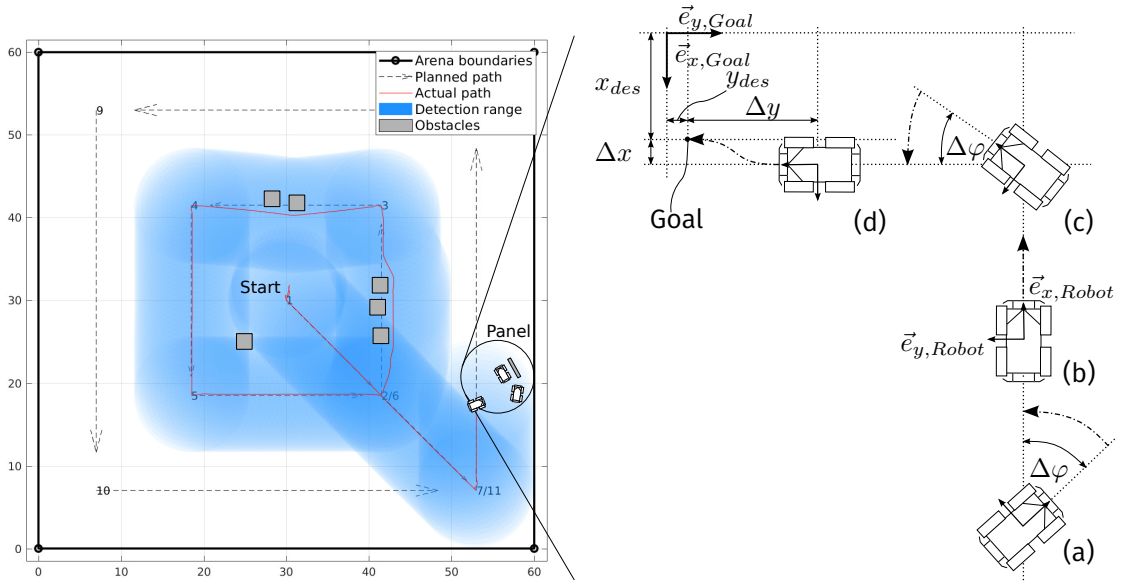


Figure 2.7: **Left:** Waypoints for complete coverage of a rectilinear search space by the explorer module. The actual path deviates due to obstacles in the way and terminates once the panel is located. The shaded region is the detection range of the panel detection algorithm. **Right:** Zoomed in version of the positioning procedure after locating the panel: (a) Facing, (b) Approaching, (c) Turning, and (d) Centering.

way towards the boundaries in counter-clockwise cycles around the center. The pitch between those cycles thereby correlates directly with the detection range of the panel detection algorithm (Section 2.3.4).

An illustration of the coverage path plan is given in Figure 2.7 (left). Furthermore, this module comes with a set of recovery behaviors to increase overall robustness. They ensure that exploration continues when encountering scenarios such as unreachable waypoints due to obstacles. For increased efficiency of the exploration, heuristic priority waypoints can be specified. They are tracked prior to those of the coverage path. Under the assumption that the robot's path does not deviate from the planned path by more than the overlap region of consecutive cycles, the entire field is covered during the exploration. The strategy, therefore, tolerates obstacles that yield to a path change of max. 1.5 m, which is half the overlap distance.

This phase is usually the first stage in a mission and terminates once the concurrently running panel detection algorithm identifies the manipulation site. In a subsequent transition phase, the robot navigates to a fixed position relative to the wrench panel. Once this intermediate goal is reached, the transition completes, and the positioning phase (Section 2.3.5) is triggered.

2.3.4 Panel Detection

An essential element for succeeding in the navigation part of the MBZIRC is the correct identification and pose estimation of the wrench panel. The panel detection software uses the Point Cloud Library (PCL) [32] and operates on the raw scans obtained by assembling laser data from one revolution of the scanner. To identify the panel, we use the known geometric dimensions of the planar rectangular sides as given by the challenge description. Our algorithm can identify any planar side of the panel by executing the following operations:

1. The point cloud is cropped to a box with a side length of 10 m around the robot. The cropping reduces the computational requirements and prevents the detection of false positives.

2. We assign a normal vector to each point in the cloud through principal component analysis of the covariance matrix from neighbors of the query point.

3. We use conditional Euclidean clustering that assigns each point to a cluster to segment the point cloud. Two points share the same cluster if they are sufficiently close (Euclidean distance) and their normals are aligned.

4. Random Sample Consensus (RANSAC) [33] plane extraction selects individual planes in each cluster. We reject non-vertical planes and those that lie outside our search space (specified as a polygon corresponding to the arena geometry, see Section 2.3.3).

5. We check all the remaining planes if their dimensions match any of the rectangular sides of the panel. Finally, we confirm that the point distribution across the plane is approximately uniform in order to reject any remaining artifacts.

If all of the above tests are successful, the identified plane's position and orientation are passed to an estimation module that keeps track of several detections. The individual detections (*measurements*) of the panel pose are fused to potential matches (*candidates*). Each candidate i holds an internal state, containing the pose of the panel's front plane with respect to a map-fixed reference frame and an improper probability $p_i \in [0, 1]$, indicating how likely it is the correct match. New measurements update existing candidates if there exists a sufficiently close one; otherwise, a new candidate will be initialized with probability p_{init} as follows:

Initializing candidates: Upon constructing a new candidate, the pose information of the measurement is transformed to an equivalent pose of the panel's front side by applying a suitable translation and rotation. If the initial measurement is ambiguous, i.e., the algorithm detected the side of the panel, one cannot decisively infer if the detection corresponds to the right or left part. In this case, an arbitrary choice is made, and we set the state of this candidate to ambiguous.

Updating candidates: The first step in updating a candidate is again transforming the measurement to an equivalent pose of the front side of the panel. In ambiguous cases, the rotation is chosen such that it supports the current hypothesis. If a previously ambiguous candidate receives an update by a conclusive measurement (e.g., the front or back side), the ambiguity can be resolved. The updating potentially involves flipping the front and back side if the wrong initial choice was made. A first-order filter updates the position and the rotation through spherical linear interpolation, and the probability is increased by p_{update}.

After each iteration, the candidate manager sorts candidates in descending probability order, normalizes the probabilities such that the highest does not exceed 1.0, and removes candidates with probabilities below a given threshold. We accept the most likely candidate (at index 0) if its probability $p_0 > p_{\text{threshold}}$, and it is significantly more likely than the next best candidate $p_0 - p_1 > p_{\text{diff}}$, if any. The parameters fulfill

$$0 < p_{\text{init}} + p_{\text{update}} < p_{\text{threshold}} < p_{\text{init}} + 2p_{\text{update}} < 1 \,, \tag{2.1}$$

hence at least three detections are necessary for acceptance of a candidate.

2.3.5 Positioning to the Wrench Panel

The purpose of this phase is to position the robot in a way that the panel is within reach of the arm, and subsequent manipulation tasks can be performed. To this end, a fixed sequence of maneuvers is executed, which can be divided into four stages: facing, approaching, turning, and centering. We illustrate the schematic overview of these stages in Figure 2.7 (right). During the first three stages, simple unidimensional velocity commands are applied, that is, a constant rotation around

the robot's z-axis in stage (a) and (c) and linear velocity in stage (b). In the last stage (d), both linear and angular velocities are controlled simultaneously. Thereby, potential residuals in the lateral and longitudinal direction and rotational offsets from the previous stages are eliminated.

2.3.6 Evaluation

Our UGV can reliably localize in diverse environments, ranging from indoor lab spaces to large open fields. Noticeably, our system and navigation routine was originally designed to operate on unpaved ground or sandy soil cluttered with objects that have to be circumvented through obstacle detection. However, since the challenge is performed on asphalt ground, all results in this document are generated on paved ground. We show in Figure 2.6b a comparison between pure wheel odometry and IMU dead reckoning, the ICP localization, and GPS measurements as (noisy) ground truth. It is evident that wheel odometry suffers from wheel slippage and possibly miscalibrated wheel radii. A significant error results both in the heading distance and direction, particularly after in-place rotations where wheel slippage is substantial and the center of rotation is hard to estimate for the skid-steer setup. The LiDAR-based solution, on the other hand, agrees very well with the GPS measurements. Although no additional loop closure optimization is made, driving the closed box-shaped path in Figure 2.6b of approximately 90 m amounts to an error in the order of 10 cm only. In the MBZIRC setup, only a single predefined starting location for the UGV was permitted. Therefore, using the constructed map from a previous test run allowed us to directly specify the field's corner points in the map frame. During the challenge runs, the ICP localization proved to be stable enough, such that we did not use any GPS information.

The exploration module worked reliably in the challenge but is not the most efficient way to search the field in light of prior information on the approximate panel location. At each run, the panel was consistently placed at the far edge of the field. Our original exploration strategy would have traced out a path length of 180 m, while manual waypoints reduced this distance to 57 m. Path following and obstacle avoidance showed reliable performance throughout the challenge and demonstrated its ability to navigate narrow passages in preliminary tests.

The panel detection module correctly identifies the pose of the wrench panel from a distance of 10 m, which is approximately the point at which the rectangular

sides become visible in the point cloud visualization for a human observer. The computational load of processing point cloud data allows update rates of 0.6 Hz, limiting the maximum speed at which the robot can drive past the panel without missing it (given the explorer's path) to 4.3 m/s. In practice, the maximum speed of our ground vehicle (1.0 m/s) does not cause any problems in this respect. A major difficulty for the detection algorithm arises from occlusions in the laser scan. Our scanner is protected by a cover held by three vertical pillars, casting conic shadows with opening angles of 7°. If the panel happens to fall inside this region, the occlusions render the detection difficult or, at times, impossible. As long as the robot remains in motion, those temporary occlusions are well handled by the candidate manager. We depict the performance of our algorithm in Figure 2.8. This plot shows the detected panel locations (front and back side) on laser data from the grand challenge. For this dataset, approximately a quarter of detected points are automatically classified as outliers by the panel detection algorithm and are not shown in the plot. Even though the points are plotted in the ICP frame, which is not entirely stationary, the 95 % confidence ellipse shows that most points lie within ±10 cm of the mean. Furthermore, measurements of the orientation (yaw) of the panel plane after outlier rejection have a standard deviation of only 0.65°. The algorithm is, therefore, capable of accurately identifying the position and orientation of the panel.

Given the panel's location, the positioning maneuver should steer the robot such that there is zero lateral offset and an orthogonal distance to the panel of 0.8 m (see Figure 2.7). We show the final position of the robot relative to the panel for several test runs in Figure 2.9. The algorithm achieves a sub-centimeter accuracy with a standard deviation of 0.021 m in the worst direction. Due to the non-holonomy, it is also clear that correcting an error in the lateral direction y is more straightforward and therefore associated with smaller variance.

The combination of panel detection and positioning algorithms achieves a high enough accuracy for positioning the robot such that the arm can reach all wrenches and the valve. However, additional information from a camera is necessary for the mission's manipulation part because sub-centimeter accuracy is required for engaging the wrench.

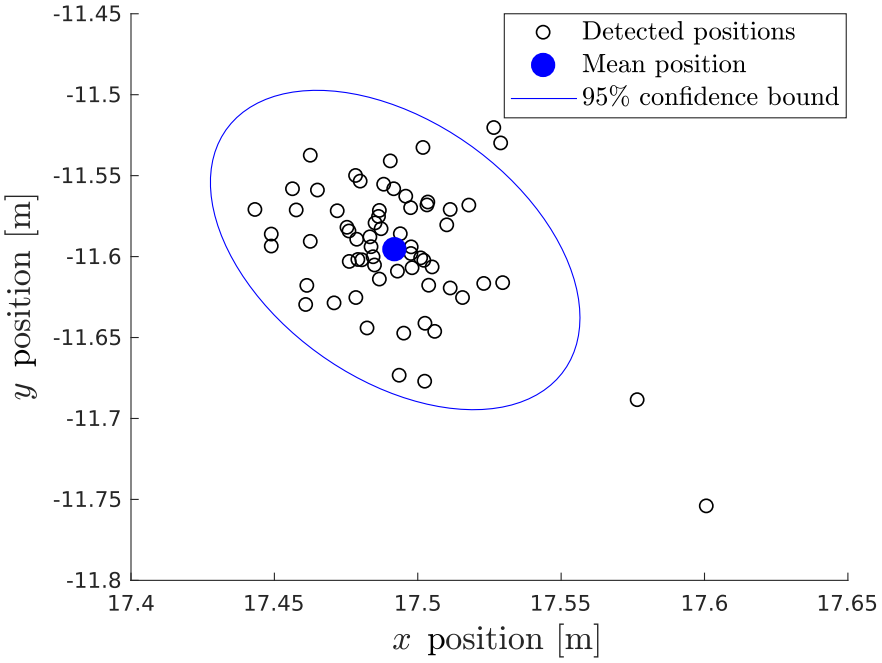


Figure 2.8: Detected panel positions (front and back side) on the field by the panel detection algorithm during the grand challenge (in ICP frame). Outliers that are filtered by the algorithm are not shown. The ellipse marks a 95 % confidence interval centered around the mean (thick blue dot) of the datapoints.

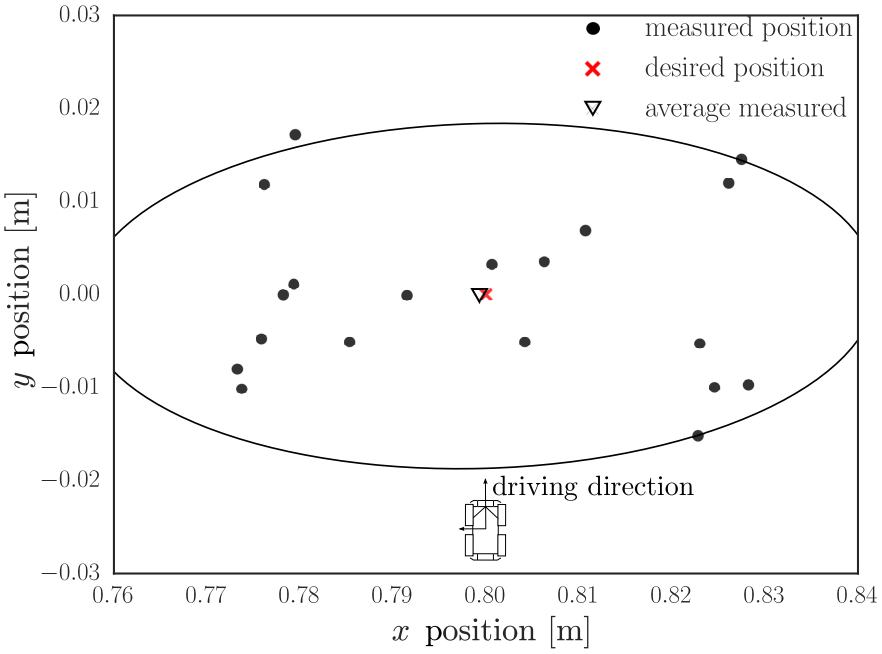


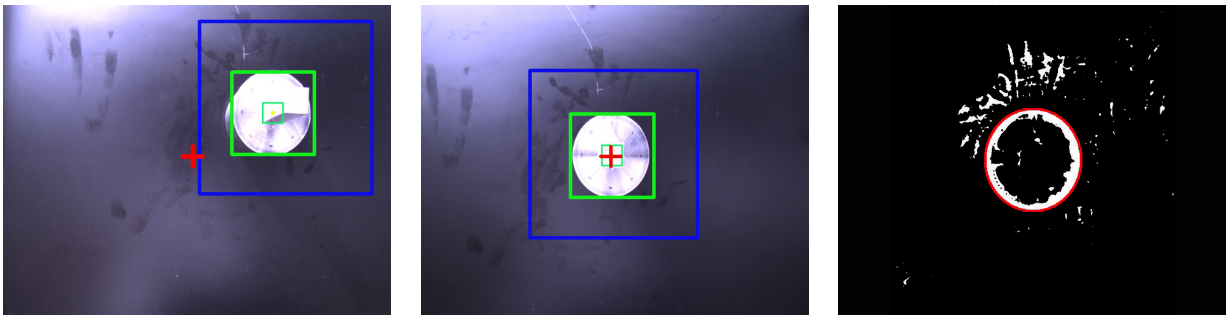
Figure 2.9: Final position of the robot's base with respect to the panel after the positioning phase. The desired x-direction (normal distance) between center of the robot and panel is 80 cm. The ellipse marks two standard deviations around the average position.

2.4 Visual Servoing

Manipulation of the valve using the appropriate wrench is a fundamental task of the mission. The critical elements for succeeding in this task are the correct detection of the valve, precise estimation of depth, selection of the appropriate tool, reliable visual servoing, and proper engagement. The choice of visual servoing for this task is motivated by the fact that the global localization and navigation presented in Section 2.3 is not accurate enough for the exact wrench manipulation. The local nature of the visual servoing does not need further calibration of the robot pose in front of the panel.
A Support Vector Machine (SVM) [34] based object detection method using Histogram of oriented Gradients (HoG) features [35] is adopted in order to achieve high robustness. HoG features are used to capture the object's shape characteristic, which is the most distinct property of the considered objects. A linear SVM model is trained using SVMLight [36] for each object, i.e., valve, wrench, wrench box-end (referred to as wrench ring), and wrench open-end (referred to as wrench head). The training set contains images of the objects augmented for different illuminations to facilitate detection in different lighting conditions. However, we did not vary the background during training since that was specified as uniformly black.

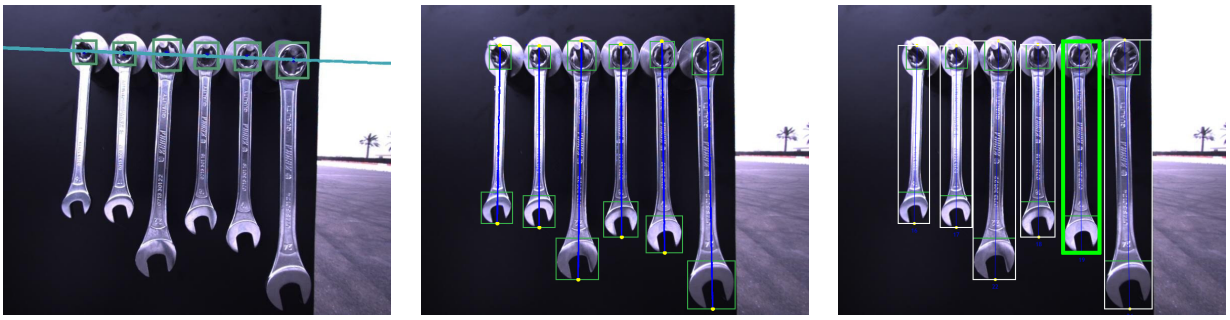
2.4.1 Valve Pose Estimation

We use the known panel layout and use the valve position as the reference point for the subsequent manipulation. To acquire the valve's location in the first place, the end-effector executes a predefined spiral motion pattern while searching for the valve. We use the end-effector camera and the HoG based object detection method mentioned above to detect the valve. Once the valve is found with a certain confidence level, Image-based Visual Servoing (IBVS) [37] is employed to align the center of the camera and the center of the valve, as shown in Figure 2.10a. During this motion, a Kalman Filter keeps track of the valve location. After aligning the camera with the valve (Figure 2.10b), the pose of the valve with respect to the robot's base can be computed using the current pose of the camera and the estimated distance between the valve and camera.
To estimate the distance, we segment the valve from the panel using adaptive thresholding [38]. Its external contour is extracted using an algorithm based on connected



(a) (b) (c)

Figure 2.10: Valve position calibration procedure: (a) Valve tracking inside a region of interest (blue). (b) Successful alignment of the camera center (red) with the valve (green). (c) Detected valve contour for depth estimation (red).



(a) (b) (c)

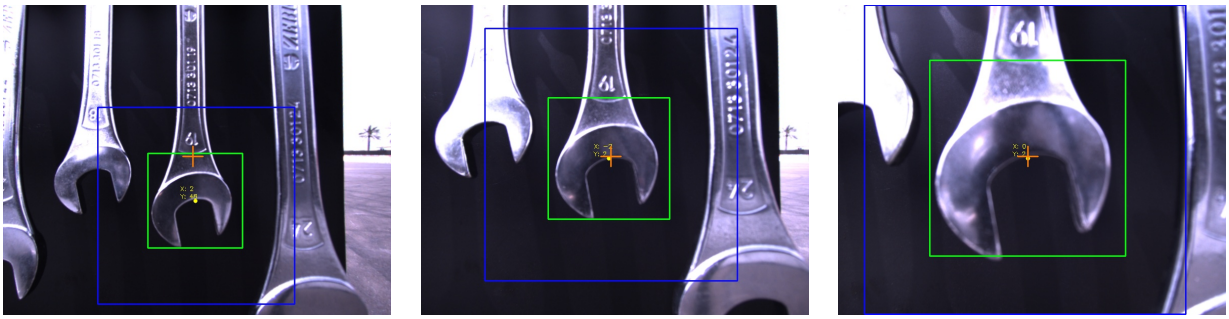
Figure 2.11: Wrench selection procedure: (a) Pin detection to estimate the camera roll angle. (b) Wrench length estimation through head and ring detection. (c) Selected wrench to be grasped (green) after the voting scheme.

components [39], as shown in Figure 2.10c. The extraction gives us the size of the valve in pixels (d_{image}). By using a calibrated camera [40] with a known focal length (f_{cam}), we determine the distance between the valve and the camera (z_{depth}) using the known size of the valve in metric standards (d_{world}) via the projective equation

$$z_{\text{depth}} = \frac{d_{\text{world}}\, f_{\text{cam}}}{d_{\text{image}}} \,. \tag{2.2}$$

2.4.2 Wrench Selection

The wrenches hang in random order on the pins but have specified dimensions. In order to operate the valve stem, the appropriate wrench has to be selected according to its slot width. The wrench norm allows the correlation between slot width



(a) (b) (c)

Figure 2.12: Progression of wrench grasping: (a) The wrench head is detected (green) within an dynamic region of interest (blue). (b) The camera center is aligned with the wrench head. (c) The endeffector approaches the wrench while staying centered.

and tool length. We inspect the hanging wrenches from a single viewpoint where all of them are visible. This position can be calculated based on the panel layout, camera lens parameters (perspective view angle), and the estimated valve position. From this view, the wrench rings are detected using the mentioned SVM detector, and the center points of the rings are accumulated from consecutive image frames. Once enough points are recorded, a line is fit over those points, as shown in Figure 2.11a, and its angle from the horizontal is calculated to correct the camera roll. To be robust against the positioning uncertainty of the UGV, we again estimate the depth from the wrench hanging pins with a similar approach as described above. From the corrected viewpoint, the head and ring of every wrench are detected using the trained SVM models. The distance (in pixels) between the mid-points of the wrench's detected ends is calculated, as shown in Figure 2.11b, and converted to metric length. To be robust against swinging of the wrenches due to wind, we use a voting scheme over several frames. In each image frame, a wrench is selected and voted for based on its estimated length. The candidate with the most votes will be finally grasped, as shown in Figure 2.11c.

2.4.3 Wrench Grasping

Once the appropriate wrench is selected, it is approached by the gripper using visual servoing. Our method uses both Position-based Visual Servoing (PBVS) and IBVS [41], [37] control techniques. First, we use the estimated depth and a PBVS approach to move the gripper close to the selected wrench, as shown in Figure 2.12a.

From there, IBVS is used to align the gripper with the wrench before grasping. During alignment, the gripper further approaches the wrench, as shown in Figure 2.12b. An SVM model trained for the wrench head along with a Kalman Filter detect and track the wrenches.
While approaching, we calculate the ratio between the size of the wrench head and the size of the image frame. As we move closer to the panel, this ratio increases and we can detect when the gripper is at the proper distance to grasp the wrench, as shown in Figure 2.12c. A first-order low pass filter on the ratio is used to avoid false detection due to the movement of the wrenches in the presence of wind.

Once the gripper has approached the wrench, the clamps close partially to bound the wrench within them but can still move up and down. Then the gripper is lowered such that the wrench head is aligned to the rotation axis of the last actuator and does not occlude the camera view. The clamps close entirely with a constant force, and the final clamp position is used to determine if the grasping was successful.

2.4.4 Evaluation

The object detection and visual servoing proved to be reliable during the MBZIRC. The primary failure source was the changing lighting conditions, which could lead to loss of target tracking. To address this, we adjusted the exposure of the camera before each trial. Table 2.2 shows the performance of the wrench ring and head detection in both trials of the grand challenge combined. As a performance measurement, we use the detection rate (or *recall*) [42]:

$$\text{Detection Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2.3}$$

where TP and FN are the *true positive* and *false negative* object count. The wrench classifier shows a high detection rate for both rings (92.5 %) and heads (95.42 %), and only a few false positive counts for the wrench ring (3.06 %).
Another challenge was the shaking motion of the wrenches due to wind gusts, which is problematic for wrench length estimation and grasping. To make the wrench selection robust, we implemented a probabilistic voting scheme comparing the length of the detected wrenches over several frames. Due to the shaking motion, we had to approach the panel slowly and filter the distance measurement

Table 2.2: Performance of the wrench detection during the grand challenge.

Classifier	Samples	Detection Rate	True Positive	False Positive
Wrench Ring	240	92.50 %	96.94 %	3.06 %
Wrench Head	240	95.42 %	100.0 %	0.0 %
Wrench	240	95.42 %	100.0 %	0.0 %

to make sure we close the gripper at the right moment. This careful approach allowed the successful recognition and grasping of the correct wrench in both trials.

Generally speaking, we focused on robust and reliable object detection and tracking in a trade-off for slower execution speed. Especially while measuring the distance to the wrench panel, we had an extended downtime. To further improve the visual servoing, faster detection of target objects would be necessary.

2.5 Manipulator Control

For the major part of the manipulation task, it is sufficient to follow the desired end-effector pose $\boldsymbol{p}_{\text{ee}}^{\text{des}}$ or twist trajectory $\boldsymbol{w}_{\text{ee}}^{\text{des}}$, especially when the end-effector moves without interactions during the visual servoing phase. However, the ability to precisely measure and control contact forces and torques demonstrates its usefulness during the interaction of the gripper with objects, for example, when handling the valve stem with a wrench. In our control formulation, we combine model-based feedback linearization with low impedance joint stabilization. The usage of SEAs adds inherent compliance to the system, improving safety and resistance against external shocks if the manipulator happens to collide with an obstacle.

2.5.1 System Model

The joint space dynamic model of the ANYpulator system can be expressed by

$$\boldsymbol{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{b}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{J}_{\text{ee}}(\boldsymbol{q})^{\top}\,\boldsymbol{\lambda}_{\text{ee}}, \tag{2.4}$$

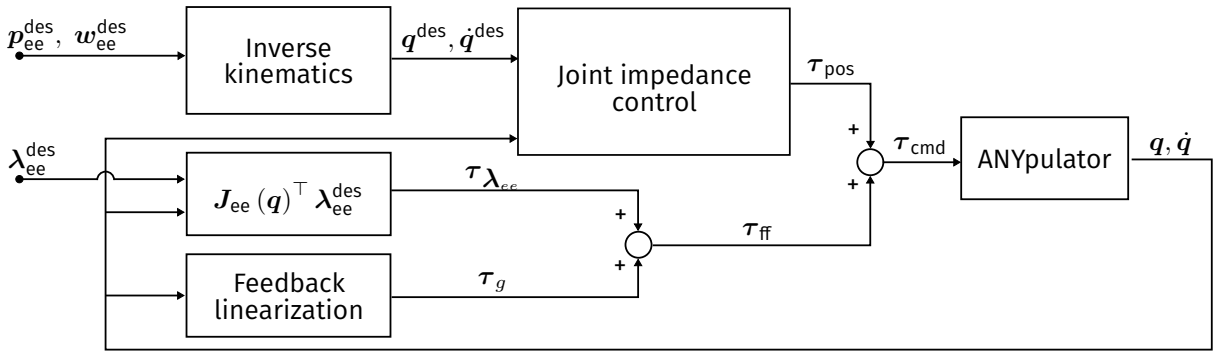


Figure 2.13: Control architecture for the ANYpulator. The feedback linearization reduces to gravity compensation for the slow movements needed during the challenge task.

as a function of the joint positions $\boldsymbol{q}$, velocities $\dot{\boldsymbol{q}}$, and accelerations $\ddot{\boldsymbol{q}}$. The symmetric mass matrix $\boldsymbol{M}$ encodes the system's inertia, $\boldsymbol{b}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ incorporates the centrifugal and Coriolis terms, and $\boldsymbol{g}(\boldsymbol{q})$ represents the torque due to gravity. The term $\boldsymbol{J}_{ee}(\boldsymbol{q})^\top \boldsymbol{\lambda}_{ee}$ represents the influence of an external wrench $\boldsymbol{\lambda}_{ee}$ acting on the end-effector. This wrench is projected into the joint space through the associated Jacobian transposed $\boldsymbol{J}_{ee}(\boldsymbol{q})^\top$. For typical applications, we want to manipulate objects with the end-effector. We thus assume external wrenches only appear there. Finally, the term $\boldsymbol{\tau}$ represents the torque applied by the actuated joints.

2.5.2 Control Formulation

The task of the control loop is to bring the end-effector pose $\boldsymbol{p}_{ee}$ to its desired pose $\boldsymbol{p}_{ee}^{des}$. The control reference is converted to desired joint positions $\boldsymbol{q}^{des}$ and velocities $\dot{\boldsymbol{q}}^{des}$ through inverse kinematics, as outlined in Section 2.5.3. A simple PID controller, running on the integrated motor control unit, generates the actuator torque $\boldsymbol{\tau}_{pos}$. To allow low PID controller gains for compliance and still achieve accurate and fast position tracking, we add model-based feed-forward torques $\boldsymbol{\tau}_{ff}$, which compensate for gravitational terms and task-specific end-effector wrenches $\boldsymbol{\lambda}_{ee}^{des}$. The resulting motor torque command $\boldsymbol{\tau}_{cmd}$ is given by

$$\boldsymbol{\tau}_{cmd} = \boldsymbol{\tau}_{pos} + \underbrace{\boldsymbol{\tau}_{\boldsymbol{\lambda}_{ee}} + \boldsymbol{\tau}_g}_{\boldsymbol{\tau}_{ff}}, \tag{2.5}$$

$$\boldsymbol{\tau}_{\boldsymbol{\lambda}_{ee}} = \boldsymbol{J}_{ee}(\boldsymbol{q})^\top \boldsymbol{\lambda}_{ee}^{des}. \tag{2.6}$$

2.5.3 Inverse Kinematics and Singularity Handling

A key component of the controller is solving the Inverse Kinematics (IK) problem for varying pose references $\boldsymbol{p}_{\text{ee}}^{\text{des}}(t)$. We use an iterative IK approach, running in a dedicated thread at a loop rate of $f = 1000\,\text{Hz}$. At each iteration, we compute the translational and rotational error $\boldsymbol{e}\,(t)$ between the desired pose and the pose from the IK solution. This error can be interpreted as a twist in operational space, which we convert back to joint space through a pseudo-inverse $\boldsymbol{J}_{\text{ee}}^{\dagger}$ of the end-effector Jacobian.

Inverse kinematics can become unstable and produce fast motions when the joint positions are in the vicinity of a singular configuration. A standard solution to avoid such dangerous situations is to reject control references parallel to singular directions in task space [43]. We achieve this by monitoring the magnitude of the Jacobian's singular values. If a singular value σ_i is below a minimal threshold $\sigma_{\text{min}} > 0$, it is assigned zero, thereby reducing the rank of the Jacobian and its pseudo-inverse. We may write

$$\boldsymbol{J}_{\text{ee}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}\,, \tag{2.7}$$

$$\boldsymbol{J}_{\text{ee}}^{\dagger} = \boldsymbol{V}\tilde{\boldsymbol{\Sigma}}\boldsymbol{U}^{\top}\,, \tag{2.8}$$

where the matrix $\boldsymbol{\Sigma} = \text{diag}\,\{\sigma_i\}$ holds the singular values of $\boldsymbol{J}_{\text{ee}}$, and we define

$$\tilde{\boldsymbol{\Sigma}} = \text{diag}\,\{\tilde{\sigma}_i\} \;\text{ with }\; \tilde{\sigma}_i = \begin{cases} \frac{1}{\sigma_i} & \text{if } \sigma_i > \sigma_{\text{min}}\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.9}$$

In effect, the system cannot produce any control action in a singular direction. The reduced dimension of the reference produces a rank-deficient Jacobian with a non-empty null space. Joint velocities projected onto this null-space have no impact on the end-effector velocity in non-singular directions and can be used for internal reconfiguration. We choose these reconfiguration velocity commands as

$$\boldsymbol{e}_{\text{null}}^{\text{des}} = \left(q_{\text{SH ROT}}^{\text{des}} - q_{\text{SH ROT}},\; q_{\text{SH FLE}}^{\text{des}} - q_{\text{SH FLE}},\; q_{\text{FLE}}^{\text{des}},\; q_{\text{FLE}}^{\text{des}},\; 0,\; 0\right)^{\top}\,, \tag{2.10}$$

$$\text{with } q_{\text{FLE}}^{\text{des}} = q_{\text{SH FLE}}^{\text{des}} + q_{\text{EL FLE}}^{\text{des}} + q_{\text{WR FLE}}^{\text{des}}\,, \tag{2.11}$$

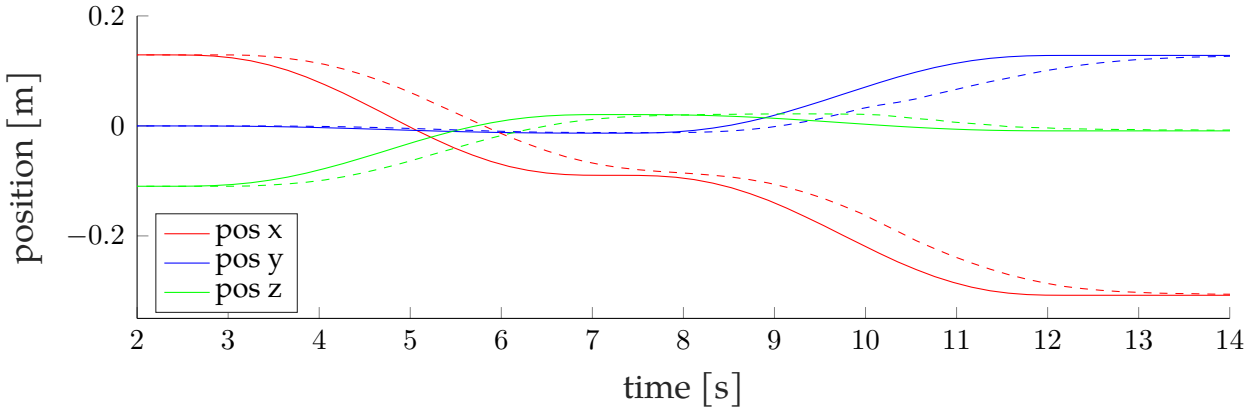


Figure 2.14: Position tracking of task-space end-effector trajectories. The solid line shows the reference position in the task space, the dashed line the measured position. The delay originates from filtering the desired joint positions *after* computing them from the desired end-effector pose; it is not a controller shortcoming.

to drive the system towards a fixed desired joint configuration when there are multiple solutions in singular configurations. The desired shoulder, elbow, and wrist flexion/extension configurations are thereby computed such that the final wrist link remains upright. The reconfiguration joint velocities are added in the null space to update the desired joint position and velocity as follows

$$\dot{\boldsymbol{q}}^{\text{des}}(t) = k_P \left(\boldsymbol{J}_{\text{ee}}^{\dagger} \boldsymbol{e}(t) + \left(\boldsymbol{I} - \boldsymbol{J}_{\text{ee}}^{\dagger} \boldsymbol{J}_{\text{ee}} \right) \boldsymbol{e}_{\text{null}}^{\text{des}}(t) \right) , \tag{2.12}$$

$$\boldsymbol{q}^{\text{des}}(t) = \int \dot{\boldsymbol{q}}^{\text{des}}(t) \, \mathrm{d}t , \tag{2.13}$$

with k_P as constant proportional gain.

2.5.4 Experimental Validation

Figure 2.14 shows the execution of a straight-line end-effector trajectory in free space. The desired joint positions and velocities with corresponding feed-forward torques update at a time step of 0.007 s. To ensure stable and smooth motion, we filtered the desired joint position with a first-order low-pass filter. This explains the delay in tracking the desired end-effector position, whereas the joint position tracking is accurate (Figure 2.15 bottom). The smoothing of the joint reference trajectory was necessary to avoid exciting the mobile base's resonance frequency, causing undesired vibrations in the whole system.

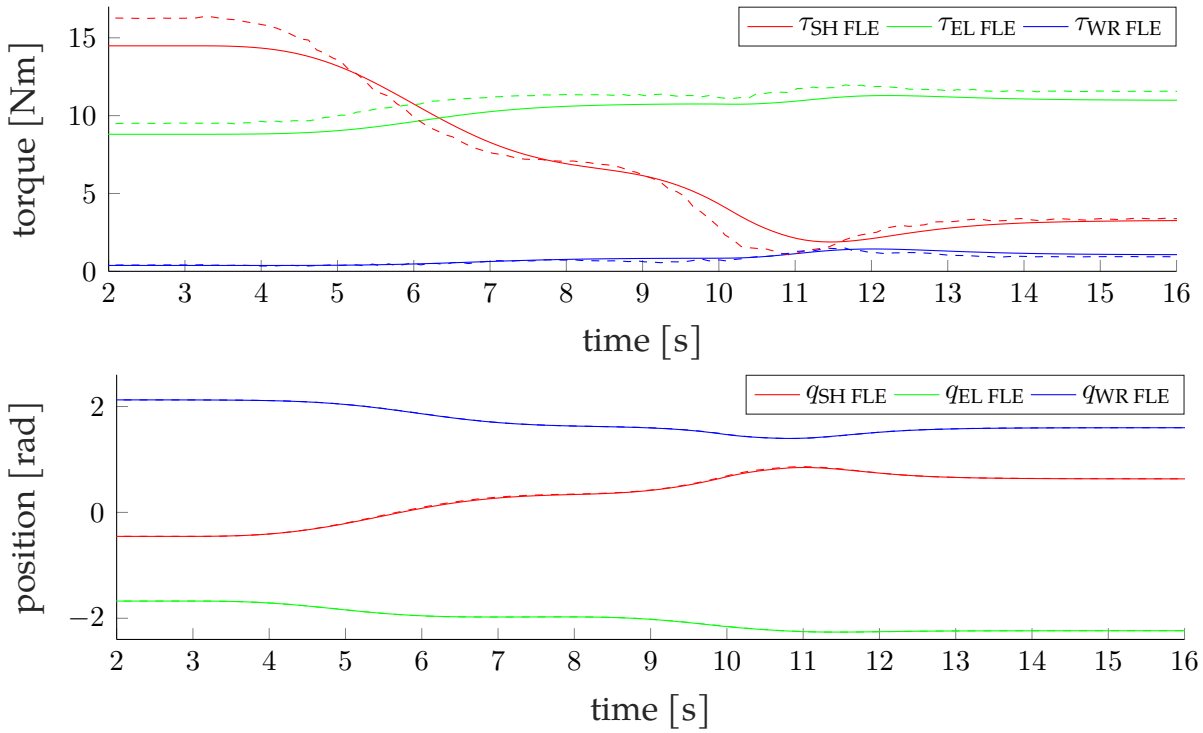


Figure 2.15: Top: Comparison of the feed-forward torque $\boldsymbol{\tau}_{\text{ff}}$ (dashed), which compensates gravitational terms $\boldsymbol{g}(\boldsymbol{q})$ and desired end-effector wrench $\boldsymbol{\lambda}_{\text{ee}}$, and the actual measured torque (solid) for the shoulder, elbow, and wrist flexion / extension joint (SH FLE, EL FLE, WR FLE). Bottom: Joint position tracking with desired (solid) and measured (dashed) joint position.

In the case of slow motions, the required actuator torque comes mainly from the gravitational terms $\boldsymbol{g}(\boldsymbol{q})$ and applied external wrenches $\boldsymbol{\lambda}_{\text{ee}}$. Feedback linearization, therefore, reduces to gravity compensation. Figure 2.15 (top) shows that the feed-forward torque tracks the actual measured torque of the actuators for the same motion as in Figure 2.14. This allows setting the gains of the motor PID controller relatively low while the actuators are still able to accurately track the desired joint position (Figure 2.15 bottom).

2.5.5 Wrench Manipulation

To complete the manipulation task, we need to ensure that the grasped wrench is precisely positioned inside the gripper. To this end, we first turn the gripper upside down, open the claws slightly to let the wrench head slide to the mechanical end-stop, and close the gripper again, ensuring that the wrench head is in the correct

position for engagement. Subsequently, the engaging procedure is executed in two stages:

Valve engaging: First, the gripper is placed at a safe distance to the valve such that the wrench head opening is pointing towards the valve stem edge diagonally (Figure 2.16a). Next, we establish contact between the wrench and the valve stem by opening the gripper slightly, which causes the wrench to slide down by gravity and hit the valve stem. To engage the gripper, we move such that the wrench head is rotating around the stem center (Figure 2.16b). The gripper stays slightly opened during this motion to allow engagement of the wrench and prevent jamming due to induced forces from the manipulator. The wrench will eventually slip onto the valve by gravity, and successful engagement can be detected by an increase in torque needed to turn the valve (Figure 2.16c). For that, we monitor the difference between actual and first-order filtered torque of the last joint and trigger engagement above a threshold of 0.18 Nm (see Figure 2.17). After engaging, the gripper is closed again for the subsequent valve turning.

Valve rotation: We have to make sure to stay engaged during a full 360 ° rotation of the valve. For this purpose, the manipulator applies a constant force of 10 N to push the wrench towards the stem center during valve rotation (Figure 2.16d). Additionally, according to the competition specifications,a torque of approximately 5 Nm is required to operate the valve stem. The resulting desired end-effector wrench $\boldsymbol{\lambda}_{\text{ee}}$ is directly applied as feed-forward torque $\boldsymbol{\tau}_{\boldsymbol{\lambda}_{\text{ee}}} = \boldsymbol{J}_{\text{ee}}(\boldsymbol{q})^{\top} \boldsymbol{\lambda}_{\text{ee}}$. Figure 2.18 shows the commanded and measured torque of the manipulator's last joint during valve stem operation with a rotation speed of 0.25 rad/s. Noticeably, the actually required torque for rotation is less than 1 Nm. An additional position tracking ensures that the valve is turned in a controlled fashion.

2.5.6 Evaluation

The compliant design of the manipulator showed its advantages in terms of robustness during the entire competition, especially at impacts. Notwithstanding, the main advantage of this design was its adaptability during valve manipulation.

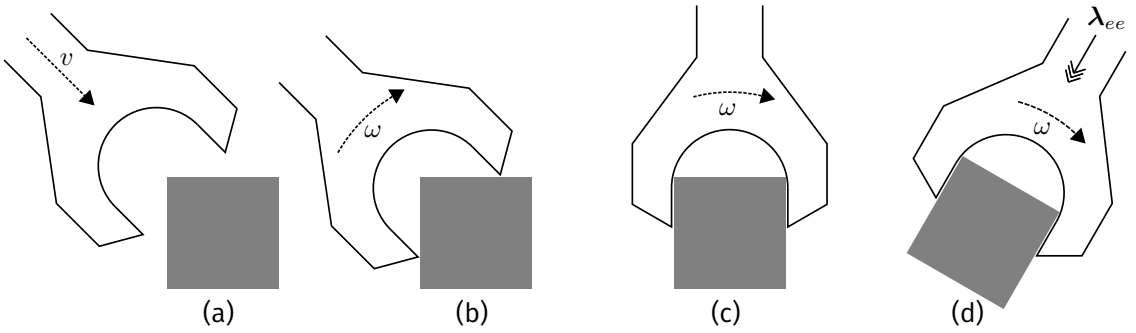


Figure 2.16: Engagement and rotation of the valve stem (grey box). The wrench head approaches diagonally (a), and a rotation ω is initiated (b). The wrench will eventually slip in by gravity (c) and starts rotating the stem (d). To avoid slipping off, a force $\boldsymbol{\lambda}_{ee}$ pointing to the valve stem is applied.

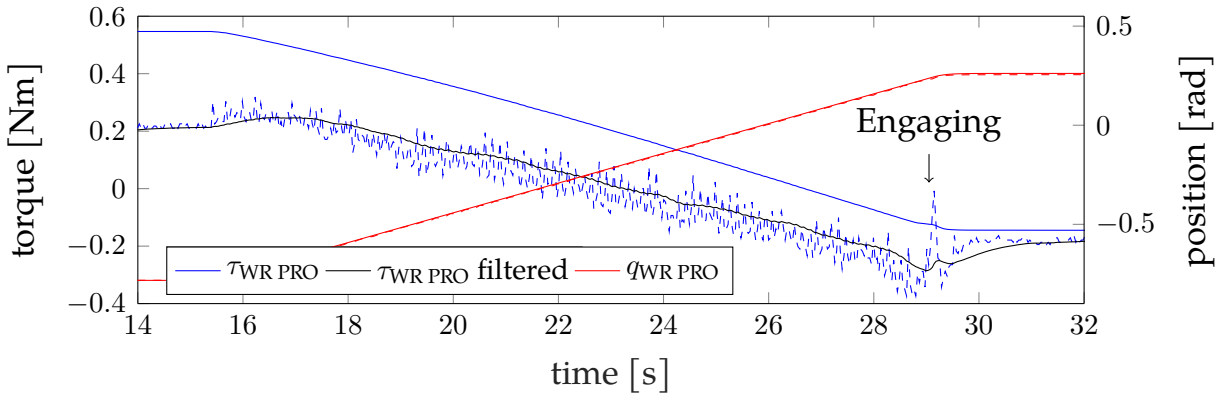


Figure 2.17: Torque τ (blue) and position q (red) tracking during valve engaging for the manipulator's last joint (WR PRO). The solid lines show the commanded state, the dashed line the actual joint position and torque. Shown in black is the filtered actual torque, which is used as reference to detect engagement.

This design allowed to overcome minor alignment errors and prevent the wrench from getting jammed on the valve stem, making the wrench engagement more reliable. On the downside, the compliance also led to oscillations and affected the end-effector precision and, thereby, the object detection during visual servoing. To overcome this, we reduced the motion speed during manipulation. The performance could be improved by more accurate estimation of the model parameters for better torque tracking, and more advanced control approaches considering the actuator dynamics (see Ch. 3).

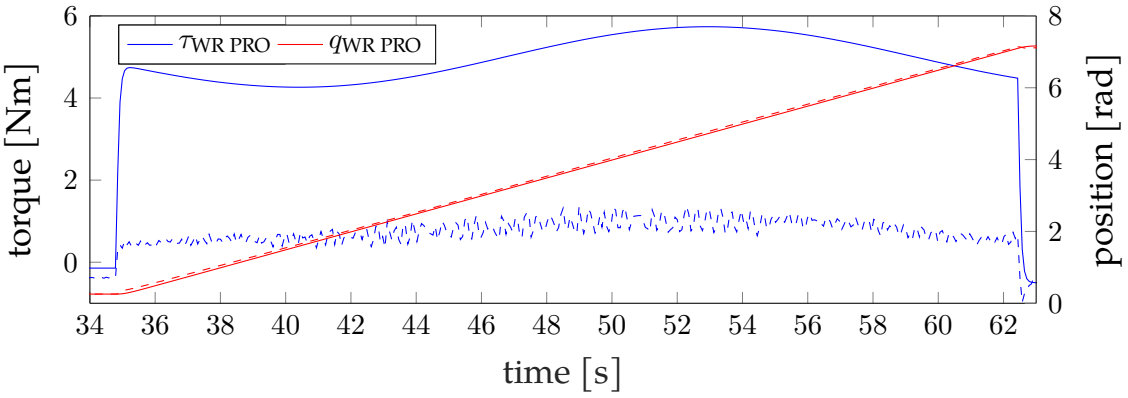


Figure 2.18: Torque τ and position q tracking during valve rotation for the manipulator's last joint (WR PRO). The desired end-effector torque (solid) corresponds to the specified 5 Nm resistance, but the actually required torque (dashed) corresponds to less than 1 Nm.

2.6 Challenge Performance and Discussion

This section critically evaluates the system as a whole and points towards areas of improvement: Both trials during the grand challenge showed similar performance regarding time consumption and fulfillment of the task[5]. The second attempt lasted approximately 327 s, of which 90 s were needed for exploration and navigation until the robot was positioned in front of the wrench panel, 95 s for measuring the valve stem and selecting the correct wrench, 110 s for grasping the wrench and engaging it with the valve, and the remaining 32 s for the actual rotation of the valve stem. During navigation, the main bottleneck was the slow average driving speed of approximately 0.3 $^{m}/_{s}$ due to the limited turning rate necessary to avoid stick-slip effects between wheels and ground[6]. During manipulation, the end-effector motion must be slow and steady for reliable and accurate visual tracking. Especially during grasping, careful approaching was necessary because of the shaking wrenches due to wind gusts. Unfortunately, official statistics about the competition have not been released by the organizers and make a comparison with other teams difficult.

The most fragile component in the mission was the precise estimation of the panel position. A solution based on a simpler planar 2D LiDAR would probably have introduced less delay and higher accuracy for this task. A related weak point

5 Watch the accompanying video about the grand challenge: https://youtu.be/iUPJ73Y5yMw
6 The wheels and the skid-steer design are made for rough terrain applications.

identified during the trials before the grand challenge was the transition between navigation and manipulation. Precise positioning in front of the panel is vital for manipulation, as the arm has to reach both the valve stem and the wrenches. Because navigation and manipulation were mainly tested separately before the competition, this flaw was not realized until the actual trials and not covered by a recovery behavior. In the future, we plan to overcome this limitation by generating joint arm-base motion references as proposed in Gawel *et al.* [44]. Another common failure source was the loss of target tracking during visual servoing because of changing lighting conditions. In order to recover from tracking loss, the state machine moved the arm to the last reference pose and restarted the tracking. Fortunately, adjustments to the camera exposure settings could be made just before a challenge trial.

In the development phase, we emphasized the system's ability to recover independently from failure cases. Suppose a fault is detected, like loss of target tracking during visual servoing. In that case, the state machine restarts the current (and possibly previous) task. If the task cannot be repeated, the autonomous mission prompts the operator for an intervention (which never happened during the challenge). Furthermore, we aimed for platform-agnostic mission control and operator interaction to promote re-usability in future applications.
The autonomy of our system independent of the operator's computer proved crucial, as the quality of the wireless connection varied during a mission and, for example, did not allow for a reliable image stream. The operator's single view visualization turned out to be suboptimal during the challenge since he was not allowed to touch the computer at all, thereby preventing changes to the map's zoom level. The localization solution worked robustly and even allowed us to eliminate GPS during the challenge since the starting points inside the field were predefined. The produced maps and recorded data stream allowed us to visualize each run nicely in a replay.

To conclude, our fundamental lessons learned from preparing and participating in MBZIRC are: (1) early testing in a realistic setting (with good mock-ups and a rapidly deployable system) is crucial; (2) system integration should have a higher priority than fully-fledged individual components; and (3) the team should have redundancy such that more than one person can resolve problems with a given hardware or software component.

2.7 Summary

In this chapter, we presented the mobile manipulation platform mANYpulator, capable of autonomously executing interaction tasks relying solely on its onboard sensing. We successfully demonstrated the applicability of our system by accomplishing the entire mission during both trials of the grand challenge of MBZIRC in autonomous mode. Solving a task like MBZIRC always entails a trade-off between finding innovative general solutions and using a particular but straightforward approach to a given task. We challenged ourselves to develop a generic autonomous system that can be deployed for various applications in real-world scenarios and as a research platform. Our mapping and localization suite showed very robust performance with minimal error and can be used without changes for other tasks, particularly in GPS denied environments. A stable localization estimate was essential for reliably detecting and positioning the robot in front of the tool panel. The panel detection module, in effect, tackled a problem that could have been solved more efficiently with a high resolution planar 2D LiDAR but may, in turn, be generalized to finding more generic objects. Coupled with our explorer, one could also extend the search space to arbitrarily shaped areas. The visual servoing algorithm worked well in practice and, given newly trained object classifiers, may also detect other kinds of objects. Opting for a visual serving solution allowed us to decouple base positioning errors from end-effector tracking performance. Finally, the low-impedance manipulation control helped mitigate misalignments of the end-effector that are hard to estimate from the monocular camera image. The compliant arm prevented jamming during wrench insertion and made the valve manipulation highly reliable.

3

Actuator-Aware Model Predictive Control for Dynamic Manipulation

Manipulators are more and more deployed to environments where they have to interact with previously unknown objects and perform tasks within not well-defined surroundings. They have to deal with high uncertainties and adapt to collaborators like other robots or humans. Those tasks require making and breaking contacts with the environment, allowing a gentle interaction while still applying significant forces, which are conditions classical rigid robot arms are not best suited.

Therefore, soft manipulators are increasingly popular in research. Soft manipulators are composed of materials or use actuation modes that are flexible and soft. Their fundamental principle, compliance, allows them to exploit the interaction between the robot and the environment. The inherent softness of these systems provides adaptability, robustness, and safety, enabling new tasks in manipulation. A way to achieve compliance in the actuation mode is the usage of compliant actuators, like SEAs. The advantage of such actuators is their torque-controllability and their inherent robustness to impacts. On the other hand, the elastic elements introduce unwanted intrinsic oscillatory dynamics to the system, cause underactuation, and reduce the system's natural frequency. Furthermore, by design, the actuators often have low friction and damping to improve efficiency. It is possible to exploit the intrinsic actuator dynamics for efficient cyclic motion tasks, but it is required to address them appropriately to achieve precise positioning at the same time. Soft manipulators are still required to move a varying payload precisely through the entire workspace, like their stiff counterparts. However, the combination of compliance with accurate and dynamic motion is still an unsolved challenge in robotics.

In this work, we provide a detailed insight into how different control approaches change the stiffness of a compliant manipulator. We propose using MPC to incorporate the actuator model to compute optimal feedforward plans in correspondence with sensed changes of the state. Using such a control policy, we can show that the natural stiffness of the robot alters only in proportion to the time step of MPC. We thus can perfectly preserve the inherent compliance of the system using a suffi-

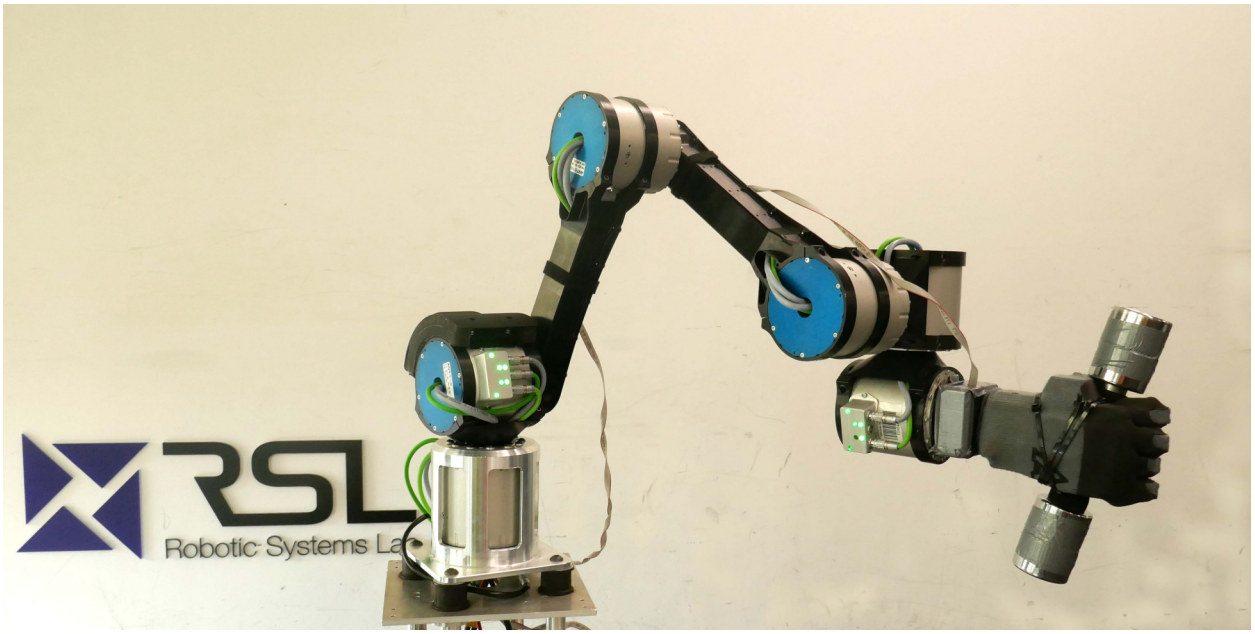


Figure 3.1: The 6 DoF robotic arm ANYpulator is composed of ANYdrive SEAs. Attached to the end-effector as payload is a dumbbell.

ciently fast MPC update loop for control. All the presented findings are empirically validated on the ANYpulator platform (see Figure 3.1).

3.1 Related Work

There exist classical control approaches like high-gain control, feedback linearization [45], and backstepping [46] to solve the tracking problem for compliant manipulators. An adaptive backstepping control approach has been introduced in [47, 48] for controlling hydraulic manipulators. A practical issue in backstepping control and feedback linearization is the need for higher-order state derivatives, which is problematic as real-world sensors are subject to noise. A robust control approach based on disturbance observers is proposed by [49] and [50] for controlling the end-effector force and position considering the actuator dynamics. Della Santina *et al.* [51] show that all of these classical approaches have a common problem; they replace the elasticity of the original plant dynamics with a desired one, objecting to the idea of carefully designing actuator compliance. Keppler *et al.* [52] report from practical experiences that approaches modifying the elastic behavior often fail in practice because they have limited robustness to unmodeled dynamics, parameter uncertainties, actuator bandwidth, and amplitude limitations. Therefore, recent

research is focusing on preserving the natural dynamics of the robot. Inspired by human observations, [51, 53] use iterative learning control (ILC) to learn a feedforward (anticipative) policy and combine it with low-gain feedback. The feedforward component does not depend on the state but only on the reference, therefore not altering the system's compliance. Alternatively, [52] presents an elastic structure preserving (ESP) control approach that tries to minimize the dynamic shaping of the internal elastic transmission. By performing a gain analysis, they show that the plant dynamics are changed significantly less with their method than for feedback linearization-based full state feedback control. However, they do not provide details on how much the actual stiffness of the system is changed. Furthermore, they extend their work by adding a desired link-side impedance without altering the inherent elastic structure [54] and applying it to visco-elastic actuators [55].

3.2 System Modeling

We performed the experiments using the 6 DoF robotic arm, ANYpulator (see Section 2.1.2), composed of ANYdrive SEAs. These actuators can be operated in different modes like joint position control, joint velocity control, joint torque control, or motor velocity control. They provide full feedback of the joint state and the internal actuator state at the control frequency.

The Rigid Body Dynamics (RBD) equation of motion for a torque-controllable robot arm is given in Eq. (2.4). We want to manipulate objects with the end-effector for typical applications, and we assume that external wrenches $\boldsymbol{\lambda}_{\text{ee}}$ only appear there. The state-vector $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})$ of the system is composed of the joint positions and velocities and the control input-vector is given by $\boldsymbol{u} = (\boldsymbol{\tau}, \boldsymbol{\lambda}_{\text{ee}})$, i.e., joint torques and the end-effector wrench are used as input.

3.2.1 Rigid Body and Actuator Modeling

The RBD from Eq. (2.4) assume to have a *perfect torque source* as input. Being a torque source is a simplified assumption for all actuators, neglecting the internal actuator dynamics. For SEAs, the transients of the internal actuator dynamics are dependent on the link-side inertia and the external load, and thus, for a multi-link robot, they are configuration-dependent. This chapter addresses this fact by includ-

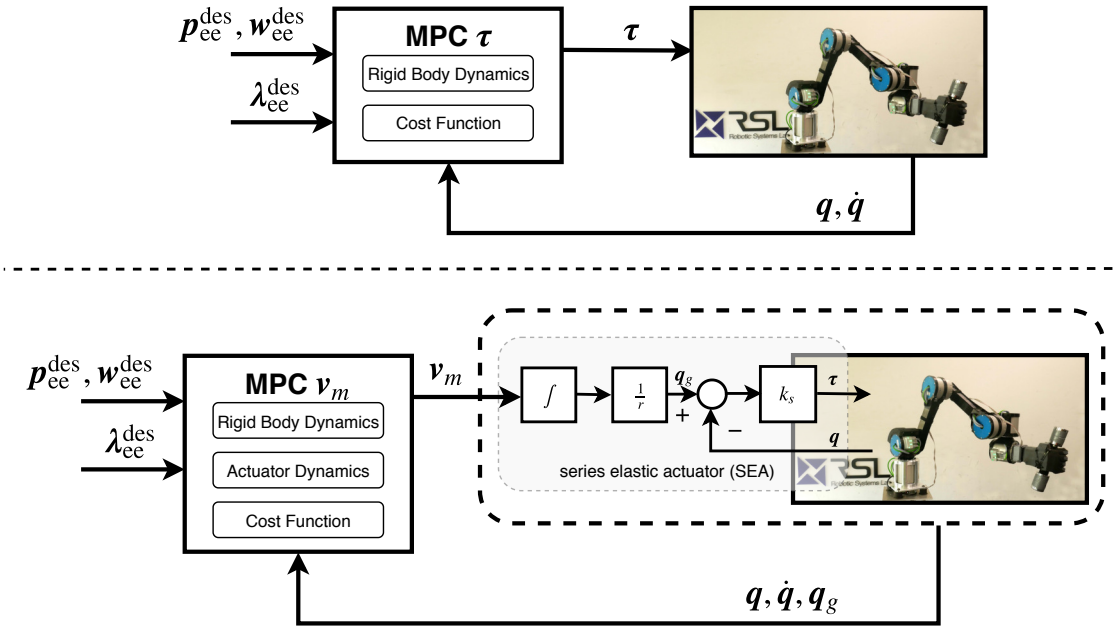


Figure 3.2: MPC control structure overview for tracking the end-effector reference given by pose p_{ee}^{des}, twist w_{ee}^{des}, and wrench λ_{ee}^{des}. The MPC with torque input (MPC τ, top) directly sends torque commands τ to the system, whereas MPC with motor velocity input (MPC v_m, bottom) considers the dynamics of the series elastic actuators and sends actuator motor velocities v_m to the system.

ing a minimal actuator model in the overall system dynamics. The series elastic actuators of our manipulator are composed of a high torque electric motor, a harmonic gear, and a rotational spring as the elastic element in series at the joint output. Joint output position q and gear position q_g are measured through high-resolution encoders. The output torque is calculated by the deflection of the spring

$$\tau = k_s(q_g - q), \tag{3.1}$$

using the spring stiffness k_s. In order to include the actuator model in the control formulation, we augment the state-vector with the gear position of the actuator: $x = (q, \dot{q}, q_g)$. The gear velocity $\dot{q}_g$ is given by $\dot{q}_g = \frac{1}{r} \times v_m$, where r is the gear ratio of the SEA and v_m is the motor velocity. The input-vector is defined as $u = (v_m, \lambda_{ee})$, i.e., motor velocity and the end-effector wrench. Note that we here assume that the SEA's motor is a *perfect velocity source*, and we, therefore, choose the motor velocity v_m as input instead of joint torque. Although this is still a simplification, the step-response of the internal motor velocity control loop is significantly faster than the torque-loop, as it has one integrator less [56].

3.3 Control Method

In this section, we introduce our proposed control approach. We first discuss the baseline controller, which tracks the end-effector reference trajectory based on an inverse dynamics approach. We then introduce our MPC framework used for task-space control of ANYpulator. We discuss two formulations of the MPC based on the models presented in Section 3.2.

3.3.1 Baseline Controller – Inverse Dynamics

As a baseline controller, we use the task-space formulation of commonly employed inverse dynamics control. A more detailed overview of task-space control (or operational-space control) is given by [57]. The goal is to track the desired end-effector motion given by the desired pose $\boldsymbol{p}_{\text{ee}}^{\text{des}}$ and twist $\boldsymbol{w}_{\text{ee}}^{\text{des}}$. The rotational and linear acceleration of the end-effector, $\dot{\boldsymbol{v}}$ and $\dot{\boldsymbol{\omega}}$, are coupled to the joint acceleration by

$$\dot{\boldsymbol{w}}_{\text{ee}}^{\text{des}} = (\dot{\boldsymbol{v}}, \dot{\boldsymbol{\omega}}) = \dot{\boldsymbol{J}}_{\text{ee}}\dot{\boldsymbol{q}} + \boldsymbol{J}_{\text{ee}}\ddot{\boldsymbol{q}}. \tag{3.2}$$

Solving Eq. (3.2) for $\ddot{\boldsymbol{q}}$ and substituting in Eq. (2.4) results in

$$\dot{\boldsymbol{w}}_{\text{ee}}^{\text{des}} = \dot{\boldsymbol{J}}_{\text{ee}}\dot{\boldsymbol{q}} + \boldsymbol{J}_{\text{ee}}\boldsymbol{M}^{-1}(\boldsymbol{q})\left(\boldsymbol{\tau} + \boldsymbol{J}_{\text{ee}}(\boldsymbol{q})^{\top}\boldsymbol{\lambda}_{\text{ee}} - \boldsymbol{b}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q})\right), \tag{3.3}$$

which can be solved for the desired input torque $\boldsymbol{\tau}$. As a control strategy for the desired acceleration $\dot{\boldsymbol{w}}_{\text{ee}}^{\text{des}}$, we apply a PID controller to track the desired pose and twist.

3.3.2 MPC Formulation

We formulate our control problem as a Nonlinear Optimal Control (NLOC) problem that can include both, the RBD of the arm and the actuator dynamics, and solve it in an MPC fashion. The finite-horizon NLOC problem is in the general form

$$\begin{aligned} \min_{\boldsymbol{u}(\cdot)} \left\{ \Phi\left(\boldsymbol{x}\left(t_f\right)\right) + \int_{t=0}^{t_f} L\left(\boldsymbol{x}\left(t\right), \boldsymbol{u}\left(t\right), t\right) dt \right\} \\ \text{s.t.} \quad \dot{\boldsymbol{x}}\left(t\right) = \boldsymbol{f}\left(\boldsymbol{x}\left(t\right), \boldsymbol{u}\left(t\right), t\right) \\ \boldsymbol{x}\left(0\right) = x_0, \end{aligned} \tag{3.4}$$

with the state and control trajectory $\boldsymbol{x}\left(\cdot\right)$ and $\boldsymbol{u}\left(\cdot\right)$. Let $L\left(\cdot\right)$ be the intermediate cost at time t, $\Phi\left(\cdot\right)$ the terminal cost at the final time t_f, and $\boldsymbol{f}\left(\cdot\right)$ the nonlinear system dynamics.

Common approaches to solve NLOC problems are Single Shooting, Multiple Shooting, and Direct Collocation [58]. In order to achieve a performant MPC loop, the NLOC problem has to be solved at a sufficiently high frequency. For this reason, we employ an optimized solver that implements a multiple shooting approach using an iterative Gauss-Newton NLOC algorithm [59]. The solver utilizes a first-order method to approximate the NLOC problem locally as a Linear-Quadratic (LQ) optimal control problem using a Gauss-Newton Hessian approximation. The LQ problem is solved by a Riccati-based solver that has linear complexity in the time horizon, making this approach efficient for long time horizons [60].

The MPC strategy is based on a real-time iteration scheme introduced by [61]. This approach regards the entire time horizon and performs only one iteration of the NLOC, before it iterates towards the rigorous optimal solutions during the runtime of the MPC loop. During all the experiments, the MPC strategies are updated with a fixed frequency of 250 Hz, and the actuator commands are set accordingly. Figure 3.2 provides an overview of the MPC schemes.

While both MPC strategies are based on a similar algorithm, the underlying formulation of the system dynamics and consequently the cost function definition are different. Figure 3.2 illustrates the primary distinction between the two MPC schemes: the difference in the state and input vectors definition. The MPC $\boldsymbol{\tau}$ only considers the RBD modeling and directly commands the joint torques. On the other

hand, the MPC $\boldsymbol{v}_m$ incorporates the actuator dynamics in addition to the RBD equation and directly designs for the motor velocities.

Cost Function

In this section, we describe the general cost function employed to track the end-effector motion. The reference motion is defined as pose, twist, and wrench trajectories in task-space.

Let $\boldsymbol{r}_{\text{ee}}$ be the end-effector position in the inertial frame. The orientation error describing the deviation between the desired and the actual orientation is defined as $\boldsymbol{e}_o \in \mathbb{R}^3$. Given the rotation matrices $\boldsymbol{R}_{\text{ee}}$ and $\boldsymbol{R}_{\text{ee}}^{\text{des}}$, describing the rotation that can be computed from the joint states and the desired rotation, we can express the rotation $\boldsymbol{R}_{\text{diff}}$ aligning $\boldsymbol{R}_{\text{ee}}$ with $\boldsymbol{R}_{\text{ee}}^{\text{des}}$ by

$$\boldsymbol{R}_{\text{diff}} = \boldsymbol{R}_{\text{ee}}^{\text{des}} \boldsymbol{R}_{\text{ee}}^{\top} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{3.5}$$

The orientation error between the two frames can be calculated through

$$\boldsymbol{e}_o = \boldsymbol{r} \sin\theta = \frac{1}{2} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}, \tag{3.6}$$

where θ and $\boldsymbol{r}$ are the angle and axis representation of the rotation $\boldsymbol{R}_{\text{diff}}$. Equation (3.6) only gives an unique relationship for $-\pi/2 < \theta < \pi/2$.

We define $\boldsymbol{w}_{\text{ee}} = (\boldsymbol{v}, \boldsymbol{\omega}) \in \mathbb{R}^6$ as the end-effector twist, where $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are the linear and rotational velocity respectively, and $\boldsymbol{\lambda}_{\text{ee}} = (\boldsymbol{f}, \boldsymbol{m}) \in \mathbb{R}^6$ is the wrench acting at the end-effector. We use the input vector $\boldsymbol{u} = (\boldsymbol{\nu}, \boldsymbol{\lambda}_{\text{ee}})$, where $\boldsymbol{\lambda}_{\text{ee}}$ are the applied end-effector wrenches and $\boldsymbol{\nu}$ can either be the actuator torques $\boldsymbol{\tau}$, or actuator motor velocities $\boldsymbol{v}_m$. Given the notation above, the end-effector task-space

tracking problem can be described through the following cost term for intermediate cost

$$\begin{aligned} L(\cdot) =& \|\boldsymbol{r}_{\text{ee}}^{\text{des}} - \boldsymbol{r}_{\text{ee}}\|_{\boldsymbol{Q}_{\text{pos}}}^2 + \|\boldsymbol{e}_o\|_{\boldsymbol{Q}_{\text{o}}}^2 + \|\boldsymbol{w}_{\text{ee}}^{\text{des}} - \boldsymbol{w}_{\text{ee}}\|_{\boldsymbol{Q}_w}^2 \\ &+ \|\boldsymbol{\nu} - \boldsymbol{\nu}_{\text{ref}}\|_{\boldsymbol{R}_{\boldsymbol{\nu}}}^2 + \|\boldsymbol{\lambda}_{\text{ee}}^{\text{des}} - \boldsymbol{\lambda}_{\text{ee}}\|_{\boldsymbol{R}_{\boldsymbol{\lambda}}}^2 \end{aligned} \tag{3.7}$$

and final cost

$$\Phi(\boldsymbol{x}(t_f)) = \|\boldsymbol{r}_{\text{ee}}^{\text{des}} - \boldsymbol{r}_{\text{ee}}\|_{\boldsymbol{Q}_{\text{pos}}^{\text{f}}}^2 + \|\boldsymbol{e}_o\|_{\boldsymbol{Q}_{\text{o}}^{\text{f}}}^2 + \|\boldsymbol{w}_{\text{ee}}^{\text{des}} - \boldsymbol{w}_{\text{ee}}\|_{\boldsymbol{Q}_w^{\text{f}}}^2. \tag{3.8}$$

The positive semidefinite weighting matrices $\boldsymbol{Q}_{\text{pos}}, \boldsymbol{Q}_{\text{o}}, \boldsymbol{Q}_w, \boldsymbol{Q}_{\text{pos}}^{\text{f}}, \boldsymbol{Q}_{\text{o}}^{\text{f}}$, and $\boldsymbol{Q}_w^{\text{f}}$ penalize errors from the desired end-effector motion trajectory. $\boldsymbol{R}_{\boldsymbol{\nu}}$ is a weight on the deviation of the reference input, and $\boldsymbol{R}_{\boldsymbol{\lambda}}$ penalizes end-effector wrench errors.

The reference input trajectory $\boldsymbol{\nu}$ is defined differently for the two MPC strategies. For the torque input MPC, the reference torque input is chosen to be the gravity term of the RBD, $\boldsymbol{\nu}_{\text{ref}} = \boldsymbol{\tau}_{\text{ref}} = \boldsymbol{g}(\boldsymbol{q})$, to not penalize the MPC for torques required for compensating ANYpulator's weight. For motor velocity input MPC, the reference motor velocity is chosen to be zero $\boldsymbol{\nu}_{\text{ref}} = \boldsymbol{v}_{m,\text{ref}} = \boldsymbol{0}$.

Because the task-space cost terms are neither linear nor quadratic with respect to states and control input, we utilize automatic-differentiation with CppAD [62] for computing the cost function derivatives $\frac{\partial L}{\partial \boldsymbol{x}}$, $\frac{\partial \Phi}{\partial \boldsymbol{x}}$ and $\frac{\partial L}{\partial \boldsymbol{u}}$, $\frac{\partial \Phi}{\partial \boldsymbol{u}}$ to obtain an efficient code necessary for a performant MPC implementation.

3.4 Stiffness Comparison

The following section discusses the impact of the opted feedback control law on the compliance of the system. It provides detailed insight into how the different control approaches alter the natural stiffness of the system. To evaluate the change of compliance, we use the joint-space stiffness matrix $\boldsymbol{K}_q$, which defines the rate at which the torques of a joint increase or decrease as the joint is deflected from its nominal position. We can compute the joint-space stiffness matrix by differentiating the torque input to the system with respect to the joint state

$$\boldsymbol{K}_q = -\frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{q}}. \tag{3.9}$$

Depending on the control law, the individual joint's stiffnesses are highly coupled, meaning that the stiffness matrix is non-diagonal. The task-space stiffness matrix $\boldsymbol{K}_c$ for the end-effector can be derived from the joint-space stiffness through the relationship $\boldsymbol{K}_q = \boldsymbol{J}_{\text{ee}}^{\top} \boldsymbol{K}_c \boldsymbol{J}_{\text{ee}}$.

In the case of torque control, we also have to consider the effect of possible joint-level feedback gains on the stiffness. A standard solution to recover an accurate tracking behavior and stabilize the motion for torque-controllable manipulators is to apply the calculated torque from a model-based control approach as feed-forward part τ_{ff} and use Proportional-Derivative (PD) feedback control on joint-level for position and velocity. For a more general overview of (joint-level) motion control, please refer to [63]. In our case, we apply PD feedback gains for the 3 DoF of the wrist, as friction and modeling errors have a more significant impact on their tracking performance. By making use of PD feedback gains, the desired torque being tracked by the actuator is computed by

$$\tau_{\text{des}} = \tau_{\text{ff}} + k_D \left(\dot{q}_{\text{des}} - \dot{q} \right) + k_P \left(q_{\text{des}} - q \right), \tag{3.10}$$

where k_P and k_D are the joint-level PD feedback gains. The desired joint-state is computed through inverse kinematics of the desired end-effector pose and twist for the inverse dynamics controller. In the case of MPC $\boldsymbol{\tau}$, the reference state is given by the MPC formulation that provides a state and input trajectory for the entire time horizon. Generally, local feedback gains are undesirable because first, they are additional parameters that have to be tuned, and second, they are adjusted to a nominal load and reference step.

Using the considerations above, we get the joint-space stiffness matrix by

$$\boldsymbol{K}_q = -\frac{\partial \boldsymbol{\tau}_{\text{des}}}{\partial \boldsymbol{q}} = -\frac{\partial \boldsymbol{\tau}_{\text{ff}}}{\partial \boldsymbol{q}} + \boldsymbol{k}_P, \tag{3.11}$$

where $\boldsymbol{k}_P = \text{diag}(k_{P,1}, \ldots, k_{P,6})$ is a diagonal matrix composed of the P feedback gain k_P (see Eq. 3.10) of each joint. We can see that the proportional gain of the PD feedback directly adds to the stiffness. For the inverse dynamics controller, we can obtain the term $\frac{\partial \boldsymbol{\tau}_{\text{ff}}}{\partial \boldsymbol{q}}$ through numerical differentiation by perturbing the joint state $\boldsymbol{q}$. In the case of the MPC $\boldsymbol{\tau}$, numerical differentiation of the MPC policy will be inaccurate [64]. Fortunately, the opted MPC solver, in addition to optimized trajectories, provides the sensitivity of the control input to state perturbation in the

form of a Linear-Quadratic Regulator (LQR) matrix gain. Thus, the term $\frac{\partial \boldsymbol{\tau}_{\text{ff}}}{\partial \boldsymbol{q}}$ can be readily computed using this sensitivity matrix.

In the case of motor velocity control (MPC $\boldsymbol{v}_m$), the joint-space stiffness can be computed based on the deflection of the SEA's spring as

$$\frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{q}} = -\frac{\partial}{\partial \boldsymbol{q}} \boldsymbol{k}_s(\boldsymbol{q} - \boldsymbol{q}_g) = -\boldsymbol{k}_s(\boldsymbol{I}_6 - \frac{\partial \boldsymbol{q}_g}{\partial \boldsymbol{q}}) \approx -\boldsymbol{k}_s\big(\boldsymbol{I}_6 - [\boldsymbol{I}_6, \boldsymbol{0}, \boldsymbol{0}]\boldsymbol{L}_{\text{lqr}}\delta t/r\big), \tag{3.12}$$

where $\boldsymbol{L}_{\text{lqr}}$ is the LQR gain matrix resulting from the MPC controller, δt is the time elapsed in between two MPC updates (about 4 ms), and $\boldsymbol{I}_6$ is a 6×6 identity matrix. The matrix $\boldsymbol{k}_s$ of the SEA's spring stiffness is given by

$$\boldsymbol{k}_s = \begin{pmatrix} 433 & 0 & 0 & 0 & 0 & 0 \\ 0 & 433 & 0 & 0 & 0 & 0 \\ 0 & 0 & 433 & 0 & 0 & 0 \\ 0 & 0 & 0 & 166 & 0 & 0 \\ 0 & 0 & 0 & 0 & 166 & 0 \\ 0 & 0 & 0 & 0 & 0 & 166 \end{pmatrix}. \tag{3.13}$$

The sensitivity of gear positions with respect to joint angles (i.e., $\frac{\partial \boldsymbol{q}_g}{\partial \boldsymbol{q}}$) can be calculated through the system dynamics

$$\frac{\partial \dot{\boldsymbol{q}}_g}{\partial \boldsymbol{q}} = \frac{\partial}{\partial \boldsymbol{q}} \boldsymbol{v}_m/r = [\boldsymbol{I}_6, \boldsymbol{0}, \boldsymbol{0}]\boldsymbol{L}_{\text{lqr}}/r. \tag{3.14}$$

Assuming that $\boldsymbol{q}_g$ has continuous second partial derivatives[1], we can write

$$\frac{d}{dt}\frac{\partial \boldsymbol{q}_g}{\partial \boldsymbol{q}} = [\boldsymbol{I}_6, \boldsymbol{0}, \boldsymbol{0}]\boldsymbol{L}_{\text{lqr}}/r. \tag{3.15}$$

Since $\boldsymbol{q}$ and $\boldsymbol{q}_g$ are independent at the MPC measurement time t_0, we will have $\frac{\partial \boldsymbol{q}_g(t_0)}{\partial \boldsymbol{q}} = \boldsymbol{0}$. Thus for a small δt, we can write

$$\frac{\partial \boldsymbol{q}_g}{\partial \boldsymbol{q}} = \int_{t_0}^{t_0+\delta t} [\boldsymbol{I}_6, \boldsymbol{0}, \boldsymbol{0}]\boldsymbol{L}_{\text{lqr}}/r dt \approx [\boldsymbol{I}, \boldsymbol{0}, \boldsymbol{0}]\boldsymbol{L}_{\text{lqr}}\delta t/r. \tag{3.16}$$

1 This assumption always holds for any DDP-based methods, if the query time is not a pre-planned switching time between different modes of operations.

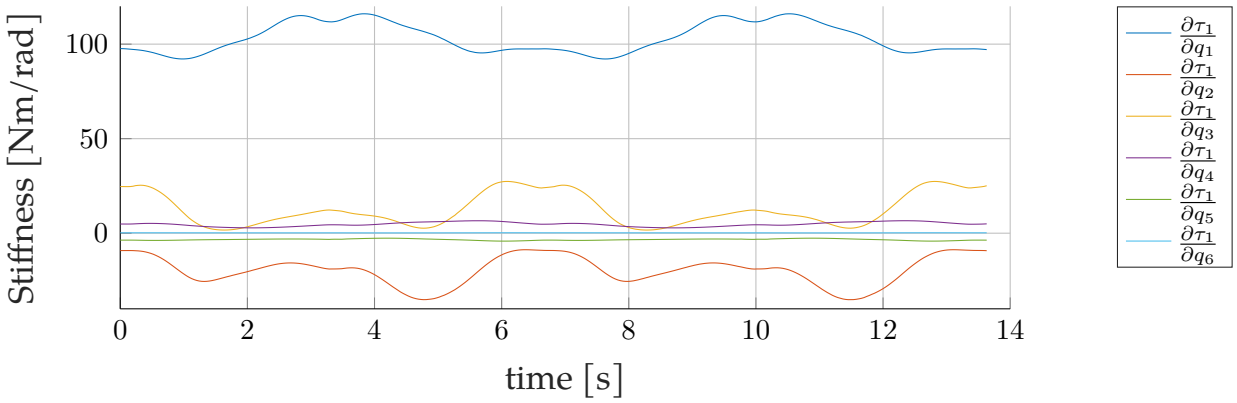


Figure 3.3: Sensitivity of joint q_1's control input τ_1 to joint position perturbations during dynamic trajectory tracking with the MPC $\boldsymbol{\tau}$ controller. The MPC formulation provides this sensitivity in the form of an LQR gain matrix.

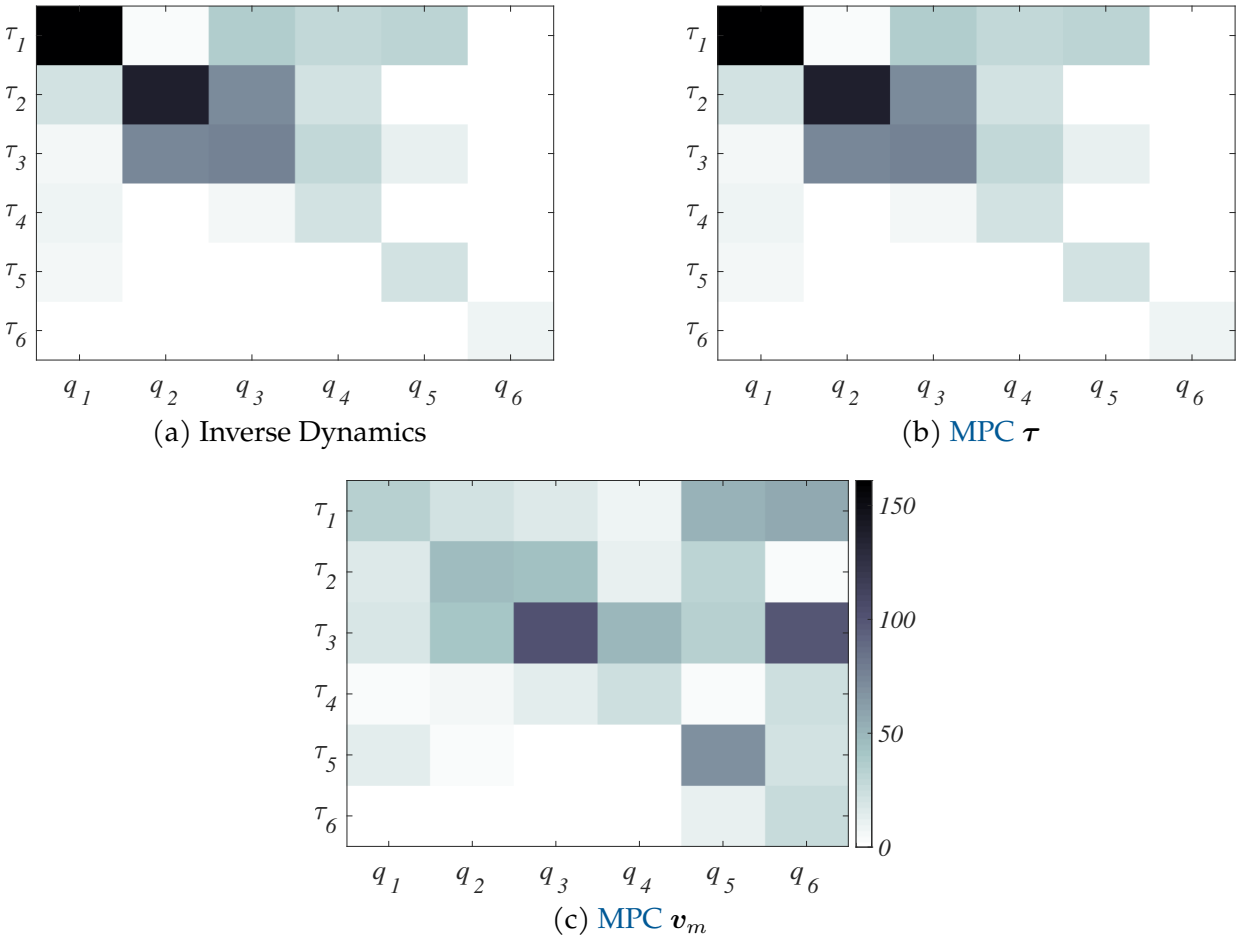
We can see that the natural stiffness of the robot is only altered in proportion to the time step of MPC.

3.5 Experimental Results

This section compares the performance of inverse dynamics control to the two receding horizon MPC strategies presented in Section 3.3 in terms of compliance change and tracking performance. The task is to move a grasped object with a total weight of 1.46 kg dynamically along a defined trajectory in task space, similar to a pick-and-place application[2]. We add the inertia of the grasped object to the end-effector's model to account for the additional weight in the control formulation. The desired motion of the manipulated object provides the reference end-effector pose trajectory $\boldsymbol{p}_{\text{ee}}(t)$. The desired end-effector wrench trajectory is set to zero $\boldsymbol{\lambda}_{\text{ee}}^{\text{des}}(t) = \mathbf{0}$ as the motion is in free space.

A limitation of the ANYpulator arm presented in Section 2.1.2 was the limited control bandwidth of less than 200 Hz due to the CAN. In these experiments, we modified the arm with a newer generation of ANYdrives, equipped with an EtherCAT bus that allows communication speed of up to 1000 Hz. The control loop frequency of 250 Hz was chosen to comply with the computational requirements of MPC $\boldsymbol{v}_m$. The parameters of the gains and weights in the inverse dynamic and MPC controllers are given in Appendix A.1.

2 Watch the accompanying video: https://youtu.be/11Pkz_GGUHw



(a) Inverse Dynamics

(b) MPC $\boldsymbol{\tau}$

(c) MPC $\boldsymbol{v}_m$

Figure 3.4: Change of the joint-space stiffness matrix $\boldsymbol{K}_q$ by the considered control approaches inverse dynamics (a), MPC $\boldsymbol{\tau}$ (b), and MPC $\boldsymbol{v}_m$ (c). The color of each entry represents the magnitude of the output torque change τ_i due to deviation of joint state q_i from its reference in Nm/rad. This stiffness is added to the natural stiffness of the system by closing the high-level control loop and by the joint-level PD feedback.

3.5.1 Stiffness Change

The joint-space stiffness matrix is generally configuration-dependent. Figure 3.3 shows the temporal evolution of the feedback stiffness gains of the first joint q_1 of the arm during the pick-and-place motion. To ease the comparison, we will focus on a representative time point on the reference trajectory in the subsequent evaluation.

Let us first look at the control approaches using torque $\boldsymbol{\tau}$ as control input and the resulting change of the joint stiffness matrix to track the reference input $\boldsymbol{\tau}_{\text{ff}}$ with joint-level feedback gains (see Figure 3.4a and 3.4b). We can see that the added stiffness is up to 162 Nm/rad for the inverse dynamics controller and 112 Nm/rad for

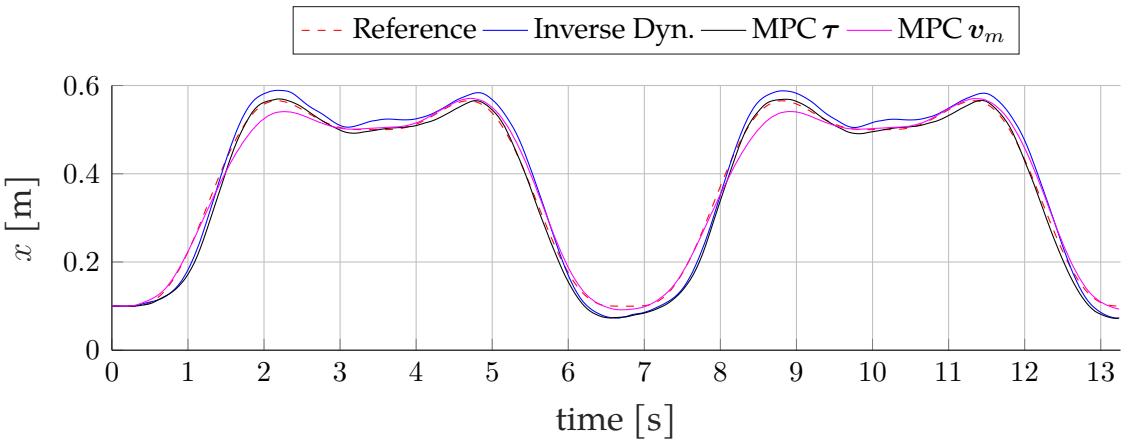


Figure 3.5: Position tracking in x direction for a task-space end-effector reference motion (red dashed) under the assumption of a correct model. Joint-level feedback gains are applied on the wrist joints for the two torque controllers (inverse dynamics and MPC $\boldsymbol{\tau}$). The three control approaches, inverse dynamics (blue), MPC $\boldsymbol{\tau}$ with torque input (black), and MPC $\boldsymbol{v}_m$ with motor velocity input (magenta) show a similar performance.

the MPC $\boldsymbol{\tau}$ controller. Generally, both controllers show a comparable increase in stiffness, which corresponds to the observation of similar tracking performance in Section 3.5.2. As expected, the high-level control law is introducing non-diagonal elements to the joint stiffness matrix. The most noticeable difference in the stiffness change between the two torque control approaches is seen for the diagonal elements of the wrist joints, although both controllers use the same PD feedback gains. MPC $\boldsymbol{\tau}$ has a higher increase in wrist stiffness, which helps to improve the rotational tracking performance (see Figure 3.7).

In the case of MPC $\boldsymbol{v}_m$ (see Figure 3.4c), we observe less change of the joint stiffness matrix compared to inverse dynamics and MPC $\boldsymbol{\tau}$. Generally, the stiffness change is more coupled and less pronounced on the diagonal, which means that MPC $\boldsymbol{v}_m$ alters to a smaller extent the natural compliance of the elastic element of the actuator. This stiffness is added to the natural stiffness $\boldsymbol{k}_s$ of the SEA's spring (Eq. 3.12) for the MPC $\boldsymbol{v}_m$ controller. In contrast, the stiffness in torque control is added to the stiffness of the force controller, which is expected to be stiffer than the elastic element. For MPC $\boldsymbol{v}_m$, the change of compliance is proportional to the time step δt between two MPC updates, meaning that the stiffness change reduces by decreasing the time step.

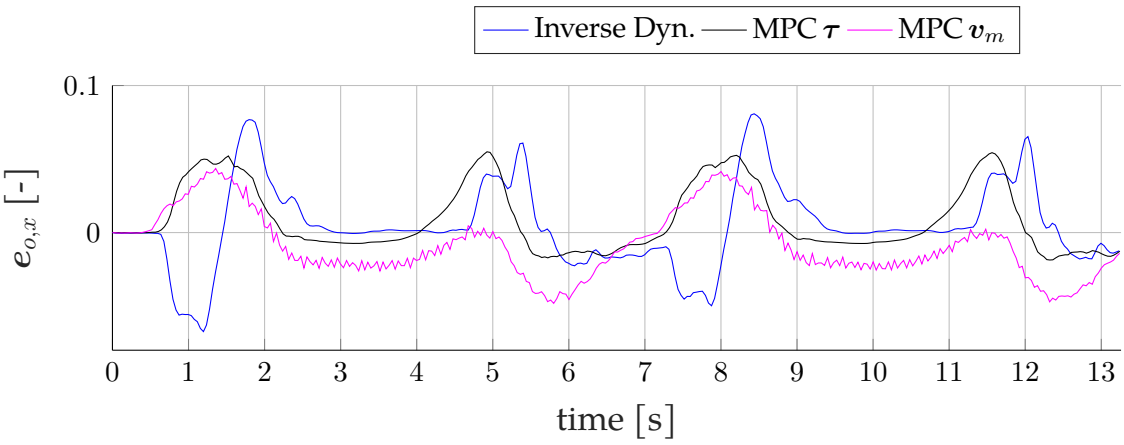


Figure 3.6: Rotation error for tracking end-effector motion reference under the assumption of a correct model. Joint-level feedback gains are applied on the wrist joints for the two torque controllers (inverse dynamics and MPC τ). The three control approaches, inverse dynamics (blue), MPC τ with torque input (black), and MPC v_m with motor velocity input (magenta) show a similar performance.

3.5.2 Tracking Performance

First, we compare the tracking performance of the three investigated controllers in case of accurate knowledge of the model. In Figure 3.5, we see that the reference end-effector position trajectory $r_{\text{ee}}^{\text{des}}(t)$ in the x-direction (dashed red) is tracked with a similar performance, whether inverse dynamics control (blue), receding horizon control with torque input (black), or SEA motor velocity input (magenta) is applied. The position tracking in y- and z-direction shows comparable performance, and we omit it here for brevity. The orientation tracking shows similar performance for the tree compared control methods, as we can see from the rotation error e_o in the x-direction in Figure 3.6. However, note that inverse dynamics and MPC τ control rely on the joint-level PD feedback gains for tracking. Not only would the tracking performance suffer, but the arm motion would diverge or become unstable without them. In contrast, our proposed method MPC v_m does not depend on joint-level feedback gains for accurate tracking. Again, the rotation tracking performance shows comparable performance in all directions, and the other plots are omitted. For quantitative comparison of the tracking error, we compute the Integral Square Error (ISE) over the time horizon T by

$$ISE = \frac{1}{T}\int_{t=0}^{T} \epsilon^2(t)\, dt, \tag{3.17}$$

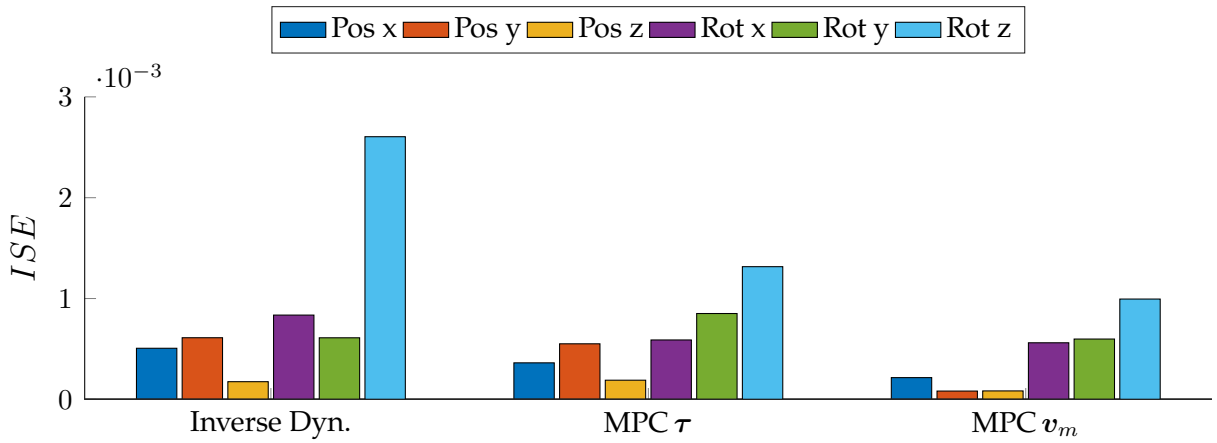


Figure 3.7: ISE for tracking the end-effector pose. Joint-level feedback gains are applied on the wrist joints for the two torque controllers (inverse dynamics and MPC $\boldsymbol{\tau}$). Using MPC $\boldsymbol{v}_m$ shows a slightly improved tracking performance, although there are no joint-level feedback gains necessary.

where $\epsilon\,(t)$ is the difference between the reference and actual value at time t. Figure 3.7 illustrates the ISE for position and rotation. The MPC approach with the torque input performs very similar to inverse dynamics control. Using the MPC approach with SEA motor velocity input, a slightly improved performance is achieved compared to the other two approaches.

As shown in Figure 3.7, we can achieve similar performance with the three different control approaches as long as the controller has a very accurate robot model. However, MPC $\boldsymbol{v}_m$ achieves this by sacrificing the least natural compliance of the system.

3.5.3 Model Mismatch

In this experiment, we attached an additional unmodeled mass of 0.92 kg to the grasped object, which corresponds to an increase of the object mass of 63 %. Figure 3.8 shows the end-effector position tracking for the same motion as in Section 3.5.2, but with the additional mass as a disturbance. Both, inverse dynamics control and MPC $\boldsymbol{v}_m$, can still track the motion in x-direction accurately (Figure 3.8 top). At the same time, the tracking performance degrades noticeably for inverse dynamics in the z-direction (Figure 3.8 bottom). The inverse dynamics controller tracking performance degrades noticeably and differs from the tracking of the MPC with

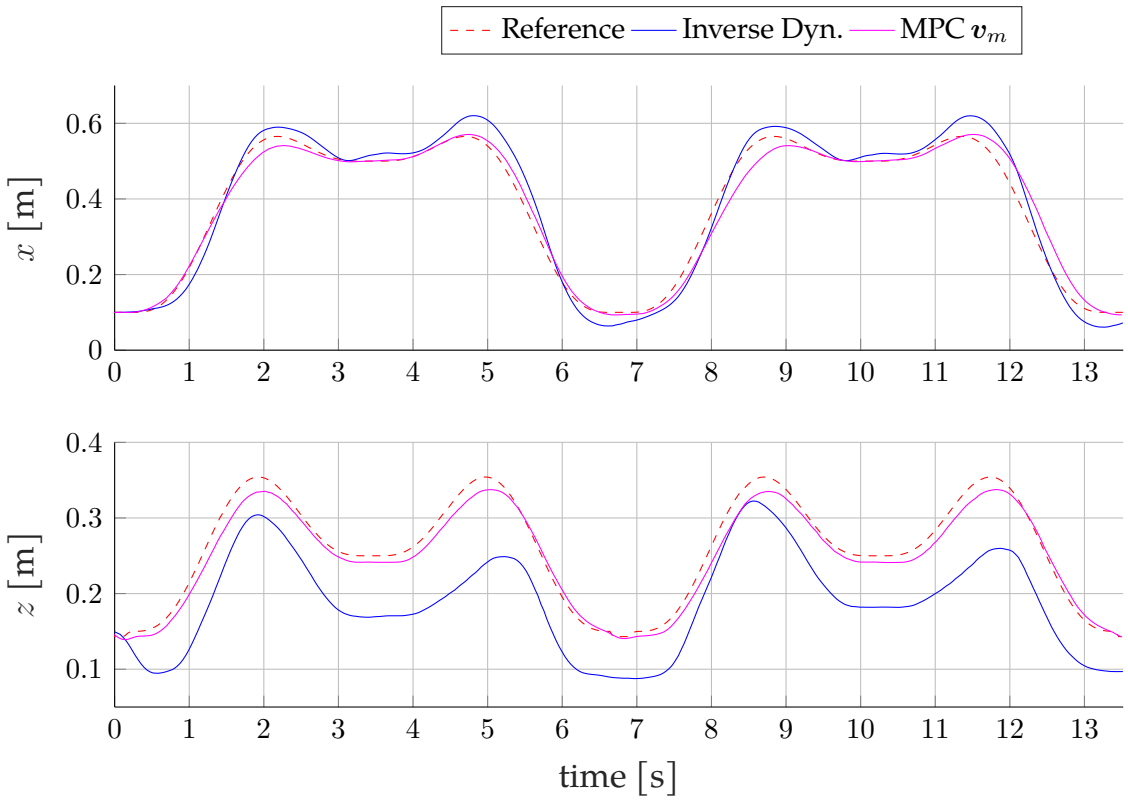


Figure 3.8: Position tracking in x and z direction for a task-space end-effector reference motion (red dashed) with an unmodeled end-effector mass of 0.92 kg. The inverse dynamics controller shows a clear offset in z-direction, whereas the MPC $\boldsymbol{v}_m$ is able to compensate the model mismatch.

SEA motor velocity input as it is not able to follow the desired trajectory in the z-direction.

The difference in the tracking performance becomes visible in Figure 3.9, where the pose tracking error is illustrated. The inverse dynamics controller shows an apparent increase in the tracking error compared to the undisturbed case (see Figure 3.7), especially in the z-direction and rotations. In contrast, our MPC approach, including the actuator dynamics, shows comparable performance, and one can see almost no influence of the disturbance mass. A comprehensive discussion on how the robustness improves by respecting the actuator dynamics is presented by [65]. Instead of directly adding the actuator dynamics to the control formulation, they modify the cost function to penalize higher frequency input. This cost function improves the high-frequency robustness and indirectly addresses the issue of the limited actuator bandwidth. In our case, we directly encode this robustness by enforcing the actuator dynamics in the control formulation.

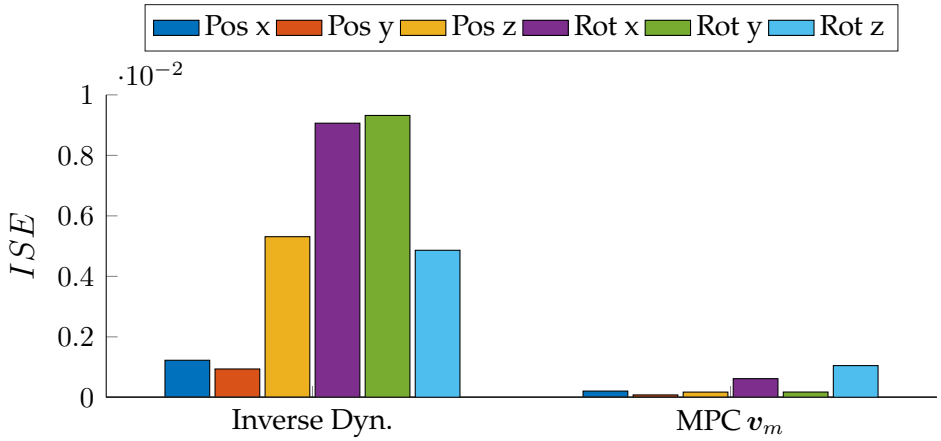


Figure 3.9: ISE for tracking the end-effector pose with an unmodeled end-effector mass of 0.92 kg. The inverse dynamics controller performance significantly worse compared to the case without the model mismatch, especially in z-direction and for the rotation. Our MPC including the actuator dynamics shows a comparable performance with almost no influence of the disturbance mass.

Notice that the most significant impact of the additional unmodeled mass is seen by the shoulder and elbow flexion/extension joint that carries the main gravitational load of the arm. Those joints have no feedback gains compensating the tracking offset. Adding integral gains on joint-level would improve the tracking performance in case of a model mismatch but increase the system's stiffness, especially at low frequencies [66].

Furthermore, note that MPC $\boldsymbol{\tau}$ is not displayed for comparison as it could not perform the dynamic motion with the unmodeled end-effector mass. The optimization-based approach tries to maximize the motion over the system's full potential, which might violate the boundaries of the real system if unmodeled disturbances appear. In contrast to MPC $\boldsymbol{v}_m$, MPC $\boldsymbol{\tau}$ does not add high-frequency robustness and fails to provide a solution for highly dynamic disturbances.

3.5.4 Large Reference Steps

This experiment shows the response behavior of the different control formulations to a sparser input trajectory, resulting in more significant reference steps. The goal is again to move a grasped object with a total weight of 1.46 kg. We apply discrete steps of 10 cm in the reference trajectory instead of a continuous curve. Figure 3.10 shows the tracking performance of the end-effector position. We expect that MPC makes a trade-off between the current and future tracking errors over the time hori-

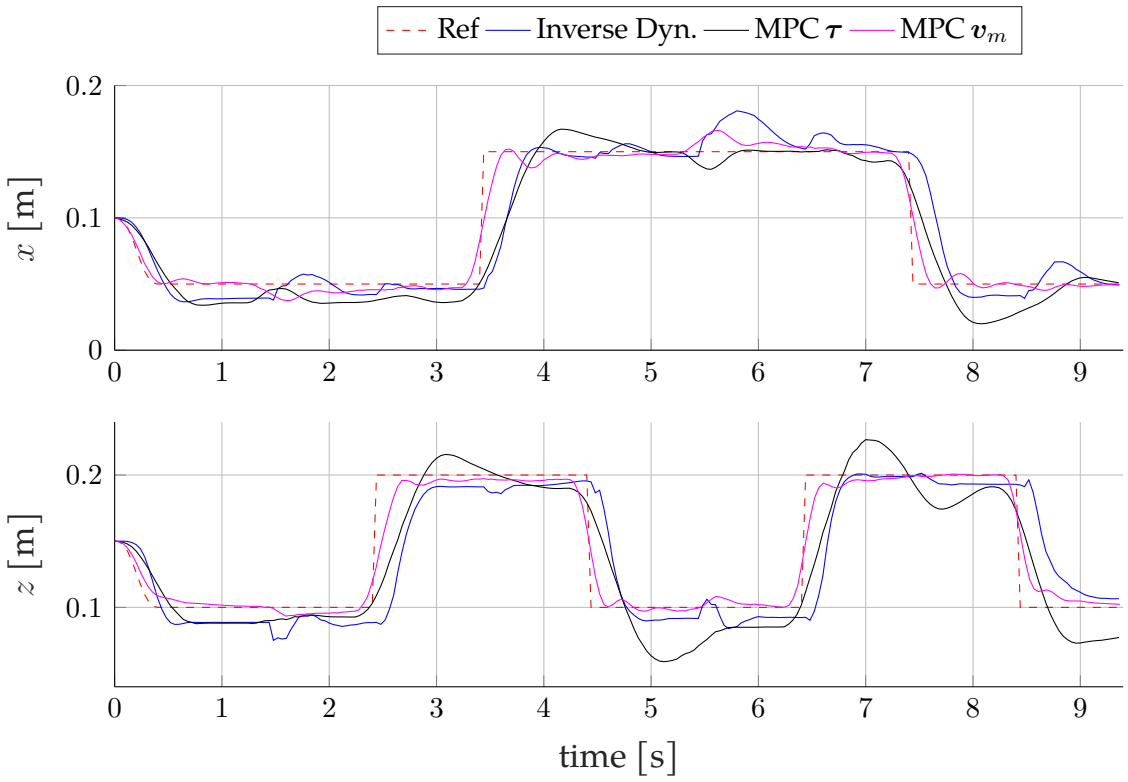


Figure 3.10: Position tracking in x and z direction for reference step motions of 10 cm (red dashed). The reference step is anticipated by the receding horizon controllers (MPC τ and MPC v_m) and the actual rising motion starts before the step takes place. MPC v_m achieves a better step response performance compared to torque control input with a faster rise time and less overshoot.

zon, creating a smoother motion. Indeed, we can see that in the case of MPC, the reference step is anticipated through the receding horizon and the actual rising motion starts before the step takes place. In contrast, inverse dynamics control only reacts after the reference change happened. The MPC formulation with SEA motor velocity control input achieves a better step response performance compared to torque control input with a faster rise time and less overshoot.

The pose tracking performance for the sparse input reference, in the form of ISE, is shown in Figure 3.11. The position error is comparable to tracking a continuous trajectory (see Figure 3.7 for the three control approaches), whereas the rotation error increases significantly for inverse dynamics control. The MPC approaches are computing optimal feedforward plans in correspondence with sensed changes of the state, making them less affected by step changes and letting them achieve more consistent performance for reference steps in any direction. The influence of the receding horizon can be presented best by having a closer look at the commanded

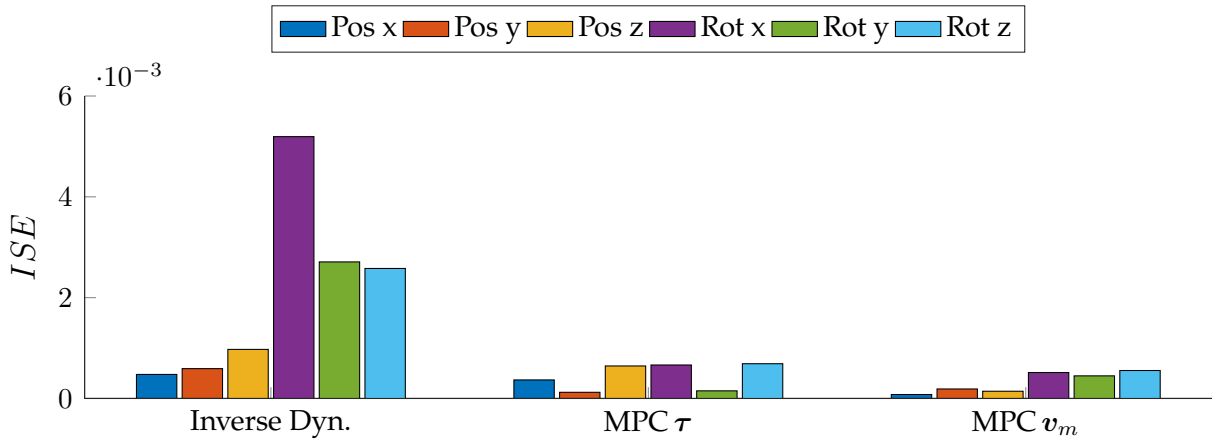


Figure 3.11: ISE for tracking reference steps of 10 cm. The tracking performance is less influenced by reference step changes in the case of applying receding horizon control and a more uniform performance is achieved for reference steps in any direction.

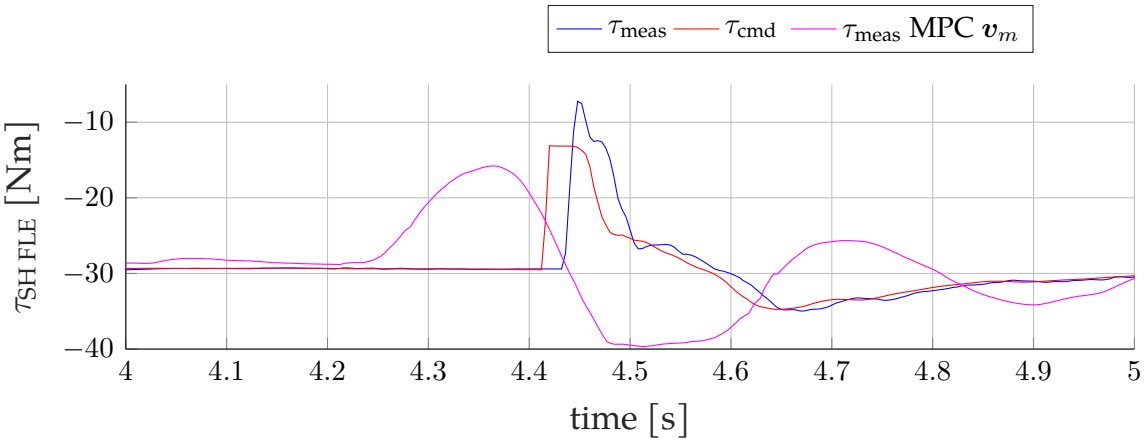


Figure 3.12: Overview of the commanded and measured torque at the shoulder flexion/extension joint during a sudden reference step. The commanded torque (red) of the inverse dynamics controller has a sudden jump due to the reference step, that cannot be tracked by the actuator (blue). In comparison, in the case of using the MPC formulation with SEA motor velocity input the measured torque (magenta) follows a smooth curve.

and applied torque at the shoulder flexion/extension joint during reference steps in Figure 3.12. We observe that a sudden command torque jump is generated for the inverse dynamics controller due to the reference step (red). However, the actuator cannot follow the torque reference immediately and overshoots significantly (blue). In the case of MPC $\boldsymbol{v}_m$ the measured torque (magenta) results in a smooth curve. Noticeably, the applied torque already changes before the reference step happens, caused by the receding horizon.

3.6 Summary

This chapter discussed the inclusion of the compliant actuator model in the receding horizon control formulation of a manipulator. The chosen MPC formulation provides a generic tool to incorporate the actuator dynamics in the task-space control that is not possible with classical inverse dynamics controller. We compare different control approaches and show how they alter the natural stiffness of the system. The proposed actuator-aware MPC formulation has the least impact on the natural stiffness and frees us from joint-level feedback gains that sacrifice compliance for tracking performance and stability. Our proposed controller is shown to be robust against deviation of end-effector inertia and can reject significant disturbances. By including the actuator dynamics, we improve the high-frequency robustness and directly address the issue of the limited actuator bandwidth as the real physical spring acts as a proportional gain to drive the link to its desired position. Furthermore, the actuator-aware MPC generates a physically more consistent motion, as it avoids demanding abrupt torque changes. All the contributions that we present are empirically validated on the ANYpulator platform.

Part II

OBJECT HANDLING

4

Large-Scale Object Mapping, Segmentation, and Manipulation

This chapter incorporates material from the following publication:

Mascaro*, R., Wermelinger*, M., Hutter, M., & Chli, M. Towards automating construction tasks: Large-scale object mapping, segmentation, and manipulation. *Journal of Field Robotics* **38** 5, pp.684-699 (2021).

Video: https://youtu.be/4bc5n2-zj3Q

Automating the robotic assembling of on-site material poses several challenges that are specific to construction sites. First, the use of arbitrarily shaped materials found on-site requires efficient perception algorithms to identify individual object instances on the fly without having any previous knowledge about their geometry. A system to manipulate these irregular objects reliably must be capable of selecting good grasping poses among many possible configurations and have the ability to plan collision-free motions in potentially cluttered environments. Another major challenge in real-world construction applications is manipulating heavy objects and exerting large forces, requiring a powerful and versatile robotic system. Moreover, the fabrication of large-scale building structures typically requires mobile robots to overcome the constrained workspace limitations of stationary manipulators. These robots need to move during construction while still locating themselves with respect to the changing and cluttered working environment and assemble structures accurately in space.

This chapter addresses these challenges by demonstrating autonomous manipulation of large-scale stones with a robotic walking excavator. By extending state-of-the-art mapping and grasp planning approaches to real environments and irregu-

* Ruben Mascaro and Martin Wermelinger contributed equally to this work. R.M. was responsible for the perception pipeline and M.W. for the manipulation tasks

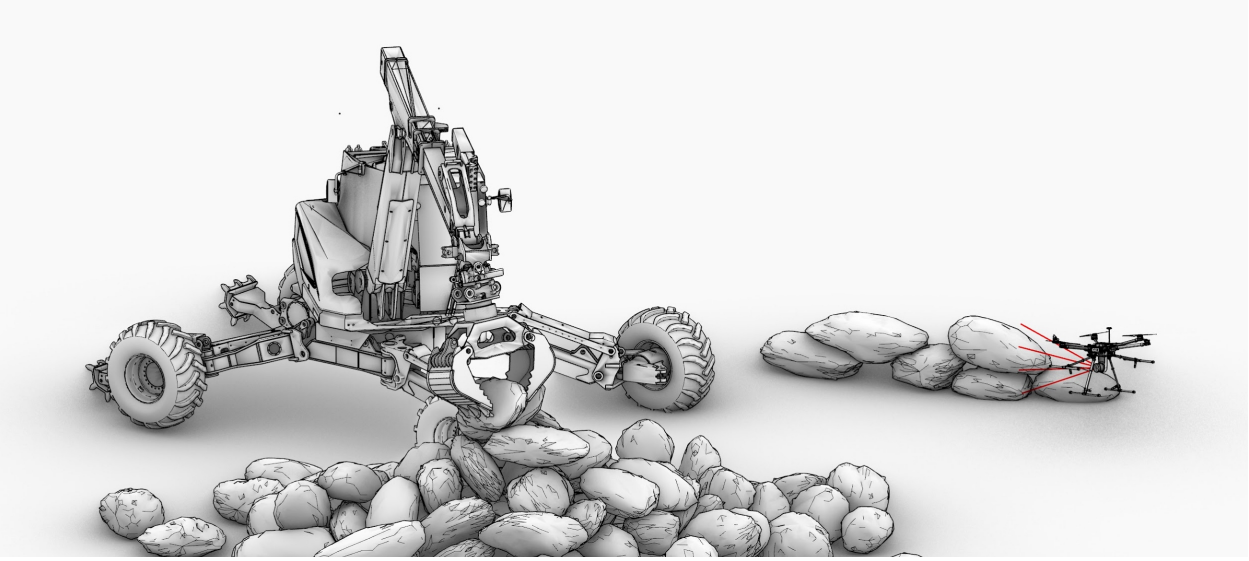


Figure 4.1: Potential application of the proposed system: robotic construction of utility structures with material found on-site. By fusing mapping data acquired from the excavator and a drone, the observed stones are segmented in the acquired map, while collision-free paths for the manipulator are computed and executed to move the stones of interest to the desired locations in the vicinity of the robot.

lar objects, we provide the robotic skills that are necessary to deploy autonomous construction machines for further construction applications (see Figure 4.1). The proposed system initially maps the close vicinity of the robot by fusing the data acquired by its onboard LiDAR sensors and a camera placed on a drone and segments single object-like instances in the resulting 3D point cloud. An object inventory allows keeping track of the segments and their corresponding poses during manipulation. Finally, based on the map and the segment inventory, collision-free grasping poses are planned to move the objects of interest to the desired locations. All experiments are executed in the real-world setting and with the autonomous excavator HEAP shown in Figure 4.2, emphasizing on the evaluation of the robustness of the integrated perception and manipulation pipelines.

4.1 Related Work

In order to autonomously build structures with material found on-site, construction robots require the ability to perceive and map the complex, unstructured surrounding space, segment individual objects of interest in it, and manipulate these objects safely. Despite the recent advances in the context of on-site digital fabrication,

most applications only consider the use of regular materials, such as pre-fabricated bricks [3], and there are still only a few robots integrating all the capabilities mentioned above. For example, in [67] and [9], completely autonomous systems are shown to construct auxiliary structures in order to achieve and maintain navigability across previously untraversable terrain. However, they use customized compliant bags as construction material or apply polyurethane foam, respectively, and not naturally occurring building materials such as stones. Closely related to our work are the demonstrators presented in Chapter 6 and [68], consisting of stationary robotic systems that detect randomly placed stones and construct balancing vertical stacks with them. Although they emphasize handling building materials of arbitrary shape, the systems above assume that the geometry of the objects of interest is known beforehand, i.e., they are initially pre-scanned in an offline step. On the other hand, our approach aims towards more generic use-cases and does not consider any previous knowledge about the objects present in the scene. Instead, it attempts to discover object-like instances in a map of the robot's surroundings that is built online and perform manipulation based on this information.

Estimating the robot's pose together with an internal representation of the observed scene, which is commonly referred to as SLAM, has been an extensively studied problem by the robotics community in the last decades [69]. Particularly, LiDAR-based SLAM approaches [70–72] have become quite popular due to their applicability to self-driving vehicles and other types of ground robots. However, these systems typically target autonomous navigation use-cases and mainly seek to achieve reasonable pose estimates over very long trajectories, producing purely geometric maps of the traversed environment. Conversely, to aid the manipulation tasks, we aim towards a higher-level scene representation where the notion of object instances is available. While object-aware mapping systems have gained attention recently [73–75], most of the existing approaches use volumetric map representations and rely on RGB-D sensing. However, the applicability of depth cameras is usually inhibited in large-scale outdoor environments due to the limited working range and the poor performance under sunlight. Our mapping approach, on the contrary, is more similar to [76], which uses a map representation based on geometric segments extracted from 3D LiDAR point clouds. The main difference is that, while in [76] these segments often correspond to partial observations of objects or structures and are mainly used for localization and loop-closure detection [77],

here we aim at obtaining complete models of the individual objects in the scene in order to assist interaction planning.

Achieving an accurate and complete reconstruction of the scene becomes especially challenging when operating on construction sites. The presence of bulky objects (e.g., building material) can limit the robot's mobility and occlude large regions of the map. To overcome this drawback, we take inspiration from works on aerial-ground registration [78, 79] and enable the augmentation of the LiDAR-based map with additional mapping data acquired by an external sensor that can observe the scene from different viewpoints, such as a camera placed on a drone. A general approach to solve the 3D global registration problem is to extract features from the input maps, match them and estimate the geometric transformation that best explains the set of found correspondences [80]. Like [77], our method aims to directly align sets of point-cloud segments extracted from the input maps, as they are typically the most salient elements in the observed scene. However, instead of treating these segments as single features, which might struggle when dealing with significant differences in viewpoint [79], we compute local descriptors on these segments and then use the segments' centroids to select a geometrically consistent set of descriptor correspondences efficiently.

We use a similar approach for grasp planning as presented in [81] to sample grasp hypotheses on the point cloud directly and to classify the grasp candidates before scoring them according to a heuristic cost function. A more detailed literature review about grasp planning methods is given in Section 5.1.

4.2 System Description

As a robot platform, we use HEAP (Hydraulic Excavator for an Autonomous Purpose), a highly customized Menzi Muck M545 excavator developed for autonomous use cases and advanced teleoperation. This machine (shown in Figure 4.2) has customized, precisely force and position controllable hydraulic actuators in the arm and the legs, making it adaptable to any kind of terrain. On the sensing side, HEAP is equipped with a Leica iCON iXE3 with two Global Navigation Satellite System (GNSS) antennas and a receiver that can be used for localization of the cabin. Real-time kinematic (RTK) corrections for the GNSS signals are received over the Internet from permanently installed base stations. In addition, SBG Ellipse2-A Inertial Measurement Units (IMUs) are installed both in the cabin and on the chassis, and

Figure 4.2: HEAP is an autonomous walking excavator based on a highly customized Menzi Muck M545. It is equipped with onboard sensors for state estimation (GNSS, IMUs, and encoders) and scanning the environment (3D LiDARs). As shown in the inset, the two LiDARs are mounted perpendicularly to each other such that a scanning motion is achieved not only while driving the robot around, but also while swinging the cabin.

Sick BCG05-C1QM0199 wire draw encoders measure the piston position and velocity of the arm cylinders, whose force is estimated with pressure sensors integrated into the servo valve control modules. Finally, two Velodyne Puck VLP-16 LiDAR scanners placed at the front edge of the cabin's roof are used for mapping tasks. As shown in Figure 4.2, the LiDARs are mounted orthogonally to each other. This way, a scanning motion is achieved while driving the robot around and swinging the cabin. In the scope of this work, the sensor mounted perpendicularly to the ground plane will be referred to as the *vertical LiDAR*, whereas the other one will be named *horizontal LiDAR*. For a more detailed description of the system's sensors and actuators, we refer the reader to [6].

Besides HEAP, an Ascending Technologies (AscTec) Neo hexacopter equipped with a Visual-Inertial (VI)-Sensor [82] is used in the experiments presented in this

work to provide additional mapping data of the environment. This data is registered into the excavator's LiDAR-based map leading to a reconstruction of the scene free of occlusions, as explained in Section 4.3.4.

4.3 Perception Pipeline

As the goal here is to manipulate objects of interest present in the scene autonomously, we are looking to achieve a consistent and up-to-date map of the robot's (i.e., excavator's) surroundings as well as accurate segmentation masks for the objects in the map. In this work, we assume no prior knowledge of the environment or the objects to be grasped for the sake of a generally applicable methodology. Therefore, we present a mapping pipeline that uses the excavator's onboard LiDAR sensors to build a 3D point-cloud map of the environment incrementally and can segment generic objects in it. Additionally, the system features a segment-based global registration module that enables fusing external mapping data, such as point-cloud maps generated from drone-borne vision sensors, into the LiDAR-based map to achieve a more comprehensive reconstruction of the scene. The object-like instances segmented out from this map are stored in an inventory that is consulted to update the map when the robot moves objects in the scene. An overview of the different modules constituting the perception pipeline is depicted in Figure 4.3.

4.3.1 Vision-based Pre-mapping Using a Drone

Intending to provide a complete map of the scene for grasp pose planning (i.e., free of occlusions), we intend to reconstruct a point cloud of the region of interest from multiple viewpoints and register it later on with the excavator's LiDAR-based map. To this end, we initially collect some visual data with a VI-Sensor mounted on the AscTec Neo hexacopter, which is piloted manually over the region of interest. The VI-Sensor's left camera trajectory is estimated online using the VINS-Mono SLAM system [83]. Once this inspection task is done, we select a subset of about 200 images from the recorded data, and we perform an offline reconstruction of the scene based on Structure from Motion (SfM) using COLMAP [84]. The scale of the reconstructed model is finally recovered by computing the 3D similarity transformation between the camera locations provided by the SfM pipeline and the corresponding

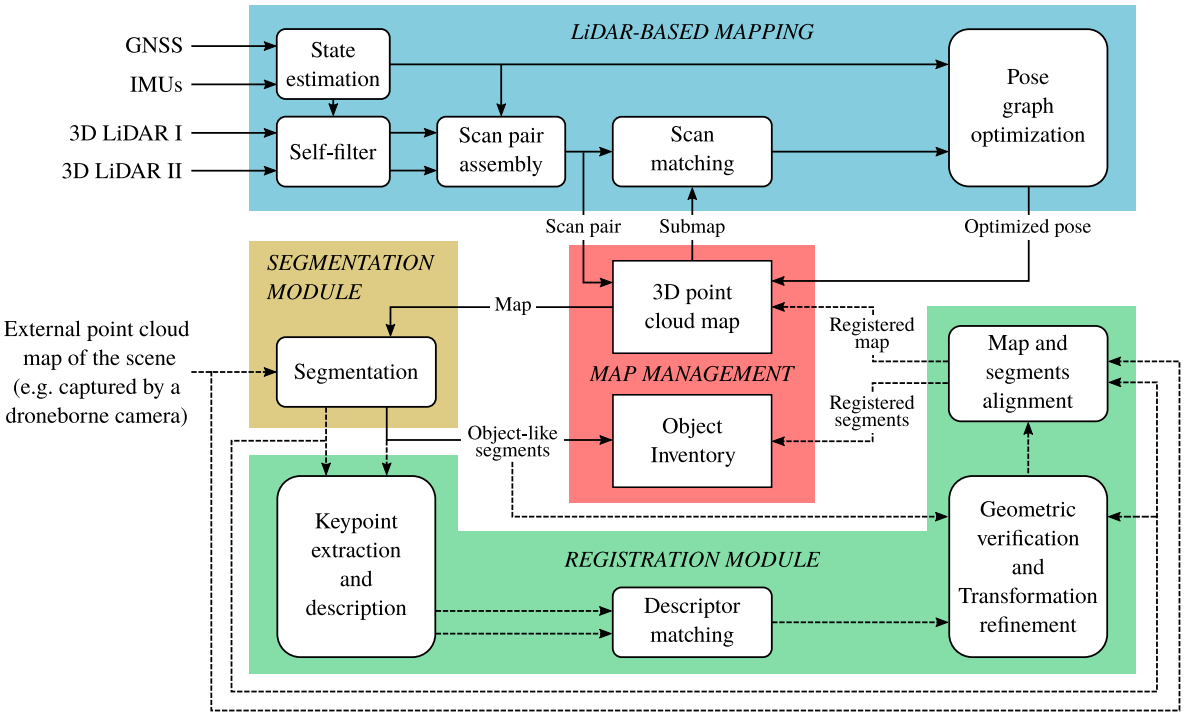


Figure 4.3: Overview of the perception pipeline developed for the excavator HEAP. The LiDAR, inertial and GNSS measurements are processed online to estimate the sensor-suite's pose and create a point-cloud map of the environment. When triggered, the segmentation routine extracts object-like segments from the current map and stores them as separate entities in the Object Inventory. Additionally, the registration module allows the fusion of externally built maps for improved scene reconstruction.

camera positions previously estimated with VINS-Mono. Although efficient vision-based online mapping algorithms exist, we decide to use a method that produces high-quality point clouds (see Figure 4.8 for an example), allowing for a better alignment with the LiDAR-based map. Note, however, that this method is used as a proof of concept and that other sensing modalities and mapping approaches able to reconstruct point-cloud maps of an observed scene (e.g., LiDAR-based) could also be employed here.

4.3.2 LiDAR-based Scene Mapping

The excavator's LiDAR-based mapping system exploits the well-established graph-based SLAM formulation [85],which typically models the problem's underlying

structure as a pose graph. In graph-based SLAM, every node in the graph corresponds to a robot pose and edges connecting these nodes encode spatial constraints between robot poses (i.e., relative transformations) that result from observations or odometry measurements. Solving the SLAM problem then determines the set of robot poses that best satisfies all the spatial constraints.

In mathematical terms, we denote a set of state variables as $\mathcal{P} = \{\boldsymbol{p}_1, ..., \boldsymbol{p}_m\}$, where $\boldsymbol{p}_i$ describes the pose of node i, and a set of measurements, as $\boldsymbol{z}_{ij}$ and $\boldsymbol{\Omega}_{ij}$ being respectively the mean and the information matrix of a single measurement relating nodes i and j. With a maximum likelihood approach, we find the configuration of the nodes $\mathcal{P}^*$ that minimizes the negative log-likelihood of all the observations. This formulation is equivalent to solving the following least squares error minimization problem:

$$\mathcal{P}^* = \underset{\mathcal{P}}{\operatorname{argmin}} \sum_{\langle i,j \rangle \in \mathcal{C}} \boldsymbol{e}_{ij}^T \boldsymbol{\Omega}_{ij} \boldsymbol{e}_{ij}, \tag{4.1}$$

where $\mathcal{C}$ denotes the set of pairs of indices for which a measurement is available, and $\boldsymbol{e}_{ij} = \boldsymbol{z}_{ij} - \hat{\boldsymbol{z}}_{ij}\left(\boldsymbol{p}_i, \boldsymbol{p}_j\right)$ is a vector error function that measures how well the constraint originated from measurement $\boldsymbol{z}_{ij}$ is satisfied by the predicted measurement $\hat{\boldsymbol{z}}_{ij}\left(\boldsymbol{p}_i, \boldsymbol{p}_j\right)$, given a configuration of the nodes $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$.

The nonlinear optimization problem in Eq. 4.1 can be effectively solved using the Gauss-Newton approach. Specifically, the proposed mapping system builds on top of the LaserSLAM [76] back-end, which performs incremental pose-graph update and optimization using the iSAM2 [86] algorithm to estimate the robot's trajectory in real-time. A customized front-end processes measurements provided by the onboard sensors to generate odometry and scan-matching constraints between consecutive nodes of the pose graph. In this work, odometry constraints are generated from state estimation, which fuses GNSS measurements with the inertial data provided by the robot's IMUs. Scan-matching constraints, on the other hand, are computed by aligning consecutive LiDAR scans using a process that consists of the three following steps:

1) **Self-filtering:** To prevent the scene map from being corrupted by the robot's arm and legs, which move primarily inside the field of view of the LiDAR sensors, each incoming scan is initially processed by a self-filter that approximates

the excavator's links with simple shapes, such as boxes or cylinders, and uses the robot's current state to discard the points lying on the robot itself [6].

2) **Scan Pair Assembly:** Once a new pair of filtered scans is available, the vertical scan $\boldsymbol{S}_k^V$ is merged with the corresponding horizontal scan $\boldsymbol{S}_k^H$ to form an "assembled scan pair" $\boldsymbol{S}_k$, i.e., a point cloud composed of both $\boldsymbol{S}_k^V$ and $\boldsymbol{S}_k^H$, expressed in the frame of the horizontal LiDAR. We transform $\boldsymbol{S}_k^V$ into the frame of $\boldsymbol{S}_k^H$, which is achieved by applying a fixed pre-calibrated transformation $\boldsymbol{T}_{\mathcal{HV}}$ from the vertical LiDAR frame $\mathcal{V}$ to the horizontal LiDAR frame $\mathcal{H}$. The vertical and horizontal scans are not synchronized. Therefore, we use their timestamps and the state estimation measurements to correct potential cabin motion between them.

3) **Scan Matching:** Constraints relating consecutive robot poses are obtained using the ICP algorithm to register the current scan pair $\boldsymbol{S}_k$ against a *submap* composed of the m previous scan pairs, expressed in the frame of the previous horizontal LiDAR's pose. Compared to performing scan matching for each scan individually, our approach has two main advantages: first, given that scan pairs contain more points than single scans, the ICP registration against the *submap* becomes more robust; secondly, since nodes are added on a scan pair basis, the pose graph keeps the same size as if a single LiDAR sensor was used.

The LiDAR-based map is created by accumulating the 3D scan pairs once the back-end optimizes the corresponding nodes. We want to avoid an uninformative accumulation of data and the growth of the pose graph when the robot is not moving. Therefore, a new node together with one odometry and one scan-matching constraint is added to the graph only if the horizontal LiDAR has traveled a minimum distance d_{poses}. Furthermore, since we are primarily interested in mapping the excavator's workspace, a local point cloud is extracted upon adding new scans by defining a robot-centric cylindrical region, whose radius r_{map} is set to be slightly higher than the maximum arm's reach. The map data is finally filtered and downsampled using a voxel grid of resolution r_{voxel} with n_{min}, a minimum number of points per voxel to consider it as occupied. The result is a 3D point cloud map that is incrementally built around the current robot location without being corrupted by the platform's moving parts.

4.3.3 Map Segmentation

When the point density of the LiDAR-based map reaches a certain threshold during an initial scanning phase, a segmentation routine is automatically triggered to detect any distinct objects of interest present in the scene. We extract such objects from the current map as a set of 3D point clusters $\boldsymbol{P}_i$ (i.e., segments) using a geometric technique inspired by [87]. This method assumes that the ground operates as a separator between the segments and requires it to be previously removed from the input point cloud. In our implementation, this is achieved by running a RANSAC-based 3D plane fitting algorithm with a distance margin d_{plane} that accounts for potential terrain undulations. After most of the points belonging to the ground have been removed, Euclidean clustering is used to grow segments. The extracted clusters then go through a RANSAC-based planarity check that discards nearly planar segments, which usually correspond to regions of the terrain not filtered out previously. Finally, the remaining non-planar segments $\boldsymbol{P}_i$ and their centroids $\boldsymbol{r}_{c,i}$ (i.e., the average of all $\boldsymbol{P}_i$'s points) are added to the Object Inventory described in Section 4.3.5.

4.3.4 Segment-based Global Registration

The registration module gives the perception system the ability to align externally built point cloud maps with the LiDAR-based map that the robot uses for localization. This registration becomes especially valuable when the goal is to perform manipulation tasks. Augmenting the robot's map with additional data acquired by an external sensor (e.g., a camera placed on a drone) helps to add information in occluded regions of this map and allows for effective 3D reconstruction of the objects in it.

Our registration approach finds the transformation that best aligns the externally built (i.e., source) map with the LiDAR-based (i.e., target) map. Therefore, it leverages the segmentation module presented in Section 4.3.3 to perform registration based on local geometric descriptors computed on segments. The resulting transformation is finally used to align the two original input maps and merge the overlapping segments from the source and target point clouds. By choosing a purely geometric method instead of relying on GPS-based co-localization strategies, our

method can deal with mapping data acquired by robots in GPS-denied environments. The proposed approach consists of the following steps:

1) **Segmentation of the Input Maps:** When the registration is triggered, the source map is initially filtered using a voxel grid of resolution r_{voxel} (i.e., the same that is used to downsample the LiDAR-based map), resulting in a point cloud that has a similar point density to the target map. Both the source and the target point clouds are segmented using the method described in Section 4.3.3. This step is beneficial to remove parts of the input maps whose geometry is not descriptive enough to allow for robust matches, e.g., planar or low point-density regions.

2) **Keypoint Extraction and Description:** From both segmented point clouds, we extract keypoints using the Intrinsic Shape Signatures (ISS) detector [88] and describe them using the Signature of Histograms of Orientations (SHOT) descriptor [89]. Besides SHOT, we tested two additional descriptors, Fast Point Feature Histogram (FPFH) [90] and Rotational Projection Statistics (RoPS) [91], but they exhibited lower performance. The former was less robust in the matching step, whereas the latter achieved comparable results to the SHOT descriptors but were considerably more expensive to compute because the triangulation of both input point clouds was required in this case.

3) **Descriptor Matching:** The matching module solves the data association problem between keypoints extracted from both input maps by comparing their descriptors. In our implementation, we perform an efficient nearest-neighbor search in the descriptor space using a kd-tree.

4) **Geometric Verification:** From the set of 3D keypoint correspondences identified in the previous step, we extract clusters of geometrically consistent matches (i.e., matches that vote for the same geometric transformation) and select the b most voted for transformations. These transformations are then used to transform the source segment centroids from the source map frame $\mathcal{S}$ to the target map frame $\mathcal{M}$. For each set of transformed source centroids, we perform a nearest-neighbor search against the set of target centroids. Matches between closest centroids are accepted if the Euclidean distance between them lies below a threshold distance d_{match}. Finally, after all candidate transformations have been evaluated, the one that gives the highest number of inlier centroid matches gets selected. To discriminate between transformations leading to the same

inlier ratios, we choose the one that minimizes the highest distance between matched centroids.

5) **Transformation Refinement:** The transformation selected in the previous stage, which we denote as $\boldsymbol{T}_{\mathcal{MS}}^{\text{coarse}}$, is used as a prior in an ICP step that refines the alignment of the source and target point clouds, yielding an improved transformation $\boldsymbol{T}^{\text{ICP}}$. To reject unsuccessful registrations, we apply a threshold d_{ICP}^{*} to the Root Mean Square Error (RMSE) of ICP, which is computed using only the segments whose centroids have been matched in the geometric verification step. The final transformation between the source and the target maps, $\boldsymbol{T}_{\mathcal{MS}}$, is then computed as follows:

$$\boldsymbol{T}_{\mathcal{MS}} = \boldsymbol{T}^{\text{ICP}} \cdot \boldsymbol{T}_{\mathcal{MS}}^{\text{coarse}} \tag{4.2}$$

6) **Map and Segments Alignment:** The transformation obtained from the registration process, $\boldsymbol{T}_{\mathcal{MS}}$, is used in the last stage to align the original maps. In addition, overlapping segments from the source and target maps are combined, forming single object models. We merge each pair of previously matched segments in a common point cloud, which is then downsampled using a voxel grid filter. This way, the output of the registration pipeline is not only a map containing the registered input point clouds but also the segmented, better-reconstructed object models, which are fed into the Object Inventory described in Section 4.3.5 to aid the manipulation tasks.

4.3.5 Object Inventory and Dynamic Object Handling

To properly deal with potentially movable object instances in the map, we set up an inventory where, for each segmented object, we store its globally referenced point cloud $\boldsymbol{P}_i$, its centroid position $\boldsymbol{r}_{c,i}$, and an identification number i. These object instances are generated by the segment-based registration process, when an external map is available, or by simply segmenting the LiDAR map using the module described in Section 4.3.3. The inventory is then used to aid the manipulation tasks by constantly updating the objects' locations in the map, as described in the remainder of this section.

When an object is grasped, we associate the transformation between the map frame $\mathcal{M}$ and the gripper frame $\mathcal{G}$, $\boldsymbol{T}_{\mathcal{MG}}^{\text{grasp},i}$, given by the state estimation, with the corresponding object instance stored in the inventory. In addition, we remove all

map points inside a sphere of radius r_{obj} centered around the grasped object's centroid in the LiDAR map. After lifting the robot's arm, the hole created in the map will be replaced gradually by newly detected points on the ground. At the same time, the self-filter described in Section 4.3.2 will prevent the object from being remapped as long as it lies inside the gripper.

As soon as the object is released in a new position, we again extract from state estimation the transformation representing the gripper pose with respect to the map frame, $\boldsymbol{T}_{\mathcal{MG}}^{\text{release},i}$, and use it together with the previously stored grasp pose to estimate the transformation $\boldsymbol{T}^i$ experienced by the corresponding object instance in the scene:

$$\boldsymbol{T}^i = \boldsymbol{T}_{\mathcal{MG}}^{\text{release},i} \cdot \left(\boldsymbol{T}_{\mathcal{MG}}^{\text{grasp},i}\right)^{-1} \tag{4.3}$$

This transformation is then applied to the object's point cloud and centroid. By representing the objects with their globally referenced point clouds, we implicitly keep track of their position and orientation in the map.

After releasing the object, a scanning motion is performed, which causes the object to be remapped by the LiDARs. At this point, an ICP step is triggered to realign the transformed object point cloud with the LiDAR map, correcting for potential inaccuracies in the predicted object's location. This way, the map and the Object Inventory are always kept consistent with the state of the environment. At the same time, the identified objects can be individually tracked as the robot is moving them.

4.4 Grasp Pose Planning Pipeline

The goal of the grasp pose planning pipeline (Figure 4.4) is to find viable grasp configurations to pick the segmented objects instances. In the case of the autonomous excavator, a *grasp configuration* is defined as a 6 DoF gripper pose where a contact configuration with the object can be performed. A grasp configuration with contact wrenches that span the object's origin is called a *force closure* grasp. The purpose of grasp pose detection is to find force closure grasps on the object of interest. The grasp pose planning consists of three steps:

- identification of the region of interest and generation of a planning point cloud
- detection of grasp candidates in the region of interest

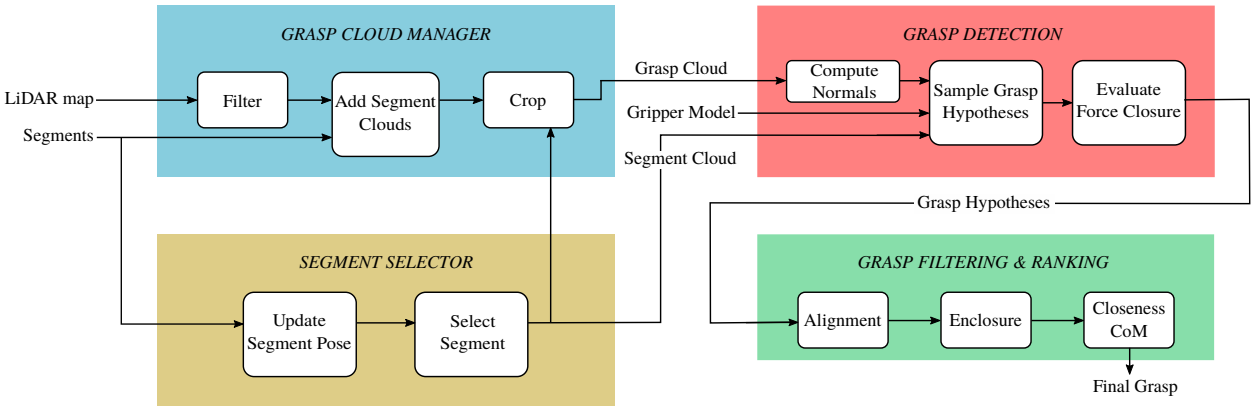


Figure 4.4: Overview of the grasp pose planning pipeline. The LiDAR map is augmented with the registered object segments and cropped to the region of interest around the selected segment. The generated grasp cloud is used for sampling grasp hypotheses which are filtered and ranked to obtain the final grasp.

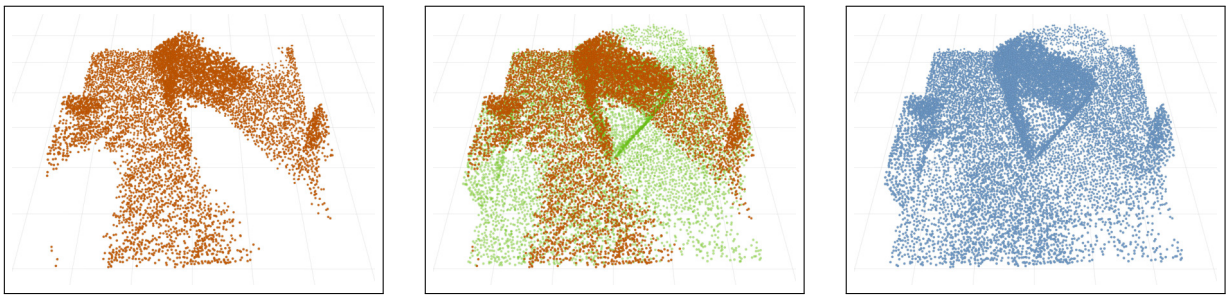
- subsequent filtering and ranking of the candidates to obtain a feasible grasping configuration

We will focus on finding grasp poses for two-finger grippers, as the excavator is equipped with a two-jaw angular gripper. Both jaws are mechanically connected and are moved by the same hydraulic actuator, giving one DoF. The jaws are mounted on the base part of the gripper that we will further refer to as palm. For collision checking between gripper and planning point cloud, the gripper shape is approximated with convex polyhedra enclosing the jaws and the palm.

4.4.1 Grasp Planning Point Cloud

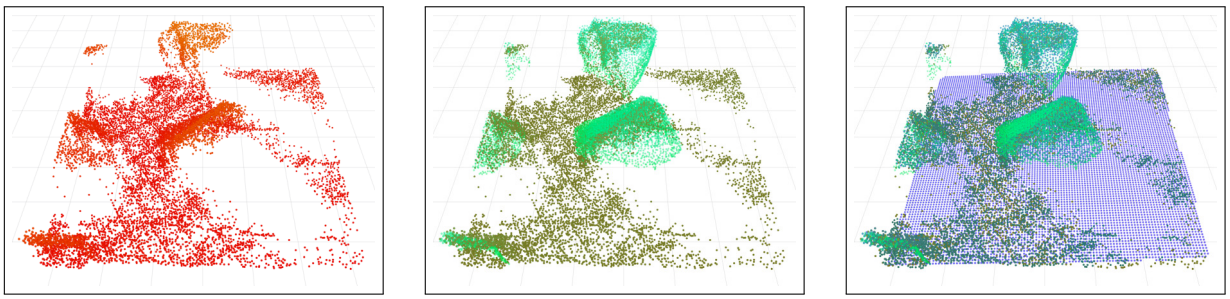
The grasp pose detection tries to find feasible grasp configurations directly on a point cloud representing the proximity of segmented objects, called the *grasp planning point cloud*. This point cloud allows to evaluate the quality of grasps and consider the collision with surrounding objects and the ground. In a typical application, the decision which objects to grasp would come from a higher level planning instance and depends on the structure to be built (see Chapter 7). In our case, we decide to grasp the largest segment with the most points from all the available segments $\boldsymbol{P}_i$ (Section 4.3.3). Starting with the largest segment is motivated by using them first for construction.

In order to obtain the planning point cloud, we crop the LiDAR-based map (Section 4.3.2) around the location of the selected segment's centroid $\boldsymbol{r}_{c,i}$ with a margin



(a) (b) (c)

Figure 4.5: Given an externally built scene point cloud, the grasp planning cloud is obtained by cropping the LiDAR map in a region of interest around the desired Segment (a) and merge it with the registered external point cloud (b). The merged map is down-sampled to get a uniform distribution (c).

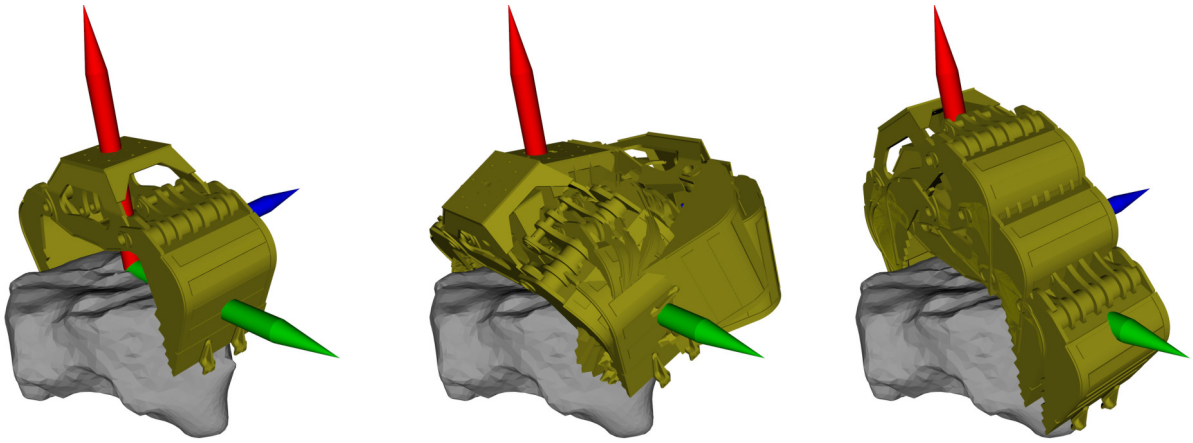


(a) (b) (c)

Figure 4.6: Without external point cloud, the grasp planning cloud is obtained by cropping the LiDAR map in a region of interest around the desired Segment (a). The registered segments are merged to the planning cloud (b) and a ground plane is fitted to fill holes in the map due to occlusion (c).

corresponding to the full gripper width (Figure 4.5a) and combine it with the registered external point cloud (Figure 4.5b). The merged maps are down-sampled to obtain a uniform point density on the grasp planning point cloud (Figure 4.5c).

If no external point cloud is available, e.g., after relocating the segments from their initial position, we could only rely on the LiDAR map for grasp planning (Figure 4.6a). However, the LiDAR-based map may be partially cluttered and occluded by the objects themselves due to the viewpoint, leading to holes in the map where no data points are available. In order to provide a complete map for grasp pose planning without an external map, the aligned object models from the Object Inventory are merged to the planning point cloud, assuring that we also have information on the opposite side to the LiDAR direction (Figure 4.6b). There might still be holes in the ground plane due to the occlusion by the objects, leading to grasps



(a) (b) (c)

Figure 4.7: A local reference frame $\mathcal{F}$ is assigned to a surface sample point with the x-axis pointing away from the object surface (a). Multiple orientations are generated for each sample point by rotating the local frame around the y-axis (b) and z-axis (c) in discrete intervals.

that would penetrate the ground. We fill these holes with a RANSAC plane estimation to find the ground plane and artificially augment over the complete planning point cloud (Figure 4.6c).

4.4.2 Grasp Detection

Grasp detection intends to generate a large number of grasp hypotheses on the planning point cloud that do not collide with the environment. The hypotheses can be filtered and ranked according to their applicability and success chance in a subsequent step. We sample N points on the desired segment $\boldsymbol{P}_i$ in the planning point cloud to generate the grasp hypotheses. Each sample p is assigned a local reference frame $\mathcal{F}$ by evaluating the Eigenvectors of the matrix

$$\boldsymbol{M}(p) = \sum_{\bar{p} \in \boldsymbol{B}_r(p)} \boldsymbol{n}(\bar{p})\, \boldsymbol{n}(\bar{p})^\top, \tag{4.4}$$

where $\boldsymbol{n}(\bar{p})$ is the outwards pointing unit surface normal at point $\bar{p}$, and $\boldsymbol{B}_r(p)$ is the r-ball around point p. The local reference frame $\mathcal{F} = [\boldsymbol{\nu}_3(p), \boldsymbol{\nu}_2(p), \boldsymbol{\nu}_1(p)]$ is composed of the Eigenvectors, where $\boldsymbol{\nu}_1(p)$ corresponds to the largest Eigenvalue and $\boldsymbol{\nu}_3(p)$ to the smallest one. This assures that the x-axis of the reference frame is

pointing away from the object, and the z-axis is pointing along the axis of minimal curvature.

We generate multiple grasp orientations for each sample point p by rotating the local reference frame around the y- and z-axis in discrete intervals (see Figure 4.7). The inverse x-axis of the rotated grasp orientation is denoted as the approach direction $\boldsymbol{n}_{\text{app}}$ of the gripper. A convex polyhedral gripper collision model is moved along the approach direction, with several opening angles, until palm or jaws are in contact with the grasp planning point cloud. We add a sampled pose to the list of grasp hypotheses if the closing region of the jaws is not empty.

Note that the grasp detection does not require a perfect segmentation of the object as the planning point cloud represents the closer vicinity of an object. However, faulty segmentation, like merging close objects to one segment, may cause that grasp contact points are placed on several different objects, which is undesired because it reduces the grasp success rate, and we are generally interested in manipulating one object at a time.

4.4.3 Grasp Filtering and Ranking

The grasp detection generates hundreds of grasps for a single object that are filtered and ranked according to their applicability and likelihood of success to obtain the desired grasp configuration.

Force Closure

In the first step, the grasp hypotheses are evaluated for force closure. We use a grasp classifier using a four-layer Convolutional Neural Network (CNN) presented in [81] that predicts if a grasp is a force closure based on the planning point cloud and a 2-finger gripper model. This classifier is applied to reduce the detected grasp hypotheses, predicting whether the grasps are force closure – however, many candidates may still be valid.

Grasp Cost

The force closure grasps are ranked and filtered by task-specific criteria that proved to provide reliable grasps. First, we rank the grasps according to their alignment with the approach direction. Given the upwards directed normal of the ground

plane $\boldsymbol{n}_{\text{ground}}$ and the approach direction of a grasp hypothesis (the normal pointing away from the palm) $\boldsymbol{n}_{\text{app}}$, the alignment cost c_{align} is given as the angle between the two vectors

$$c_{\text{align}} = \arccos\left(-\boldsymbol{n}_{\text{ground}}^{\top}\boldsymbol{n}_{\text{app}}\right). \tag{4.5}$$

We prefer approaching directions that are normal to the ground plane because they correspond to a nominal configuration of the excavator's arm. Furthermore, they help ensure that the arm is not colliding with surrounding objects. The 50 % grasps with the lowest alignment cost c_{align}, or a minimum of 40, are selected and ranked on how much the gripper encloses the object. An enclosing (or power) grasp is preferable to a pinching (or precision) grasp as it is more likely to withstand disturbances. The enclosing cost c_{enc} is represented by the distance between the palm and the centroid (CoM) in the approach direction

$$c_{\text{enc}} = \boldsymbol{r}_{\text{CoM}}^{\top}\boldsymbol{n}_{\text{app}}, \tag{4.6}$$

where $\boldsymbol{r}_{\text{CoM}}$ is the position vector from the palm center to the object centroid $\boldsymbol{r}_{c,i}$. Again, the 50 % grasps with the lowest cost, but a minimum of 20, are selected. Finally, we minimize the closeness of the grasp to the centroid of the object and select the best grasp based on the cost c_{dist} for execution. The closeness of a grasp to the centroid is given by the distance

$$c_{\text{dist}} = \frac{\|\boldsymbol{r}_{\text{CoM}} \times \boldsymbol{n}_{\text{app}}\|}{\|\boldsymbol{n}_{\text{app}}\|} \tag{4.7}$$

between the centroid and a line along the approach direction going through the Tool Center Point (TCP). This criterion is motivated by the need to avoid torsional moments on the grasped object during motion, leading to rotational shift of the object in the gripper. The best-ranked grasp is selected to be executed.

4.5 Experiments

To show the applicability and repeatability of the presented system, we implemented the different modules using ROS [25] and integrated them on the robotic excavator HEAP to perform autonomous manipulation of large objects. The goal is to map, segment, and grasp a set of randomly placed, irregularly-shaped stones au-

tonomously and move them to user-predefined positions while constantly updating the map in the process, as shown in the complementary video footage[1].

4.5.1 Experimental Setup

We use a set of seven gneiss stones which show variety in properties like shape and size, ranging from $0.5\,m^3$ to $1\,m^3$ approximately. The geometric stone models are not available beforehand and therefore need to be discovered and segmented on the fly.

We perform two experiments, each of them starting with the stones randomly placed on rough terrain within reach of the excavator's arm. In a first step, a point cloud of the initial stone arrangement is reconstructed from the data acquired by the drone-borne VI-Sensor using the method explained in Section 4.3.1 (see Figure 4.8 for an example). Upon creating the vision point cloud, we start building the LiDAR map online by swinging the excavator's cabin. When the LiDAR sensors have scanned the region of interest, the vision point cloud is registered into the LiDAR map using the approach described in Section 4.3.4, and the segmented object instances are fed into the Object Inventory. Then, the robot starts planning viable grasp configurations and paths to pick the detected objects and place them at different user-predefined positions while updating the map accordingly. In the scope of these experiments, the stones are moved to equally-spaced fixed locations on the ground. In order to achieve actual architectural construction, additional planning is necessary to decide the exact location of the objects in the target structure. Assembly planning is an involved process as not only geometric fit but also structural stability and functionality of the target structure have to be considered. We further discuss this in Chapter 7.

All the excavator's movements are planned and executed autonomously except for closing and opening the gripper We place an operator in the cabin for supervision and safety only. For motion planning, we use a whole-body trajectory planning framework that is suitable for both wheeled (legs) and non-wheeled (arm) limbs [92]. However, since the focus of this work is stone manipulation and not the traversing of the terrain, we keep the excavator base always at the same desired location during manipulation. For the arm motion itself, a simple heuristic-based planning approach suffices. The arm trajectory is composed of waypoints that re-

1 Watch the accompanying video: https://youtu.be/4bc5n2-zj3Q

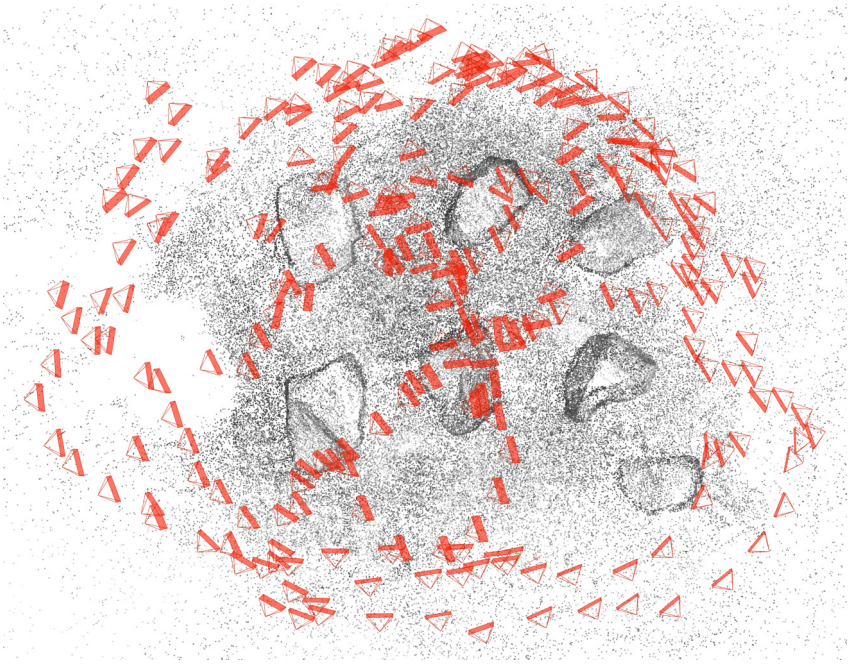


Figure 4.8: Point cloud of the second experiment's initial stone arrangement, reconstructed by the vision-based mapping framework COLMAP. The images were recorded by a drone and the red frusta represent the camera poses from which the registered images were taken.

spect collision constraints with the excavator's cabin and legs, and approaches the start and goal poses perpendicularly to the ground. These waypoints are then interpolated with Hermite splines to put together a trajectory. The arm controller used to track the trajectory relies on a hierarchical optimization-based inverse kinematics approach that computes joint velocities and enforces kinematic limits [93].

In the first experiment, the stones are grasped and moved once, whereas, in the second one, the process of grasping and relocating all the stones is repeated four consecutive times without resetting the map. With this, we demonstrate that the perception pipeline effectively handles objects being moved by the robot. Therefore, the map can be used for planning grasp poses and safe motions throughout an extended application.

4.5.2 Segmentation and Global Registration

Figure 4.9 illustrates the maps obtained from the initial mapping phase in the two experiments before moving any stones. These maps are created by swinging the excavator's cabin for about 20 seconds before triggering the registration routine to

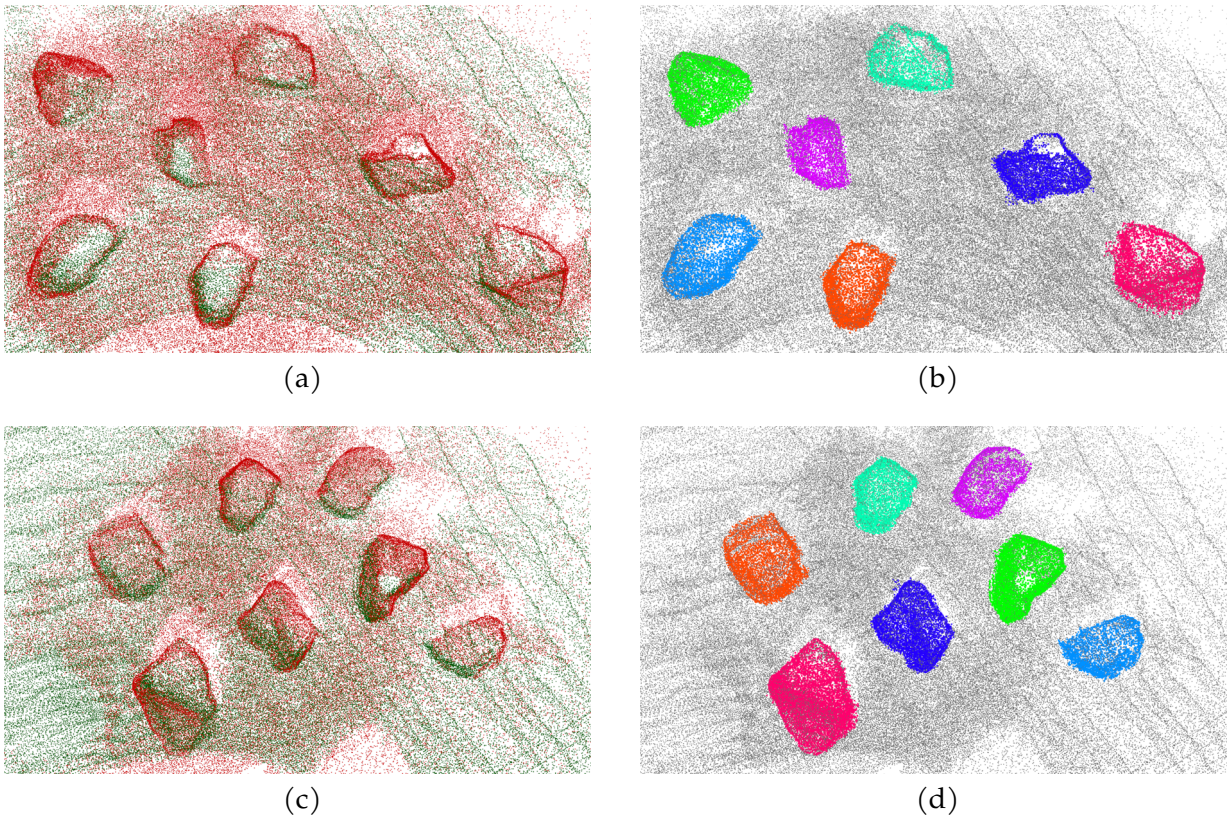


Figure 4.9: Example results of the map registration and segmentation process during the first (a, b) and the second (c, d) experiments. The left pictures show the aligned LiDAR (green) and vision (red) maps, whereas the right pictures illustrate the merged map (grey) overlaid with the segmented object instances (shown in different colors).

align the previously reconstructed vision point cloud with the LiDAR map, which is being built online. For each stone arrangement, we show the registered vision and LiDAR-based point clouds, as well as the segmented object instances, which are obtained after merging corresponding segments extracted from both input point clouds. These qualitative results evidence that the registration module is beneficial for adding information in occluded regions of the LiDAR map and improving the completeness of the segmented object models.

We evaluated the robustness of the segmentation and registration modules against a variable number of accumulated LiDAR scans. Therefore, we used additional LiDAR data collected while swinging the excavator's cabin after setting up each of the experiments mentioned above and the reconstructed vision point clouds of the two initial stone arrangements. The recorded data is played back and, while the LiDAR map is being built, the registration routine is triggered every five seconds

Table 4.1: Mean computation times and standard deviations (in ms) of each step involved in the registration process, as computed on an Intel Xeon E-2176M CPU.

Submodule	Experiment 1		Experiment 2	
	LiDAR	Vision	LiDAR	Vision
Segmentation	52 ± 8	58 ± 1	61 ± 17	58 ± 1
Keypoint extraction	57 ± 7	47 ± 2	48 ± 14	53 ± 2
Keypoint description	17 ± 2	18 ± 1	20 ± 12	21 ± 1
Descriptor matching	279 ± 23		284 ± 64	
Geometric verification	8 ± 1		7 ± 2	
Transform. refinement	470 ± 226		420 ± 275	
Total	998		974	

for a total duration of one minute. Each trial is defined as successful if the RMSE of the final ICP step, as defined in Section 4.3.4, lies below d^*_{ICP}, and the seven stones get detected as single objects in the map.

Results show that approximately 15 seconds need to be spent building a LiDAR map of the initial region of interest, which in our experiments is about 50 m^2, before achieving the first successful registration. The subsequent registration attempts (here, 20 attempts in total) are executed with an overall success rate of 90 %, proving our segmentation and registration methods to handle different point densities in the LiDAR map. The observed failure cases are caused by the fact that many of the extracted 3D features are wrongly matched, and the algorithm cannot find a geometrically consistent transformation that is close enough to the optimal one. However, in the real-world application, these cases are detected by monitoring the RMSE of the final ICP step. We let the system further accumulate LiDAR scans until a successful registration is achieved.

For the sake of completeness, in Table 4.1, we report the computational times of the individual steps in the registration pipeline, including the initial segmentation of the input point clouds, when executed on an Intel Xeon E-2176M CPU. As it can be observed, the complete segment-based registration routine is executed in

Table 4.2: Sizes of the LiDAR-based and vision maps (mean and standard deviation given in number of points) in the registration experiments. Note that the size of the LiDAR maps varies as the number of accumulated scans increases, whereas the vision point clouds are pre-computed and therefore their size is fixed.

Experiment 1		**Experiment 2**	
LiDAR	**Vision**	**LiDAR**	**Vision**
92301 ± 9443	73547	123564 ± 29622	98190

approximately 1 second in both of our experiments, which makes our approach suitable for online operation. The sizes of the maps used to perform the experiments detailed in this section are reported in Table 4.2. Additionally, the most relevant parameters of the perception pipeline are summarized in Table 4.3.

4.5.3 Grasp Pose Planning

The stones were grasped and relocated five times each, resulting in a total of 35 grasp attempts. The best-ranked grasp from the grasp pose planning pipeline was selected and could be executed with a success rate of 88.6 %. A grasp is deemed successful if it is collision-free, force closure, and allows moving the object without dropping it. Grasp attempts that are not successful are detected by applying a closing force while lifting the stone. In case of slippage or dropping, the gripper jaws further close during the lifting process, which is detected by the wire draw encoder of the gripper piston. A sudden drop of the object after lifting can be recognized by monitoring the arm cylinder forces. Note that this notion of grasp success does not account explicitly for any safety margin, i.e., whether the grasped object can withstand an external disturbance force. However, it implies robustness to orientation changes and shaking during placing trajectory execution. If the best-ranked grasp could not be executed successfully, the grasp pose was manually adjusted to relocate the object.

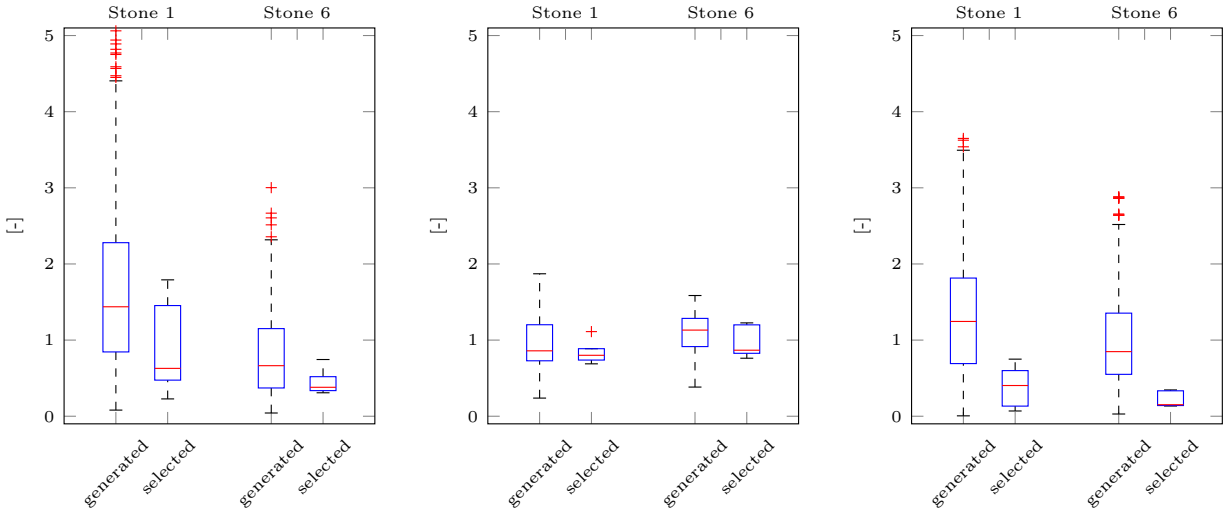
The grasp filtering and ranking are designed to successively filter grasps with high costs for specific criteria and finally select a grasp that achieves the lowest cost for the distance between the TCP and the stone centroid. Figure 4.10 compares the

Table 4.3: Parameters of the perception pipeline used during the experiments.

LiDAR-based Mapping	
Min. distance between poses, d_{poses}	1 cm
Num. of scan pairs per submap, m	10
Local map radius, r_{map}	10 m
Voxel grid resolution, r_{voxel}	3 cm
Min. point count per voxel, n_{min}	1
Radius for object removal, r_{obj}	1 m
Segmentation and Registration	
Plane fitting distance threshold, d_{plane}	10 cm
ISS detector salient radius, r_{ISS}	6 cm
SHOT descriptor radius search, r_{SHOT}	10 cm
Max. num. of candidate transformations, b	10
Max. dist. btw. matched centroids, d_{match}	1 m
ICP RMSE threshold, d^*_{ICP}	3 cm

costs of the grasp filtering and ranking by the three criteria consisting of the alignment of the approaching direction to the ground plane (a), the enclosing of the grasp as the distance between the centroid and the palm (b), and the closeness of the grasp to the centroid (c) for all generated grasps predicted being force closure and the finally selected grasps. For better comparison, each grasp cost is normalized by its median value of all generated grasps. We show the evaluation for two different stones labeled with 1 and 6 in Figure 4.11. Whereas stone 1 has a more roundish shape is stone 6 an example of a flat stone. The evaluation for the other stones shows similar results and is left out for the sake of brevity.

We can see that for the roundish stone 1, the generated grasp hypotheses have a more considerable variation in the cost representing alignment to the ground plane (see Figure 4.10a) than for the flattish stone 6, meaning that the generated grasps



(a) Alignment (b) Enclosing (c) Distance CoM - TCP

Figure 4.10: Grasp costs of the hierarchical filtering steps in terms of the alignment of the grasp approaching direction to the ground plane normal in (a), the enclosing, measured as distance from the palm to the centroid in approaching direction in (b), and the closeness of the tool center point to the centroid of the stone in (c). Shown are the cost distribution of all generated grasps predicted being force closure, and the selected grasps of stone 1 and 6. For comparison, each grasp cost is normalized by its median value for all generated grasps.

are approaching the object from directions all around it. Whereas stone 6 is flat and has to be pinched by the gripper, stone 1 is better suited for an enclosing grasp as it is larger and more roundish, resulting in a better enclosing cost (see Figure 4.10b). Because the distance between the TCP and the centroid is the last filtering criteria, we can observe the most significant impact on the cost for the selected grasps (see Figure 4.10c). We see that sequential filtering improves all cost criteria while maintaining a balance between them, leading to a success-promising grasp selection.

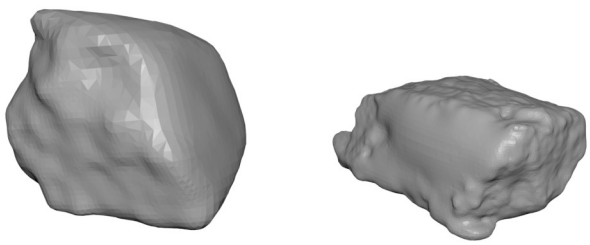
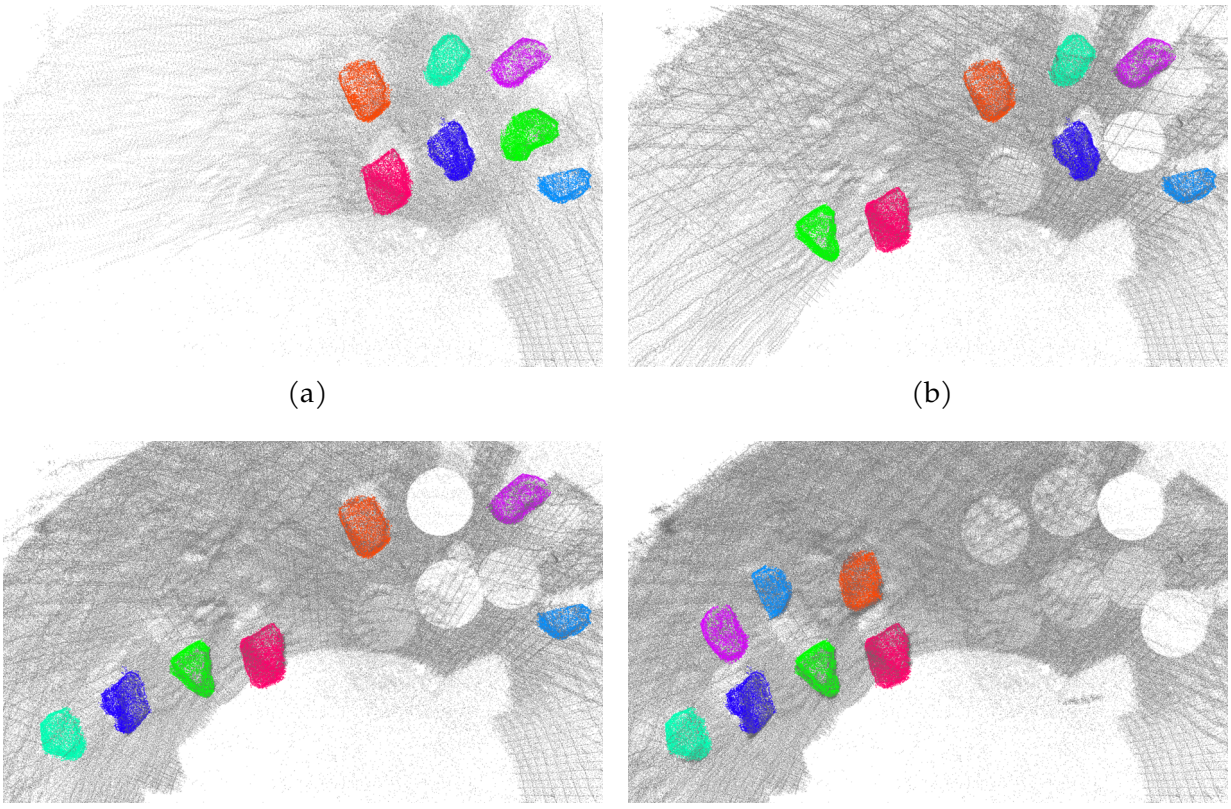


Figure 4.11: Mesh reconstruction from point-cloud data of stone 1 (left) and stone 6 (right). Stone 1 has a rather roundish shape, whereas stone 6 is flatish.



(a) (b) (c) (d)

Figure 4.12: State of the excavator's map at four different instants of the second experiment: (a) before moving any stones, (b) after moving the 2nd stone (c) after moving the 4th stone and (d) after moving the 7th stone. The object instances are effectively tracked (note that same color is always associated with the same stone across all figures) and realigned with the map when moved by the robot.

Four grasps out of 35 had to be adjusted manually because the selected grasping pose could not be executed successfully (slippage of the stone). Especially for flat but relatively thin objects, the grasp detection provided only a small number of grasp hypotheses (around 10) labeled force closure due to the point cloud's noise and the fact that the grasp has to be placed close to the ground. Thus, the subsequent filtering and ranking select the grasp based only on where the TCP is closest to the object's centroid without considering the other criteria.

4.5.4 Dynamic Object Handling

Figure 4.12 depicts the map state overlaid with the segmented objects in four different instants during the second experiment mentioned in Section 4.5.1. The il-

lustrations qualitatively show that the object models are effectively tracked (the same color is always associated with the same stone across all four figures) and realigned with the LiDAR-based map after being moved. Although no ground truth data is available to quantify the precision of the object models and their estimated locations in the map, the high success rate achieved by the grasp pose planning module (see Section 4.5.3) indirectly indicates that the overall accuracy of the perception pipeline is sufficient for the task at hand.

4.6 Summary

This chapter has introduced an integrated perception and grasp pose planning system for autonomous manipulation of large-scale irregular objects with a robotic excavator. The core of the perception pipeline constitutes a LiDAR-based mapping algorithm coupled with a segmentation approach that enables detecting object-like instances in the observed scene. Furthermore, it is enhanced with multi-sensor data fusion capabilities, allowing for the registration of externally built maps of the regions of interest, e.g., 3D reconstructions from images captured by a drone-borne camera, as shown in the experiments. An inventory helps to keep track of the identified objects' poses during manipulation. The grasp pose planning pipeline, on the other hand, is capable of sampling grasp hypotheses in a 3D scene given the information provided by the mapping system, i.e., a point-cloud map of the robot's surroundings and the segmented objects in it. Besides selecting the grasp hypotheses by predicted force closure, a custom filtering and ranking step increases the grasp reliability. The described mapping and manipulation tools were used in an actual application to build the wall-like irregular stone-based assemblies on an architectural scale (see Chapter 5, 7).

5

Grasp Pose Planning and Object Reorientation

This chapter incorporates material from the following publication:

Wermelinger, M., Johns, R., Gramazio, F., Kohler, D., & Hutter, M. Grasping and Object Reorientation for Autonomous Construction of Stone Structures. *IEEE Robotics and Automation Letters* **6**, 5105, (2021).

Video: https://youtu.be/aS0Kttqd_Gk

The assembling of elaborate structures on construction sites requires manipulating objects in potentially cluttered and obstructed scenes. Planned placement locations of objects in the already built structure are usually close to adjacent objects within a densely packed assembly. To execute this plan, one of the biggest challenges is the reliable manipulation of irregularly shaped objects, including the grasping and reorientation necessary to place them into tightly constrained locations.

In this chapter, we enhance the approach for generating grasp configurations of an excavator-mounted 2-jaw gripper presented in Section 4.4, enabling sampling of grasps on a specific object only and avoiding collision at its current and desired location. We combine reconstructed object meshes and a LiDAR-based map to evaluate the force closure of grasps using a learning-based classifier while also considering collision constraints at both ends of the pick-and-place sequence to assess the feasibility of relocating an object to the desired location. Furthermore, we include a method for reorienting objects if no direct placement is feasible. It aims to find a stable intermediate placement pose by letting the mesh reconstruction of the object settle in a physics simulation and verifying that collision-free grasp configurations are available for placement. Conventional approaches for assembling complex-shaped objects in a robotics cell focus on clearly defined geometries [94], simplifying the grasp planning and eliminating alignment errors, whereas we handle arbitrarily shaped objects without simple geometric features like straight edges. Besides, an

Figure 5.1: The autonomous excavator HEAP assembling a dry stone retaining wall with irregularly shaped stones. As shown in the inset, an arm mounted LiDAR is used for mapping and stone localization.

automated excavator has a limited end-effector range of motion compared to an articulated robot arm, which needs to be incorporated in the grasp planning and object reorientation method.

The proposed approach enabled constructing the world's first large-scale wall using an autonomous excavator (more extensive than the robot). We manipulated more than a hundred stones that each weighs several hundred kilograms and have a unique and highly diverse geometry (see Figure 5.1). Furthermore, we show a detailed evaluation of the grasp and reorientation success rate and discuss slippage cases and possible improvements.

5.1 Related Work

Grasp planning is often divided into *analytical* and *empirical* (data-driven) methods. Analytical methods evaluate the performance of a grasp according to physical properties such as stability, equilibrium, dexterity, and dynamic behavior but

only under the assumption of having the exact object model and its pose in the scene [95]. Grasps could be synthesized from a constraint optimization problem over one or several measures of the mentioned properties that is hard to solve for our non-convex objects. On the other hand, empirical methods rely on sampling grasp configurations and ranking them according to metrics typically coming from simulation trials, physical trials, or human labels [96]. They place more weight on the perceptual object representation and better accommodate uncertainties in perception and execution, such as those present in our case of reconstructing and localizing irregularly shaped objects. Multiple empirical methods rely on grasp detection approaches to generate grasp configurations in cluttered scenes directly, without the need to first localize single object instances [97–99]. These approaches begin by generating a large number of grasp candidates on the input scene point cloud, usually originating from an RGB-D camera. They then evaluate the probability of the candidates being a grasp, e.g., in terms of force closure, using a classifier or regression system trained on a large amount of labeled data. These methods generalize well for new objects, as they detect grasps based on graspable regions independent of object instances but are prone to find contact points that are spread across multiple objects. In our outdoor case, depth images of the scene are not directly available. Therefore, we perform the grasp detection on the point cloud generated by 3D LiDAR mapping. We combine the LiDAR-based map with reconstructed object meshes to complement the point cloud used for grasping and ensure that contact points are only located on the object of interest. To sample grasp hypotheses on the point cloud and classify the grasp candidates, we use a similar approach as presented in [81]. However, we perform a further selection step from the sampled grasps where the remaining candidates are filtered for heuristic criteria derived from the specific task to obtain the final grasp candidate. The idea of filtering the generated hypotheses is that we are not necessarily interested in the one optimal grasp [100], but we are rather interested in a sufficiently good grasp that fulfills the task-specific criteria.

Object reorientation is the task of moving an object between different 3D orientations to make it accessible for further manipulation tasks such as assembly. Traditional methods use pick-and-place motions, where the manipulator grasps the object firmly and rotates it to the desired stable pose [101]. This process may repeat several times, depending on the robot's range of motion and workspace limitations, making it necessary to plan the motion sequencing resulting in a constraint satisfac-

tion problem [102]. In order to avoid time-consuming motion sequence planning, we design a reorientation method that allows the object to be brought to the desired orientation for placement within a single reorientation motion, taking advantage of the gripper's ability to rotate continuously. An alternative method to using less gripper and arm motion in constrained workspaces is pivoting [103]: an object is pinch-grasped and can passively rotate about the grasp axis. At the same time, it remains in contact with the ground surface during motion, allowing the object to reorient around an arbitrary edge and to decouple object rotation and gripper motion. Pivoting works best for regular, prismatic objects, which is not the case in our application, and would require multiple pivoting motions. Furthermore, the gripper has to be able to switch between a pivoting grasp and a firm grasp, e.g., by changing gripping force or the finger-object contact geometry [104], which is not possible for the hydraulic gripper of the excavator. More dynamic motion primitives for object reorientation such as throwing and catching [105] or in-hand pivoting with inertial forces [106, 107] have only been presented for simple object geometries with clearly defined rotation axes and are not suitable for rotating objects with the target size and weight.

5.2 Grasp Pose Planning

A *grasp configuration* for the autonomous excavator is defined as a 6 DoF pose of its 2-jaw angular gripper where a contact configuration with the object can be performed. We aim to find force closure grasps, which are grasp configurations with contact wrenches that span the centroid of the object. Compared to Section 4.4, we are looking here for grasp configurations that are viable at the pick *and* desired place location in the assembly structure. Additionally, we ensure that the contact configuration is spanning a single object solely. For collision checking between the gripper and planning point cloud, we approximate the gripper shape with convex polyhedra that encompass the jaws and palm. The collision check at the desired placement location requires the object mesh to be completely reconstructed beforehand. For reconstruction, we segment the point cloud of the excavator's surroundings to identify object instances, perform an initial grasp, and scan the object using excavator cabin-mounted LiDARs as described in Chapter 4. The grasp pose planning consists of three steps: generation of a grasp point cloud; detection of grasp candidates on the grasp point cloud; and subsequent filtering and ranking of

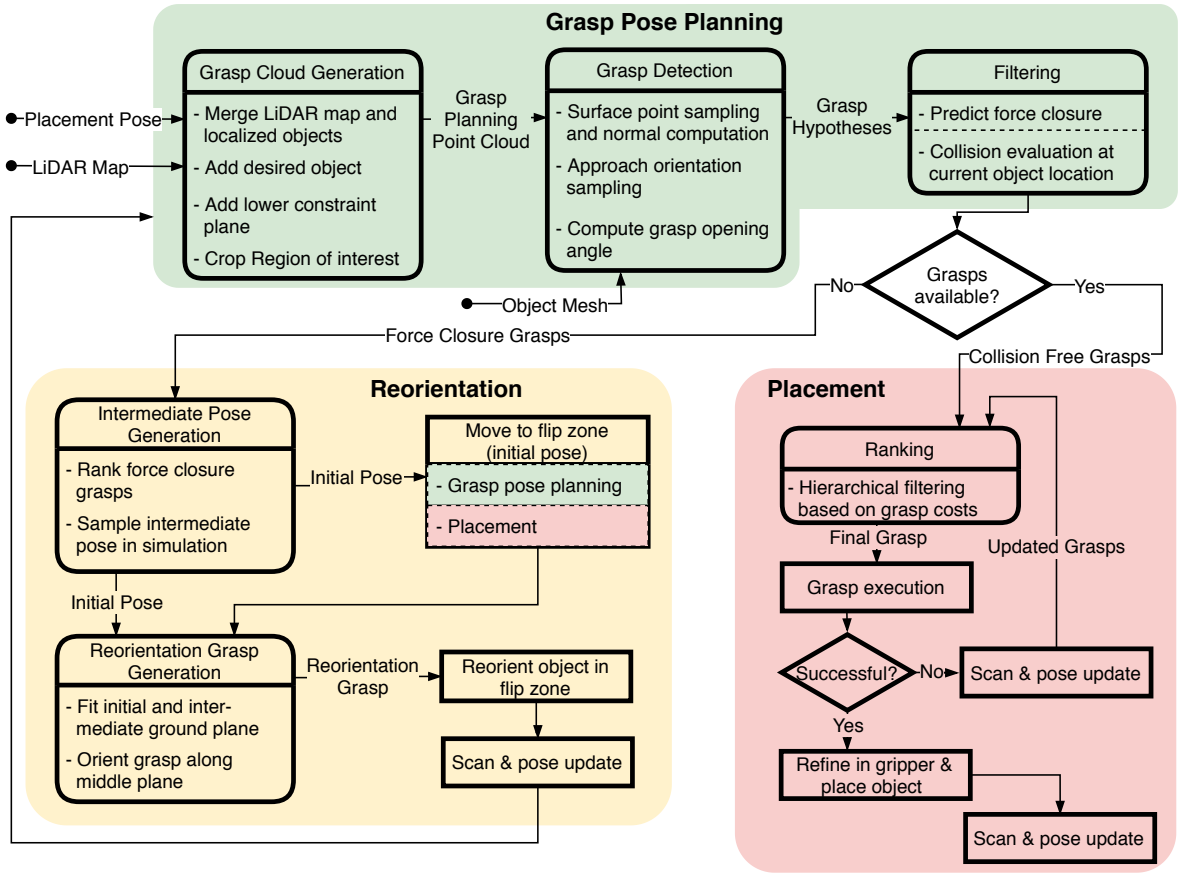


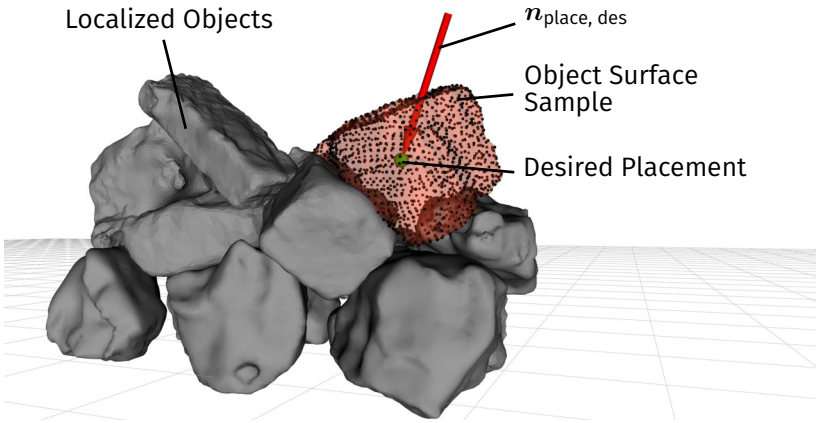
Figure 5.2: The grasp pose planning and object reorientation pipeline.

the candidates to obtain a feasible grasping configuration. If none of the sampled grasp configurations are valid at the pick location, a reorientation of the object (Section 5.3) is attempted. An overview of the grasp pose planning and reorientation procedure is depicted in Figure 5.2 and the accompanying video footage[1].

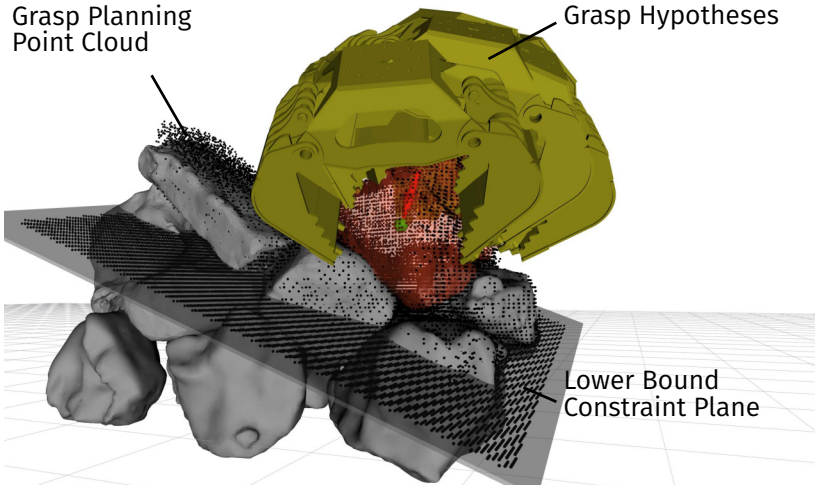
5.2.1 Grasp Planning Point Cloud Generation

The collision detection of possible grasp configurations is done directly on a point cloud representation of the scene, called the *grasp planning point cloud*. Grasp configurations are sampled at the desired placement location in the structural assembly, as ultimately, the grasp has to be designed such that the placement is feasible.

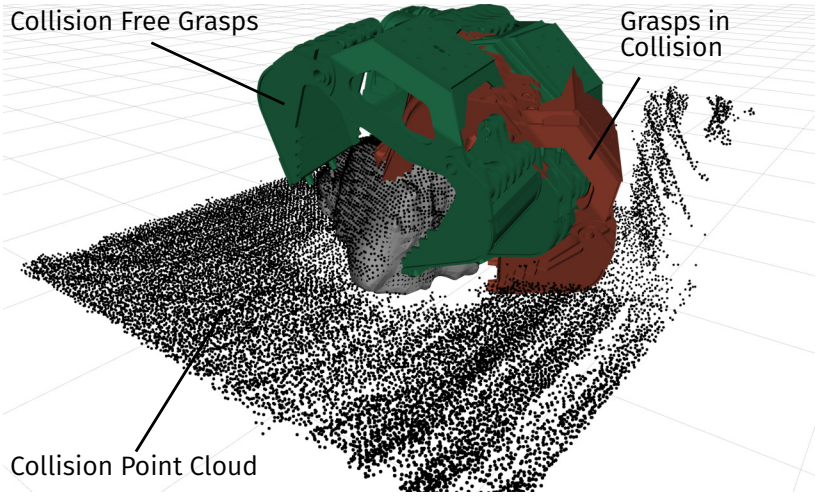
1 Watch the accompanying video: https://youtu.be/aS0Kttqd_Gk



(a) Grasp Samples at Placement Location



(b) Grasp Planning Point Cloud



(c) Collision Point Cloud at Pick Location

Figure 5.3: To place an object at a desired location in the assembly structure (red), the desired placement direction $\boldsymbol{n}_{\text{place, des}}$ with highest margin is computed. Object surface points are sampled to get the surface normal as possible grasp approach directions (a). Grasp hypotheses are generated on the grasp planning point cloud (b) and reprojected to the object pick location for collision detection (c).

LiDAR Mapping

To obtain the grasp point cloud, we create a map of the excavator's surrounding using a LiDAR sensor mounted on the stick (second link of the excavator's arm, see Figure 5.1). We use the laser-based graph SLAM approach that provides a consistent map without self-see points presented in Section 4.3.2, but skip the *Scan Pair Assembly* step, as we are using a single sensor in this application. Using an arm-mounted LiDAR sensor, it is possible to physically adjust its viewpoint to see the back of objects and scan regions in the map that are higher than the robot location itself, which is necessary to build tall structures like walls.

Object Models and Ground Plane

Even with a LiDAR that can change its viewpoint, the scene can still be incomplete due to occlusion, especially within the assembly itself. Therefore, we augment the map point cloud with point clouds of the object models localized in the scene and with the desired object at placement. We compute the desired motion direction for object placement, called desired placement direction $\boldsymbol{n}_{\text{place, des}}$ (see Figure 5.3a), by conducting ray casting from the object centroid at the desired pose (green sphere). Rays that do not hit the ground or surrounding objects are averaged to obtain the placement direction with the highest margin to obstacles. A lower bound constraint plane orthogonal to the desired placement direction is added at the lowest point of the desired object surface in the placement direction. The idea of the constraint plane is that points below this plane can be cropped and ignored, reducing the size of the point cloud significantly and speeding up the collision detection. Finally, the grasp planning point cloud is cropped to a box region of interest around the desired object position with an edge length twice the maximum gripper aperture of 1.8 m (see Figure 5.3b).

5.2.2 Grasp Detection

Grasp detection aims to generate a large number of grasp hypotheses on an object. A grasp hypothesis is valid if the gripper does not collide with the environment, verified by intersecting the gripper model with the grasp planning point cloud, and the contact configuration spans a single object. To generate the grasp hypotheses, we use the sampling presented in Section 4.4.2 but draw the sample points p only from

the point cloud representation of the desired object $\boldsymbol{P}_i$ (see Figure 5.3a) instead of the complete grasp planning point cloud. With this grasp detection approach, we can sample grasps only on the object of interest while ensuring no undesired contact with other objects or surroundings during placement.

5.2.3 Grasp Filtering and Ranking

The grasp detection generates hundreds of grasps for a single object that are subsequently filtered and ranked according to their applicability and likelihood of success. In the first step, the grasp hypotheses are evaluated for force closure as described in Section 4.4.3.

Collision at Pick Location

Subsequently, the valid grasps and the desired placement direction are projected to the actual location of the object $\boldsymbol{P}_i$ in the scene to check whether they are also valid for picking the object. Like the grasp planning point cloud, the LiDAR map around the object is augmented with the localized objects and used to evaluate the collision of the grasp hypotheses at the pick location (see Figure 5.3c). All colliding grasps are discarded. If none of the detected grasps is collision-free at the pick location, a reorientation of the object (see Section 5.3) is necessary to achieve the desired placement.

Grasp Cost

The remaining collision-free grasps are ranked and filtered by task-specific criteria that proved to provide reliable grasps (see Section 4.4.3). We modified the alignment cost c_{align} to prefer grasp approach directions (the normal pointing away from the palm) $\boldsymbol{n}_{\text{app}}$ aligned with the desired placement direction $\boldsymbol{n}_{\text{place, des}}$ because this gives the most margin with respect to a collision. Therefore, we compute the alignment cost c_{align} as the angle between the two vectors

$$c_{\text{align}} = \arccos\left(\boldsymbol{n}_{\text{place, des}}^{\top}\boldsymbol{n}_{\text{app}}\right). \tag{5.1}$$

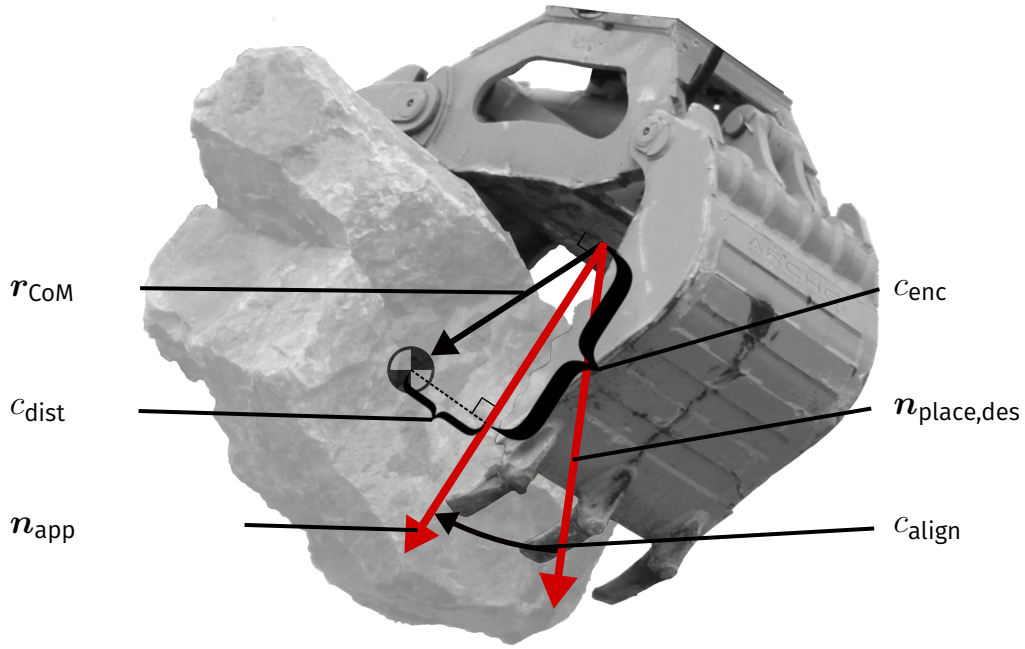


Figure 5.4: The grasp cost is dependent on the distance of the gripper palm to the centroid r_{CoM} and the alignment of the approach direction n_{app} to the desired placement direction $n_{\text{place, des}}$.

The enclosing cost c_{enc}, the cost for the closeness of a grasp to the centroid c_{dist}, and the ranking and filtering are presented in Section 4.4.3. All measures of the grasp costs are illustrated in Figure 5.4.

5.2.4 In-hand Pose Refinement and Failure Recovery

As the gripper is closing symmetrically and not adapting to the contacts, the stone might slightly move with respect to the gripper during the closing and lifting (while still maintaining the grasp). In order to increase execution accuracy, the stone pose in the gripper is updated while moving to the placement location. During the cabin swing motion, the gripper rotates to scan the grasped stone with the cabin roof mounted LiDARs, and the stone is re-registered with respect to the gripper using an ICP update (see Figure 5.5).

If the stone slips during the gripper closing and the grasp cannot be maintained, the gripper opens again, followed by a short mapping sequence to refine the stone's pose in the world frame by running an ICP update. The grasp with the next-best rank is executed to recover from slippage and to proceed with stone placement.

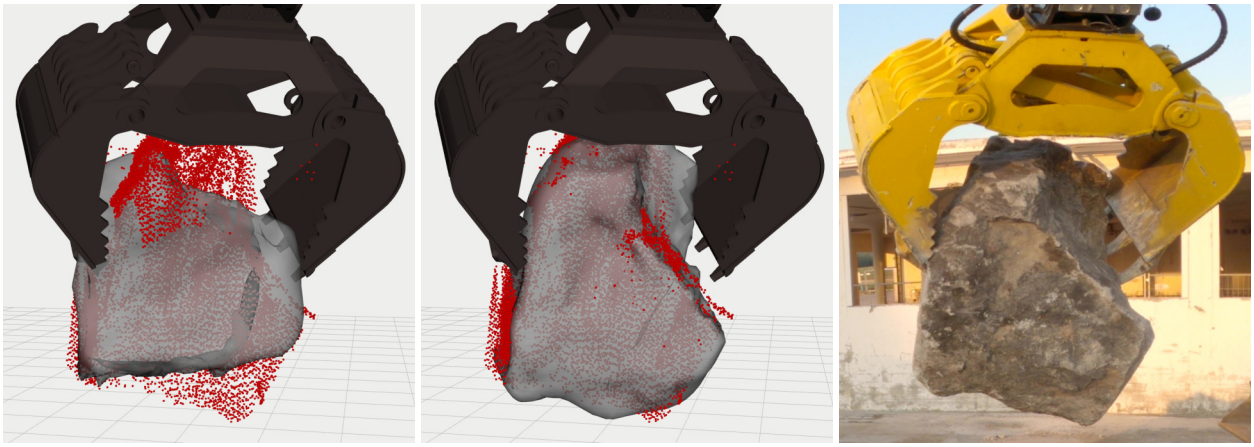


Figure 5.5: In-hand pose refinement: The gripper is rotated in front of the cabin mounted LiDARs to generate a scan (red) of the grasped stone (grey). An ICP update step is used to align the in-hand stone pose to the scan.

5.3 Object Reorientation

In order to allow for maximum flexibility in the online structural assembly planning, the assembly planner has no notion of the prior object orientation, meaning that it may request to place the objects in an arbitrary orientation. As such, it might not be possible to find a grasp pose that allows the object to be moved directly from its current pose to that desired location. This infeasibility is typically due to the excavator's limited range of motion or collision constraints at the pick and place locations. In this case, an intermediate reorientation of the object is necessary to bring the object from its current pose to a configuration that enables it to find a grasp pose suitable for placement.

5.3.1 Intermediate Pose Generation

First, we find a nominal desired grasp configuration at the desired placement pose by ranking all force closure grasp according to the grasp cost presented in Section 5.2.3 (including grasps with a collision at pick location) and use the best-ranked grasp. We preferably want to reorient the object to an intermediate pose such that the desired grasp approaches the object from the top, guaranteeing to be in the motion range of the excavator and having the least chance of occlusion. In order to find a physically consistent and stable intermediate object pose, the object is put in a physics simulation such that the placement direction of the desired grasp is per-

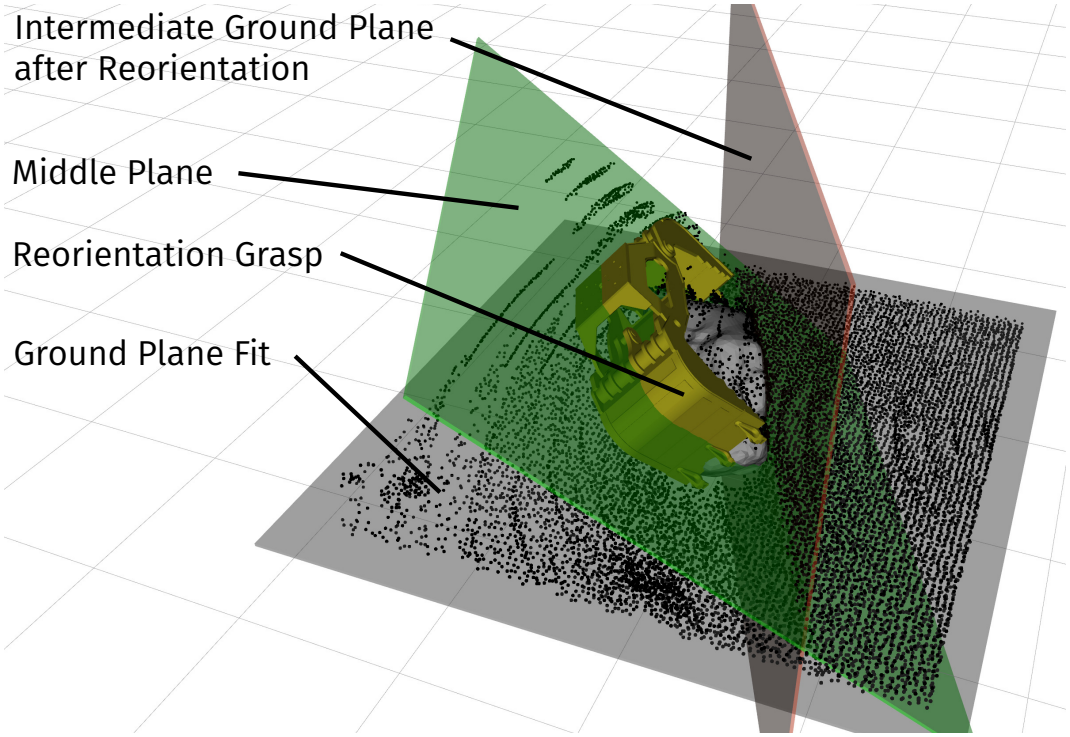


Figure 5.6: The reorientation grasp is aligned along the middle plane between the current ground plane and the desired intermediate ground plane after flip.

pendicular to the ground, and the object is slightly above the ground. The physics simulation is stepped forward to let the object settle on the ground, and we verify whether the desired grasp pose is still feasible, i.e., collision-free and reachable. If the desired grasp pose is not feasible on the settled object, the initial orientation in the simulation is slightly perturbed to obtain a different settled pose. This process is repeated until a feasible settled object pose is achieved that serves as an intermediate object pose. The reorientation process preferably takes place on even ground to avoid collisions, and in our case. the simulation uses flat ground as well. However, it works for arbitrary ground as long as it is mapped and represented in the simulation.

5.3.2 Reorientation Grasp Generation

First, we move the object from the initial to the intermediate pose. Therefore, a designated grasp is needed that respects the collision constraints from the initial and the intermediate pose. Namely, the surrounding ground planes of the initial and intermediate pose are quite restrictive collision constraints. They mutually ex-

clude many possible solutions found by the grasp pose planning pipeline presented in Section 5.2 and enforce grasp configurations aligned with both ground planes. Therefore, we compute the desired grasp configuration based on the ground planes at the initial and intermediate object pose (see Figure 5.6):

1. We project the intermediate ground plane with respect to the object frame to the initial object pose.

2. The approach direction of the reorientation grasp is oriented along the middle plane between the initial ground plane and the projected intermediate ground plane. The grasp is aligned such that the gripper jaws move along the middle plane.

3. The open polyhedral gripper model is moved along the approach direction until there is contact with the ground planes or the object, and the gripper jaws are closed until they are in contact with the object.

Suppose contact with the ground plane occurs before contact with the object. In that case, the intermediate ground plane and reorientation grasp can be offset slightly, such that the object is dropped from a small height to avoid collisions.

5.3.3 Flipping zone

The reorientation grasp approach direction is inherently located within the angular domain between the ground plane and a maximum tilt of $\frac{\pi}{4}$ rad from the ground plane. Because of the limited motion range of the excavator's joints, such an end-effector configuration can only be reached if the object is relatively close to the chassis. In order to facilitate the reorientation maneuver, the object of interest is therefore first moved to a predefined *flipping zone* on the side of the excavator's chassis. Additionally, it is rotated around the z-axis in the world frame, such that the approach direction of the reorientation grasp is pointing towards the chassis (see Figure 5.7). After moving the object to the flipping zone, we execute the reorientation grasp and lift the object to a height of 1.5 m above ground. It is finally rotated and placed at the same position but with the orientation of the intermediate pose. Each time the object is relocated, its pose in the world frame is refined by running an ICP update using the arm-mounted LiDAR to compensate for deviations during the grasping or settling of an object (see also Section 7.2.2). With the reori-

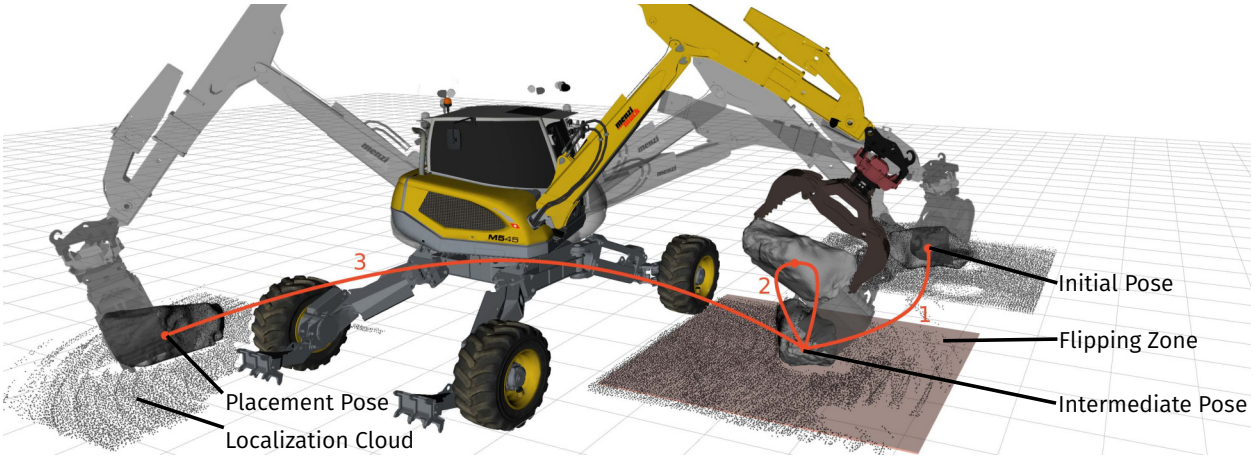


Figure 5.7: An object that has to be reoriented in order to be placed, is moved from its initial pose to the *flipping zone* close to the excavator's chassis (1). A regrasp is performed that allows to rotate the object to its desired intermediate orientation (2). Before moving it to its final placement pose, another regrasp is necessary that considers the collision constraints at placement (3). The object is localized each time it is moved using ICP registration of the object mesh to the LiDAR-based map.

ented object pose, the grasp pose planning pipeline is rerun to obtain a grasp pose for placing the object in the desired assembly location. Figure 5.8 shows the entire reorientation maneuver of an object placed in the flipping zone, from executing the reorientation grasp to the final grasping for placement.

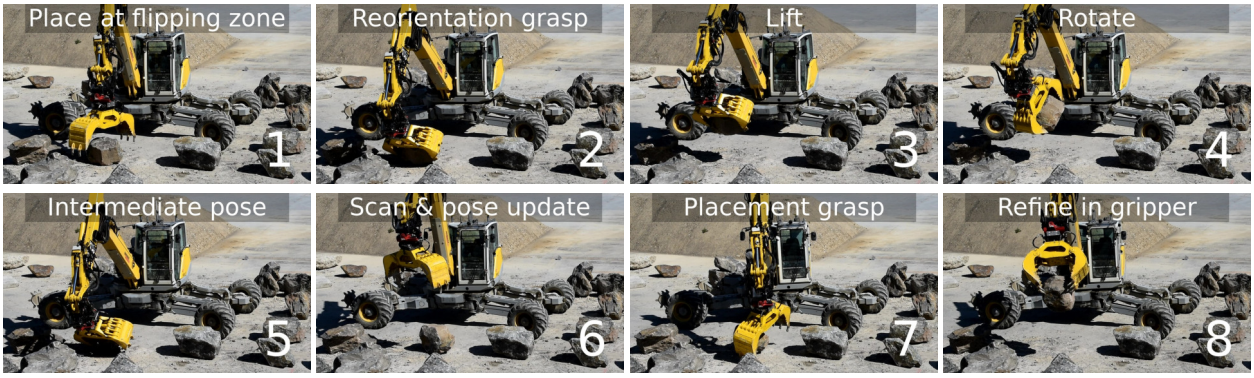


Figure 5.8: Object reorientation maneuver of a stone in the flipping zone, from the execution of the reorientation grasp, over the intermediate pose, to the final grasping for placement.

Table 5.1: Grasp attempt success rate.

# Grasps	Succ. Grasps	Slip
443	364 (82.2 %)	79 (17.8 %)

5.4 Experiments

The experiments were conducted using HEAP, a highly customized Menzi Muck M545 12t walking excavator developed for autonomous applications and advanced teleoperation [6] (see Section 4.2). The fully mobile 25 DoF excavator can lift objects to a height of 9 m and and freely manipulate items weighing up to 3000 kg. We measure the end-effector position with respect to the cabin through arm link mounted IMUs for these experiments. The IMU's measure the orientation of the links w.r.t. the world and are thus less prone to propagating an error down from joint play or inaccuracies in the conversion from piston position to joint position. A LiDAR mounted on the arm (near the gripper) is used to map and localize the objects, as it provides a top view of the available material and in-progress structure. Two additional LiDAR scanners are placed at the front edge of the cabin's roof to provide 3D scans for reconstruction and in-hand refinement of the objects.

5.4.1 Experiment Description

The applicability of the proposed grasp approach is directly shown in constructing a large-scale, double-faced dry-stone wall with dimensions approx. 10 m x 4 m x 2 m (W x H x D). In order to find stone place locations that comply with the target wall geometry and are structurally stable, we use the geometric assembly planner presented in Chapter 7. The wall planner solves for stable stone poses, using the reconstructed surface mesh in a multistage process involving shape matching, iterative alignment to the target shape, and physics-based settling combined with heuristics adapted from traditional manual dry stone masonry. The geometric planner provides the desired position and orientation of the stones in the target structure and a placement direction corresponding to the least occlusion. The stones

used for construction are irregularly shaped stones in the size of 445-2425 kg (average 1030 kg). During the several-day construction process, stones were regularly dumped and spread in the excavator's close vicinity, where they could be grasped, scanned, and digitally reconstructed for assembly and grasp planning. The task of the grasp pose planning was to find a viable grasp pose that enabled picking the desired stone and placing it at the planned pose without colliding with the existing wall.

In the course of this large-scale experiment, a single wall was built: it required 145 stone placements in the wall, from which 76 object reorientations (52.4 %) were needed because no grasp pose could be found that fulfilled the collision constraints at the pick and place locations. The high number of required reorientations is because the assembly planner is presently not considering the current orientation of an object, such that it has maximum flexibility in finding viable solutions for the structure.

5.4.2 Grasp Success Evaluation

In order to perform the 145 placements in the wall, a total number of 443 grasp attempts were executed, with 364 grasps (82.2 %) being labeled as a success (meaning that the stone could be lifted and placed at a new location), whereas 79 slipped out of the gripper during the grasp attempt (see Table 5.1). To place a stone in the desired pose, usually, multiple grasps had to be performed. The 364 successful grasps are distributed as follows: In 145 cases, the stone was placed in the wall, 76 grasps were performed to place the stone in the designated flipping zone, and 77 grasps were used to reorient the object (see Table 5.2). In 66 cases, the stone was grasped successfully but placed outside of the wall structure (OoW). We indicate this scenario separately, as the grasp was successful (meaning that the stone could be lifted and relocated), but the placement could not be directly executed. This scenario could be due to potential collision at placement because the stone moved during the gripper closing or if the human observer considered the placement to lead to a high risk of a partial wall collapse. In this case, we placed the stone again on the ground to replan and adjust the grasp for subsequent placement. In Figure 5.9, a selection of the executed placement grasps at the desired stone location is shown.

Table 5.2: Distribution of grasp attempts.

# Succ. Grasps	Placements in Wall	To Flip. Zone	Reorient. Grasp	OoW Placement
364	145 (39.8 %)	76 (20.9 %)	77 (21.2 %)	66 (18.1 %)

Table 5.3: Distribution of grasp attempts that lead to placement.

Placements in Wall	direct	direct w/o slip	from Reorient.	Reorient. w/o slip
145	69	42 (60.9 %)	76	32 (42.1 %)

Table 5.3 compares the occurrence of slippage for stones that could be placed directly or had to be reoriented. We can see that since reorientation requires multiple grasps, the chance of slippage increases, and the share of placement sequences without slippage is lower (42.1 %) than in the case of direct placement (60.9 %). Through the possibility of updating the object pose in the world frame and regrasping when slippage appears, all 145 placement sequences could be carried out in the end with an average of 3.06 grasp attempts.

In Table 5.4, we report the average hierarchical filter cost terms. The average cost terms are slightly lower for successful placements than in the case of slippage, indicating that the cost terms point in the right direction. However, the high stan-

Table 5.4: Normalized costs (mean and standard deviation) of placement and slip grasp.

Cost	Placement Grasps	Slip
c_{align}	0.9979 ± 0.7646	1.3152 ± 1.0986
c_{enc}	1.0297 ± 0.3182	1.1705 ± 0.2860
c_{dist}	1.1546 ± 0.7966	1.3108 ± 0.7833

Figure 5.9: Examples of stone placements at the desired location. The placement grasps comply with the challenging collision constraints and enable to fit objects even in tight spaces.

Table 5.5: Execution duration from receiving desired placement pose to placement in wall.

	Bottom Third [s]	**Middle Third [s]**	**Top Third [s]**
No reorientation	554	765	826
Reorientation	940	1558	1332

dard deviation of the cost terms shows that they are not distinctive enough to predict grasp success ultimately. It is crucial that a grasp approach, as in our case, is including slippage detection and regrasping to achieve a high placement success rate. The reorientation grasps had significantly higher average alignment cost c_{align} (3.0347 ± 0.9603) as they are inherently tilted against the placement direction.

5.4.3 Construction Rate and Accuracy

The process of updating poses, in-hand refinement, and regrasping is time-consuming but improves the construction accuracy. We achieved an average positioning deviation of 0.128 m between the desired placement pose and the ICP update of objects in the structure that is comparably low to the object size. Note, however, that there is no absolute ground truth available, as the ICP update contains the errors from state estimation and laser mapping.

In Table 5.5, we present the average execution duration for pick-and-place an object. This duration covers the complete sequence from receiving the desired placement pose until the stone is put into the wall, including grasp planning, pose up-

date, in-hand refinement, trajectory tracking, and possibly reorientation or failure recovery. The average execution duration increases from 707 s for direct placement in the wall to 1308 s for objects required to reorient because multiple grasps have to be executed. Besides the total duration, we display in Table 5.5 the influence of the construction height on the assembly rate. Placements in the bottom third of the wall were generally executed faster as they were less occluded by adjacent stone and 'easier' to place, resulting in less failure recovery. Compared to the total execution time, the grasp planner is comparably fast. For every grasp, 2000 surface points were sampled with ten grasp orientations at each point. The planner took an average of 16.08 ± 2.26 s to run on an Intel Xeon E3-1505M octa-core 2.8 GHz CPU: 11.62 s for generating the grasp hypotheses, 4.04 s for classifying force closure, and 0.42 s to evaluate collision at pick location and cost computation. We achieved an assembly rate of approx. 3 m^2 of wall per day. For comparison, trained dry stone masons construct 1-2 m^2 of wall per day, but this number depends on the available stone size and type [108]. A human operator could increase the construction speed itself, as we operated the robot at reduced speed for safety reasons. Nevertheless, manually creating an assembly with irregular objects of this size is highly challenging and not practicable as a trial-and-error approach for finding form fits.

5.4.4 Slippage Reasons

Grasping irregular objects is challenging, as a slight deviation in pose or surface shape leads to varying grasp quality. The contact quality depends highly on local shape features like minor dents and protrusions, and these features have a significant impact on the grasp success. The accuracy of the contact points is mainly influenced by slight asymmetries of the grasp, which can cause the object to move during gripper closing, thus decreasing the success rate. Another common cause of slippage was that cone-shaped objects were pushed out of the gripper by the closing force.

5.5 Summary

We have presented an approach for grasping and reorienting large-scale stones with an automated excavator and evaluated its applicability in real-world experiments

by assembling a dry stone structure. The grasp planner is tailored to placing objects in involved assemblies with a 2-jaw gripper. It samples grasp configurations on a point cloud combining the LiDAR-based map and reconstructed object meshes and validates them for collision at the pick and place location. If no direct placement is feasible, we search for an intermediate placement pose by letting the reconstructed object settle in a physics simulation. We have evaluated the effectiveness of our approach in an extended experiment of placing over one hundred stones in a dry-stone wall. We show a high primary grasp success rate (82.2 %) and illustrate how the system recovers from slippage by relocating the object and re-planning the grasp correspondingly.

Part III

DISCRETE ASSEMBLIES

6

Vertical Stone Towers

This chapter incorporates material from the following publications:

Furrer*, F., Wermelinger*, M., Yoshida*, H., Gramazio, F., Kohler, M., Siegwart, R., & Hutter, M. *Autonomous Robotic Stone Stacking with Online next Best Object Target Pose Planning* In *IEEE international conference on robotics and automation (ICRA)*, pp. 2350-2356. IEEE, 2017.

Wermelinger*, M., Furrer*, F., Yoshida*, H., Gramazio, F., Kohler, M., Siegwart, R., & Hutter, M. *Greedy Stone Tower Creations with a Robotic Arm* In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, pp. 5394-5398. Lawrence Erlbaum Associates, 2018.

Video: https://youtu.be/bXz52KMGUng

We want to target discrete rigid elements, such as stones or concrete rubble, as a building material. A fundamental question is how to specify, plan, and execute placements of such elements to achieve the desired final target structure. Especially planning with irregularly shaped objects is a complex problem since both the state and action space are continuous, and structural stability is strongly affected by complex friction and local contact constraints. In this chapter, we investigate the possibility of using reconstructed object models in a physics simulation to assess the stability of a structure. We show that the compositions made out of multiple objects can be executed in the real world by a robotics system.

Our goal is to construct vertical balancing towers with found objects while maintaining the structure in static equilibrium using a manipulator, revealing the following challenges. Firstly, individual object instances need to be identified. Secondly, grasping and stacking poses are not obvious, requiring a novel algorithm to pick

* The authors contributed equally to this work. F.F. was responsible for the object detection, M.W. for the manipulation tasks, H.Y. for the pose searching algorithm.

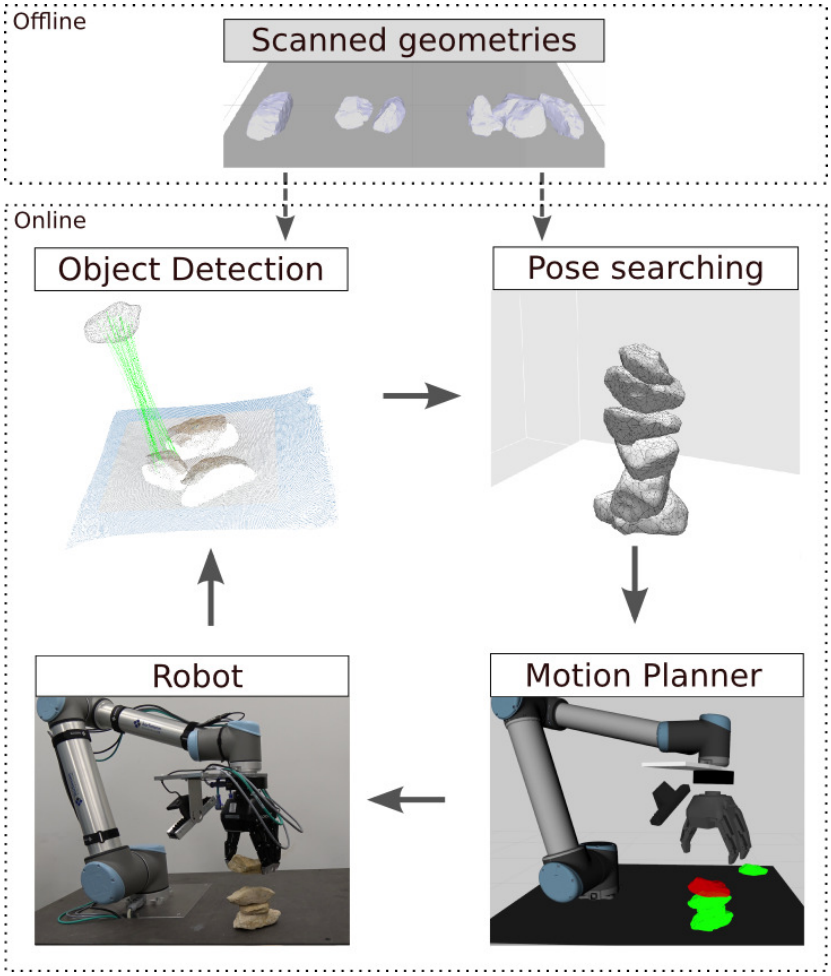


Figure 6.1: In an offline step we scan a set of objects (top). These objects, or a subset of it, can be distributed arbitrarily on the work-space and get detected by our object detection pipeline (middle-left). From the detected objects the presented pose searching algorithm proposes the next stable stack (middle-right). A motion planner (bottom-right) is used to generate the trajectories to replicate the proposed stack with the robot arm (bottom-left). After placing the object, its pose is measured and used as base for the subsequent pose searching step.

a 'good' next pose among infinitely many. Thirdly, the stacking task may be performed in delicate situations, requiring systematic structural evaluation and target re-planning after each object placement. To achieve this, we developed a holistic workflow including precise object detection, motion control, and planning the next target pose (see Figure 6.1). As part of this autonomous workflow, we describe an algorithm suggesting stable poses for stacking, validated in a real-world experiment. Due to the vertical tower's instability, it is natural to observe errors between the desired target pose and an actually stacked pose. Thus, the workflow emphasizes the resultant pose evaluation and a dynamic re-planning of the target pose.

6.1 Related Work

Computational structural analysis methods have been explored with rigid discrete elements, such as simple brick-like geometries. Livesley [109, 110] gives a numerical method for limit analysis of discrete rigid block structures. Block, Ciblac & Ochsendorf [111] employed graphical statics in interactive design tools with structural analysis feedback, and Whiting, Ochsendorf & Durand [112] extended the limit analysis by Livesley to design a guidance system by adding infeasibility metrics. These works analyzed a static equilibrium of a given geometric configuration with apparent geometric contact surfaces (or support polygons). In our case, with irregularly shaped elements, we need to start from contact detection, which is a core function of physics engines and then acquire a contact surface.

From an architectural design motivation, simulation with a physics engine has been explored by Nielsen & Dancu [113]. However, their construction input was to place the next stone at the lowest possible location. As for design with irregularly shaped objects, Lambert and Kennedy developed an application for guiding masonry construction with a geometric packing algorithm in two-dimensional convex shapes, but it is limited to two-dimensional tiling [114].

Humans synthesize the Center of Mass (CoM) position and the support polygon to infer the stability of objects [115]. On the other hand, physics engines, such as Open Dynamics Engine (ODE), have been used for evaluating the structural stability of object compositions [116]. Similarly, our algorithm employs a physics engine to extract the CoM position and dynamic characteristics, as well as the contact points between the irregularly shaped objects.

Integrated autonomous systems dealing with object detection and picking these objects with robotic arms have been proposed for industrial applications [117, 118]. These works have successfully detected irregularly shaped stone-like objects for geometric packing. However, they are limited to place these objects in containers, and hence, did not need to consider structural stability.

6.2 Object Detection

Before starting with the stacking algorithm, we need to find the objects in the scene. Additionally, during the course of the object stacking, we want to be able to track

the locations of the objects. In the scope of this work, we are only considering pre-scanned (the scanning method is described in Section 6.4) models of the objects to be detected in the scene. Therefore, we present an object detection pipeline that consists of the following steps. We start by *extracting 3D keypoints* from raw point clouds of an RGB-D sensor. These keypoints are then described using *keypoint descriptors* and matched to keypoints from a pre-scanned object in a *descriptor matching* step. Using these matches and a *clustering* step, we find an initial alignment of the scene and the pre-scanned object and *refine* it by applying an ICP algorithm. As a final step of the object detection pipeline, we *verify that we have enough inlier points* by applying the identified pose transform of the object to the scene.

6.2.1 Keypoint Extraction and Description

From an RGB-D sensor, we get a scene point cloud $\boldsymbol{P}_{\mathcal{C}}$ in camera frame $\mathcal{C}$. To get keypoints, we used two methods, a simple voxel-based subsampling and the PCL implementation of ISS [88], which can describe the keypoints and be used as a keypoint detector. Besides the ISS descriptor, we tested two additional descriptors, namely the FPFH descriptor [90] and the RoPS descriptors [91]. Here, the RoPS descriptors were giving us the best results in the matching step, at a slightly higher computational cost.

6.2.2 Descriptor Matching and Clustering

We compare a keypoint $\boldsymbol{k}_{\mathcal{C},\text{scene}}$ of a scene point cloud with a keypoint of a point cloud of a pre-scanned object $\boldsymbol{k}_{\mathcal{O},\text{object}}$ in object frame $\mathcal{O}$. To find a pair of corresponding keypoints $\boldsymbol{k}_{\mathcal{C},\text{scene}}$ and $\boldsymbol{k}_{\mathcal{O},\text{object}}$, we set up a kd-tree in descriptor space to find the nearest neighbors. Then we use an approach presented in [119] to verify that the matched keypoints are geometrical consistent. We select the b best transforms $\boldsymbol{T}_{\mathcal{CO},j,\text{matching}}$,
$j \in \{1,\dots,b\}$, that give the most geometrical consistent matches. The transforms $\boldsymbol{T}_{\mathcal{CO},j,\text{matching}}$ project the object point cloud $\boldsymbol{P}_{\mathcal{O},\text{object}}$ into the camera frame $\mathcal{C}$.

6.2.3 Transform Refinement and Verification

Using an ICP step, we refine these transforms $\boldsymbol{T}_{\mathcal{CO},j,\text{matching}}$ to align better the two complete filtered point clouds $\boldsymbol{P}_{\mathcal{C},\text{scene}}$ and $\boldsymbol{P}_{\mathcal{O},\text{object}}$. We denote these refined transforms by $\boldsymbol{T}_{\mathcal{CO},j,\text{refined}}$. In the last step, we check for the inlier ratios of the transformed point clouds and then select the one which has the highest ICP-score, given that we have an inlier ratio of the model points larger than a threshold value. In our application, we set this threshold value to 20 % resulting in the final transform $\boldsymbol{T}_{\mathcal{CO}}$.

6.2.4 Object in Robot Arm Frame

To transform the point cloud of the localized object $\boldsymbol{P}_{\mathcal{O},\text{object}}$ into the robot arm frame $\mathcal{R}$, we apply the previously detected best transform $\boldsymbol{T}_{\mathcal{CO}}$, a fixed pre-calibrated transform $\boldsymbol{T}_{\mathcal{TC}}$ from the camera frame $\mathcal{C}$ to the robot arm tooltip frame $\mathcal{T}$ [120], and the transform given by the robot state $\boldsymbol{T}_{\mathcal{RT}}$ between the robot arm frame $\mathcal{R}$ and the tooltip frame $\mathcal{T}$:

$$\boldsymbol{P}_{\mathcal{R},\text{object}} = \boldsymbol{T}_{\mathcal{RT}} \cdot \boldsymbol{T}_{\mathcal{TC}} \cdot \boldsymbol{T}_{\mathcal{CO}} \cdot \boldsymbol{P}_{\mathcal{O},\text{object}}. \tag{6.1}$$

6.3 Pose Searching

The global goal is to construct a vertical tower consisting of irregularly shaped objects from a subset $\mathcal{S}$ of available objects $o_i \in \mathcal{S} \subseteq \mathcal{O}$, where $\mathcal{O}$ denotes the complete set of given objects. Within the set $\mathcal{S}$, we want to find the best object and its target pose. The search space is twofold: discrete object space and continuous pose space. In order to find a stable pose on a vertical stack, our pose searching method places each object o_i on the top object of the existing stack in a dynamic simulation using a physics engine. For evaluating each object's 'goodness' with a particular pose $\boldsymbol{p}_i$, we introduce a cost function that maximizes the support polygon S_i's area A_i of the newly placed object o_i and minimizes other considerable parameters, such as kinetic energy. Throughout this process, several initial poses are tested with fixed initial positions but randomized orientations. We are sampling our initial orientations randomly to keep the problem viable in large problem sets, where a holistic pose sampling would become intractable. The returned cost value is inter-

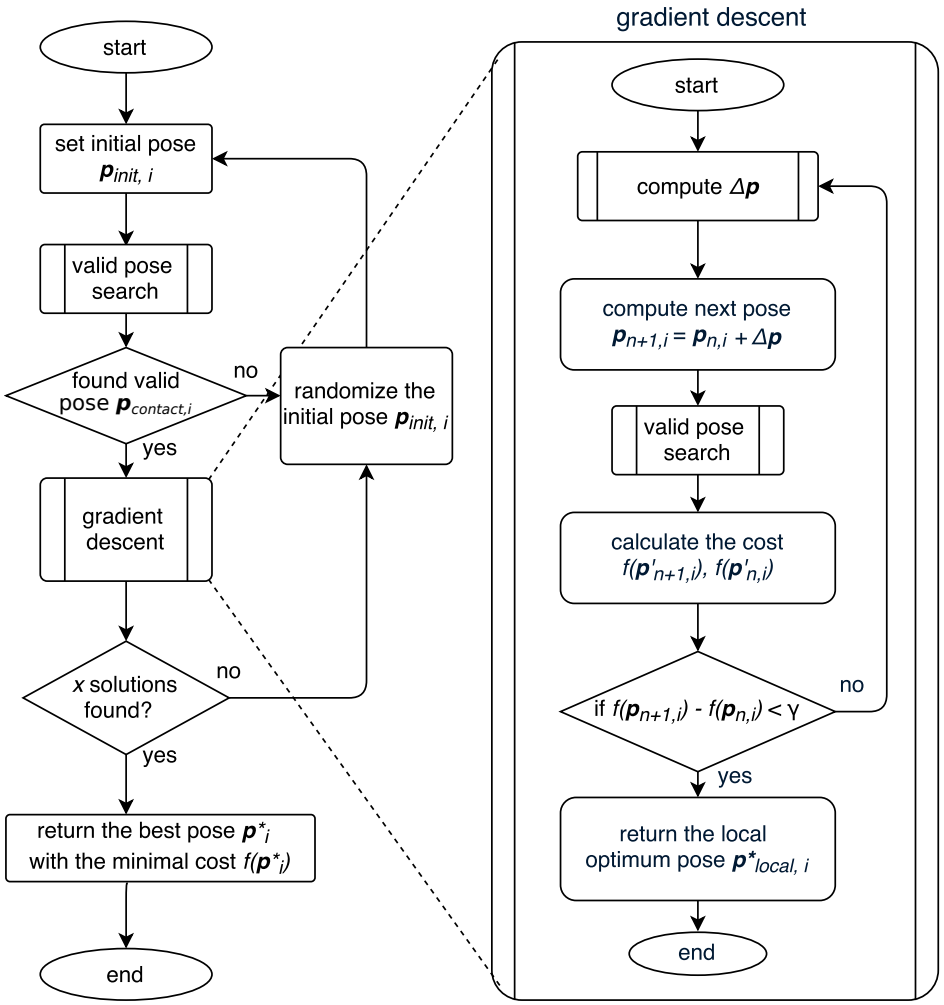


Figure 6.2: Illustration of our pose searching algorithm to find the best pose for object o_i. The valid pose search sub-routine is described in Algorithm 1 and the cost calculation in Algorithm 2. The gradient descent sub-routine is further depicted on the right. Note that we omitted the subscript $_{\text{local contact}}$ for clear presentation.

changeable among available objects in $\mathcal{S}$; thus, we find the best object o^* with the best pose $\boldsymbol{p}^*$. Figure 6.2 illustrates the complete pose searching workflow with its sub-routines.

6.3.1 Overview of the Algorithm

The pose searching algorithm iteratively evaluates poses with valid contacts (support polygon) between a newly placed object o_i and the existing stack. Once a pose with valid contacts is found, the algorithm refines the pose with gradient descent. To find a valid contact pose, we set the object to an initial pose in simulation close

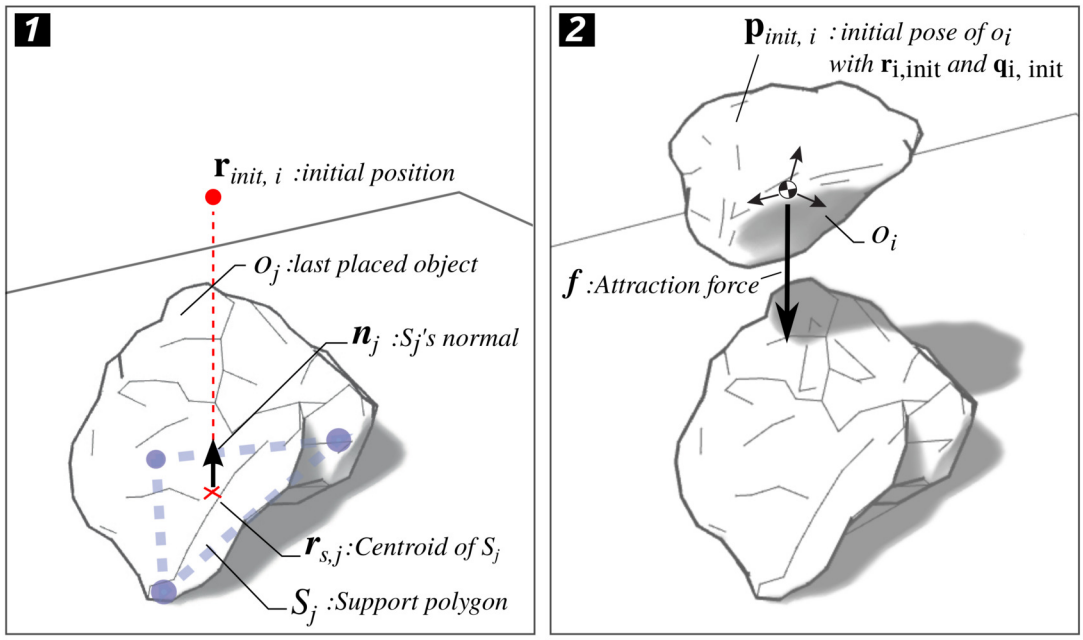


Figure 6.3: (1) The initial position $\boldsymbol{r}_{\text{init},i}$ of object o_i is set along the normal direction $\boldsymbol{n}_j$ of the previously placed object o_j's S_j. (2) The initial pose of object o_i with attraction force $\boldsymbol{f}$.

to the existing stack but not yet touching it. The initial pose $\boldsymbol{p}_{\text{init},i}$ of a new object $o_i \in \mathcal{O}$ consists of its initial position $\boldsymbol{r}_{\text{init},i}$ and its initial orientation $\bar{\boldsymbol{q}}_{\text{init},i}$. The initial position is set with an offset from the centroid $\boldsymbol{r}_{s,j}$ of the last-placed object's support polygon S_j along the normal direction $\boldsymbol{n}_j$ of S_j (see Figure 6.3, left). We obtain the initial orientation by rotating the detected object's orientation around a randomized axis with the random angle $\theta \in [-\theta_{\text{init}}, \theta_{\text{init}}]$.

Based on the initial pose $\boldsymbol{p}_{\text{init},i}$, the valid contact pose $\boldsymbol{p}_{\text{contact},i}$ is found through a sub-routine named *valid pose search* by applying an attraction force $\boldsymbol{f}$ parallel to a thrust line input (in our case, along the gravitational axis). We then run gradient descent (see Figure 6.2, right) to iteratively improve the contact pose $\boldsymbol{p}_{\text{contact},i}$ using the cost function presented in Section 6.3.3. After the local optimum pose $\boldsymbol{p}^*_{\text{local},i}$ is found, the orientation of the initial pose $\boldsymbol{p}_{\text{init},i}$ is randomized again to find the next valid contact pose $\boldsymbol{p}_{\text{contact},i}$. After computing a certain number n_{local} of local minima, we find the pose $\boldsymbol{p}^*_i$ with the lowest cost as the solution pose of o_i. We iterate the process over the available subset $\mathcal{S}$ to find the best object o^*.

The whole algorithm performs physics engine update steps only when necessary to avoid jittering effects in the physics simulation. Objects in an existing stack are set to be immobile in the whole process, except when the cost is calculated.

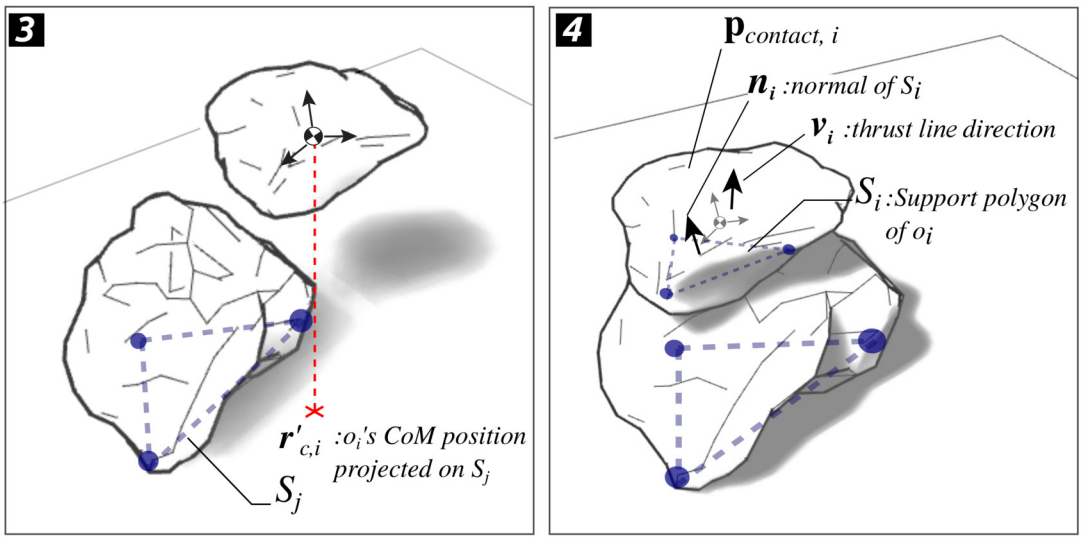


Figure 6.4: (3): Projection $\boldsymbol{r}'_{c,i}$ of the CoM position $\boldsymbol{r}_{c,i}$ onto the support polygon S_j. (4): Contact pose $\boldsymbol{p}_{\text{contact},i}$ resulting from the valid pose search algorithm.

6.3.2 Valid Pose Search

Algorithm 1 Valid pose search algorithm

function ValidPoseSearch($o_i, \boldsymbol{p}_{\text{init},i}$)
  set o_i's pose to $\boldsymbol{p}_{\text{init},i}$
  do
    apply force $\boldsymbol{f}$ to o_i
    step physics simulation once
    if $\boldsymbol{r}'_{c,i} \notin S_j$ **then return** false
  while $n_{\text{contact}}(\boldsymbol{p}_i) < 3$
  pause physics simulation
  $\boldsymbol{p}_{\text{contact},i} \leftarrow$ *current pose*
  if $E_{\text{kin}}(\boldsymbol{p}_{\text{contact},i}) < E_{\text{kin,stable}}$ **then return** true
  else return false

For evaluating the physical stability of object o_i, it is a valid approach to analyze whether $\boldsymbol{r}'_{c,i}$, the projection of the CoM position $\boldsymbol{r}_{c,i}$ onto the support polygon S_i, is inside the support polygon or not (see Figure 6.4, left). In order to find a valid support polygon S_i for an irregularly shaped object o_i, we consider the contact points of the object to other objects. The assumed contact situations are simple; either on a flat surface for the first stack, or collision between two rigid body objects with parallel contact normals.

We depict the details of the valid pose search method in Algorithm 1. To assure that $\boldsymbol{p}_{\text{init},i}$ results in a valid contact pose $\boldsymbol{p}_{\text{contact},i}$, we apply an attraction force $\boldsymbol{f}$ along the thrust line direction vector $\boldsymbol{\nu}_i$ to the object o_i (see Figure 6.3, right). During this process, we continuously check whether the projection $\boldsymbol{r}'_{c,i}$ of the CoM position lies within the support polygon. As soon as the number of contacts $n_{\text{contact}}(\boldsymbol{p}_i)$ between o_i and the existing stack is at least three (see Figure 6.4, right), the resulting pose $\boldsymbol{p}_{\text{contact},i}$ is evaluated by o_i's kinetic energy $E_{\text{kin}}(\boldsymbol{p}_{\text{contact},i})$ with a threshold value $E_{\text{kin,stable}}$. By evaluating the kinetic energy, we limit the viable set of poses to the ones that cause minimal motion of the existing stack. This approach for finding a valid contact pose $\boldsymbol{p}_{\text{contact},i}$ guarantees to satisfy the following constraints:

$$\begin{aligned} E_{\text{kin}}(\boldsymbol{p}_{\text{contact},i}) &\leq E_{\text{kin,stable}} \\ \boldsymbol{r}'_{c,i} &\in S_j \\ n_{\text{contact}}(\boldsymbol{p}_{\text{contact},i}) &\geq 3. \end{aligned} \tag{6.2}$$

6.3.3 Cost Calculation

We assign a cost to each valid contact pose $\boldsymbol{p}_{\text{contact},i}$ to compare its 'goodness' in terms of robust object poses, allowing further stacking. Therefore, we maximize the area of the support polygon S_i and minimize other considerable parameters, such as kinetic energy E_{kin}, and surface normal deviation from the thrust line $\boldsymbol{\nu}_i$ for reducing shear forces. To robustly find the support polygon S_i from the sparse contacts between o_i and the existing stack, contacts over several simulation update steps, in our case 10 steps, are collected and simplified [121]. After acquiring 3D point sets, Principal Component Analysis (PCA) reduces a 3D to a 2D point set. Processing the 2D point set with Delaunay triangulation [122], the polygon mesh S_i is created for calculating its area A_i and surface normal $\boldsymbol{n}_i$ (see Figure 6.4).

Given the area A_i of the support polygon S_i, the kinetic energy $E_{\text{kin}}(\boldsymbol{p}_{\text{contact},i})$, the dot product $\|\boldsymbol{n}_i \cdot \boldsymbol{\nu}_i\|$, where $\boldsymbol{\nu}_i$ is the thrust line direction vector, the length $\|\boldsymbol{r}_{c,ij}\|$ between $\boldsymbol{r}_{c,i}$ and the CoM of the previously stacked object $\boldsymbol{r}_{c,j}$, we define the cost function as

$$\begin{aligned} f(\boldsymbol{p}_{\text{contact},i}) = w_1 A_i^{-1} &+ w_2 E_{\text{kin}}(\boldsymbol{p}_{\text{contact},i}) \\ &+ w_3 \|\boldsymbol{r}_{c,ij}\| + w_4 \left(1 - \|\boldsymbol{n}_i \cdot \boldsymbol{\nu}_i\|\right), \\ \text{s.t. } w_j \geq 0 \; \forall j \in 1, \ldots, 4 & \end{aligned} \tag{6.3}$$

where w_j are manually selected weights of the individual energy function components. An overview of the cost calculation algorithm can be seen in Algorithm 2.

Algorithm 2 Cost calculation

```
function CalculateCost(o_i, p_contact,i)
    set all objects in stack mobile
    contactsArray[] ← 0
    for k ← 0, k < 10 do
        step physics simulation once
        contactsArray[] ← current contacts set
    contacts ← simplify(contactsArray[])
    contactPlane ← PCA(contacts)
    contacts ← projection(contacts,contactPlane)
    S_i ← 2D DelaunayTriangulation(contacts)
    get A_i(S_i), E_kin(p_contact,i), ||r_c,ij||, ||n_i · ν_i||
    reset poses of all objects in stack
    set all objects in stack immobile
    cost ← f(p_contact,i)
return cost
```

After assigning the cost to the valid contact pose $\boldsymbol{p}_{\text{contact},i}$, gradient descent is performed for searching the local optimum pose $\boldsymbol{p}^*_{\text{local},i}$, as depicted in the right of Figure 6.2. In this process, we iteratively calculate a small pose step $\Delta\boldsymbol{p}$ consisting of $\delta\hat{\boldsymbol{r}}$ and $\delta\hat{\tilde{\boldsymbol{q}}}$, as given in Eq. (6.5) and Eq. (6.6), to obtain an updated initial pose for valid contact pose searching:

$$\begin{aligned} \boldsymbol{p}_{\text{local contact},i}[n+1] &= \boldsymbol{p}_{\text{local contact},i}[n] + \Delta\boldsymbol{p} \\ \text{s.t. } \Delta\boldsymbol{p} &= (\delta\hat{\boldsymbol{r}}, \delta\hat{\tilde{\boldsymbol{q}}}) \end{aligned}, \tag{6.4}$$

where $\boldsymbol{p}_{\text{local contact},i}[0] = \boldsymbol{p}_{\text{contact},i}$. The translation $\delta\hat{\boldsymbol{r}}$ assigns a positive constant offset z_{const} in contact normal direction to avoid placing the object o_i at invalid poses overlapping with the existing stack. Given ϵ_r as a small value, $\boldsymbol{T}_i$ as a homogeneous transformation matrix from world frame to the thrust line aligned frame at $\boldsymbol{r}_{s,j}$ (see Figure 6.3), and z_{const}, we obtain $\delta\hat{\boldsymbol{r}}$ as:

$$\begin{aligned} \delta\boldsymbol{r} &= \boldsymbol{T}_i\left(\frac{\partial f}{\partial r_x}^{-1}, \frac{\partial f}{\partial r_y}^{-1}, z_{\text{const}}\right), \\ \text{and } \delta\hat{\boldsymbol{r}} &= \epsilon_r \frac{\delta\boldsymbol{r}}{\|\delta\boldsymbol{r}\|}, \\ \text{s.t. } z_{\text{const}} &\geq 0. \end{aligned} \tag{6.5}$$

For rotation, we describe the orientation with a quaternion $\bar{\boldsymbol{q}}$ in axis-angle representation as (axis, angle). Given ϵ_q as small rotation angle, and $\boldsymbol{T}_i$, we obtain $\delta\hat{\bar{\boldsymbol{q}}}$ as:

$$\begin{aligned} \boldsymbol{v}_{\text{axis}} &= \boldsymbol{T}_i\left(\frac{\partial f}{\partial q_x}^{-1}, \frac{\partial f}{\partial q_y}^{-1}, \frac{\partial f}{\partial q_z}^{-1}\right) \\ \text{and } \delta\hat{\bar{\boldsymbol{q}}} &= (\boldsymbol{v}_{\text{axis}}, \epsilon_q). \end{aligned} \tag{6.6}$$

For minimization of the cost, we iterate the process described in Eq. (6.4) until the difference of sequentially returned costs becomes smaller than the threshold γ:

$$f(\boldsymbol{p}_{\text{local contact},i}[n]) - f(\boldsymbol{p}_{\text{local contact},i}[n+1]) < \gamma. \tag{6.7}$$

We write the optimization process as,

$$\begin{aligned} \boldsymbol{p}^*_{\text{local},i} &= \underset{\boldsymbol{p}_{\text{local contact},i}}{\text{argmin}}\ f(\boldsymbol{p}_{\text{local contact},i}) \\ \text{s.t. } A_i(S_i) &\geq A_{\text{min}}, \end{aligned} \tag{6.8}$$

where A_{min} is the minimal support polygon area.

After finding a local optimum pose $\boldsymbol{p}^*_{\text{local},i}$, a new randomized rotation is assigned to the initial pose $\boldsymbol{p}_{\text{init},i}$. The process is repeated until n_{local} local solutions are found, as shown in the left of Figure 6.2, and the pose with minimum cost is selected as a solution $\boldsymbol{p}^*_i$ for object o_i. We iterate the entire process over every object of the available subset $\mathcal{S}$, returning the best object o^* with the best pose $\boldsymbol{p}^*$.

6.4 Experimental Setup

To show the applicability and repeatability of the presented pose searching and object detection methods, we implemented the algorithms, using ROS [25], on a robotic platform to perform autonomous dry-stacking, as in Figure 6.1. The goal is to create a vertical tower out of randomly placed irregularly shaped objects whose mechanical and geometric properties are known.

Figure 6.5: Limestones with irregular shape are used to create the vertical stacks.

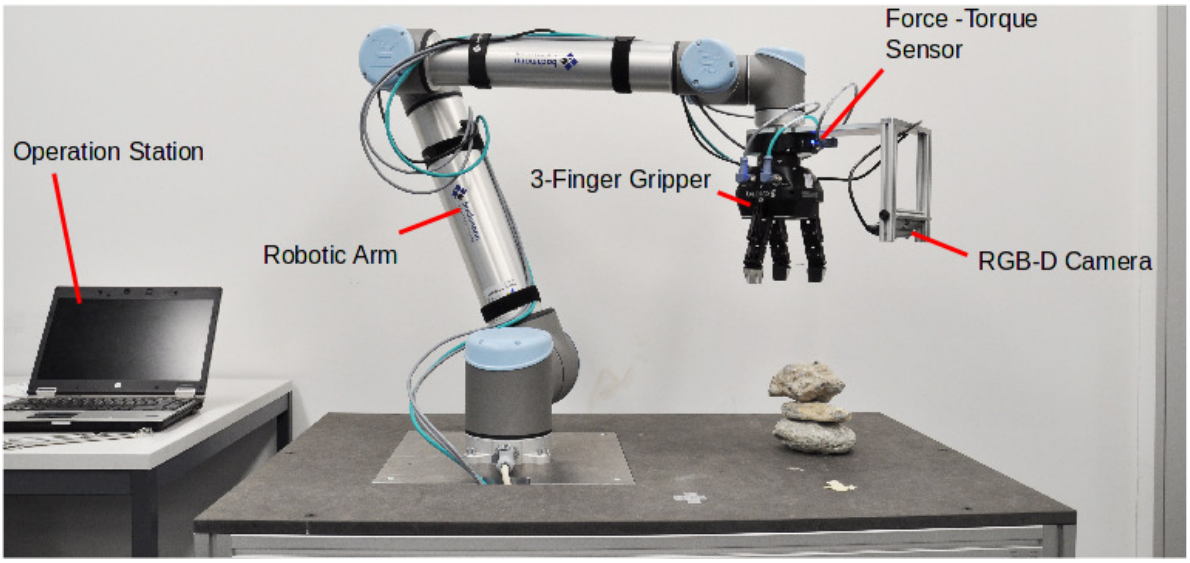


Figure 6.6: An overview of the used hardware setup: a ROBOTIQ 3-finger gripper, a FT150 force torque sensor, and an Intel®RealSense™ SR300 RGB-D camera are attached to a UR10 arm.

6.4.1 Experimental Setup

We use a set of six natural limestones (Figure 6.5) as objects because they show challenging properties for the stacking task like irregular shape and low friction coefficient. The point cloud and mesh model of the object's geometric shape were previously acquired with an ATOS Core high precision scanner. These models are used to detect the objects in the scene and to simulate them in a physics engine used in the pose searching algorithm. To lower the computation cost, we reduced the mesh model for the pose searching algorithm to a homogeneously triangulated mesh with 500 faces. The reduced meshes showed a reasonable trade-off between reducing the computation cost and generating a contact situation that correlates with the real world. The weight, CoM position, and moment of inertia of each stone were measured and added to the geometric model description. The friction coefficient was estimated with a low value of $\mu_{\text{stone}} = 0.1$. We use a robotic arm equipped with a three-finger gripper for manipulating the objects, as depicted in Figure 6.6. The object detection is performed with an RGB-D depth camera mounted on the robot arm. The size of the objects is selected to fit in the finger stroke of the gripper. MoveIt! [123] is used to generate collision-free motions of the robot. A force-torque sensor mounted at the attaching point of the gripper is used to detect impact during the placing of the object.

The workflow of autonomously creating a vertical stack of arbitrarily placed stones is shown in Figure 6.1. This task is performed by continuously executing a loop consisting of object detection, pose searching, and object manipulation. First, the objects are detected and localized in the scene, resulting in a set of available stones $\mathcal{S}$. For each trial, we used an alternating subset of four stones from the complete set of the six limestones. From this set $\mathcal{S}$, the presented pose searching algorithm proposes the next stable stack. In order to replicate the proposed stack, a collision-free grasping configuration from a predefined set of feasible configurations is chosen, and a motion planner generates executable trajectories. After placing the stone at the proposed pose, we detect the updated pose and validate if the stacking was successful. The updated stone pose is used as a foundation for the next pose searching step. If the robot could successfully execute the proposed stack, the next proposed stable stack is computed from the remaining set of stones. The stacking task is terminated once the pose searching finds no longer a feasible solution from the available objects or the stacking was not successful.

Table 6.1: Parameters of the pose searching algorithm.

Parameter	Value	Parameter	Value
w_1	0.179	A_{min}	$1e^{-5}\,\text{m}^2$
w_2	0.472	$\|\|\boldsymbol{f}\|\|$	100 N
w_3	0.094	$E_{\text{kin,stable}}$	20 J
w_4	0.255	θ_{init}	$\frac{\pi}{4}$ rad
n_{local}	5		

Table 6.2: Mean execution times of the vertical stone tower trials.

Task	Mean time (s)	σ time (s)	Fraction (%)
Pose search	66.2	7.0	24.4
Object detection	17.0	4.4	6.3
Manipulation	166.7	17.2	61.5
Other tasks	21.1	4.1	8.8

6.4.2 Results

The robotic system performed the vertical stacking task in eleven consecutive runs with an alternating set of four stones[1]. In two of these runs, the system succeeded in constructing a stack out of all four available stones. In six cases, the system was able to stack three stones vertically but failed to place the fourth stone, and in three cases, the system did not succeed in stacking the third stone. For the pose searching algorithm presented in Section 6.3, we used the parameters given in Table 6.1. The average cost of the last pose the system was able to stack successfully was 0.7425. Whereas, in the case where the robot failed to place the object, the average cost was 1.8018, which shows that these poses were already identified as less favorable than those that were successfully stacked. If the cost at a previous step is high, the probability increases that the following stone placement will fail. See, for example, the high cost of the second stone in runs 6 and 7 (see Figure 6.7). Table 6.2 shows

1 Watch the accompanying video: https://youtu.be/bXz52KMGUng

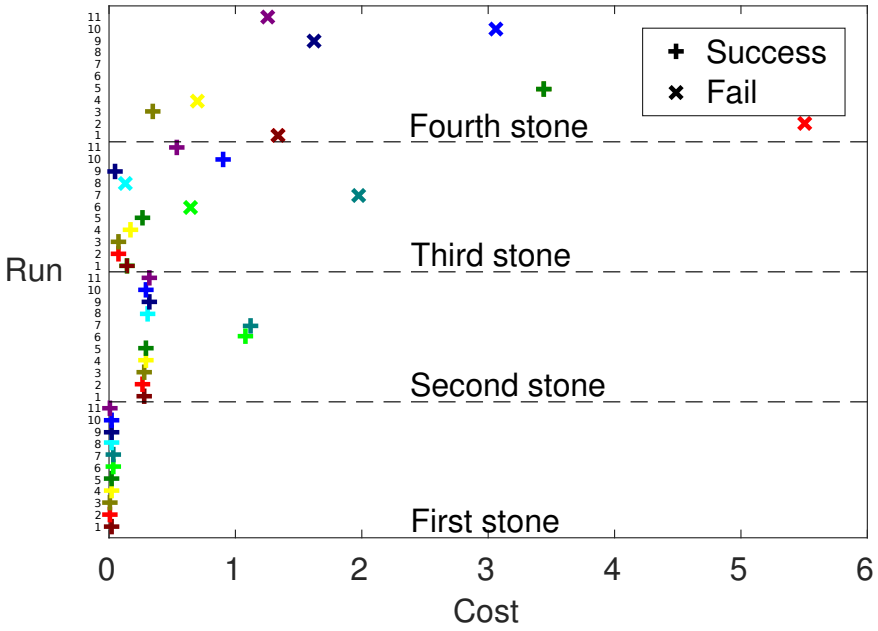


Figure 6.7: The cost of the selected target pose of an individual stone for all eleven runs at each level of the stack. A higher cost indicates a less preferable target pose. '+' denotes a successful stacking, failed attempts are represented with a 'x'. Each color corresponds to an individual run.

the mean execution times of the different parts of the presented approach, although computational and execution speed was not the focus of this work. On average, a trial to construct a vertical stack lasted 271.0 s. The main fraction of this time is spent on the manipulation task that includes path planning, arm and gripper motion since we operate the robot with reduced speed for safety reasons. The remaining time is spent on the pose searching algorithm itself, object detection, and other tasks, such as visualization and stopping the motion before capturing depth images. The stacking workflow is not yet optimized in terms of construction time. It could be greatly improved by parallelizing manipulation with pose searching and object detection, and by increasing the operation speed.

6.5 Summary

This chapter introduced an autonomous robotic system that constructs a vertical balancing tower out of irregularly shaped stones without using mortars or extra materials. Its workflow consists of a continuous loop with object detection, target

pose search, physical manipulation, and evaluation. We presented an object detection pipeline suited to localize irregularly shaped objects in a scene and a target pose searching algorithm to generate stable stacks. The proposed algorithms were implemented on a robotic system and tested in an experimental setup: a fixed platform in a controlled environment with flat terrain and pre-scanned objects. The system showed to be able to perform stacking tasks autonomously, contributing to a preliminary setup for detecting irregularly shaped objects and validating the usage of a physics engine in the proposed pose searching algorithm.

7

Autonomous Dry Stone

This chapter incorporates material from the following publication:

Johns, R. L., Wermelinger, M., Mascaro, R., Jud, D., Gramazio, F., Kohler, M., Chli, M. & Hutter, M. Autonomous Dry Stone: On-Site Planning and Assembly of Stone Walls with a Robotic Excavator. *Construction Robotics* **4** 3, pp.127-140 (2020).

On-site robotic construction has the potential to enable architectural assemblies that exceed the size and complexity practical with laboratory-based prefabrication methods and offers the opportunity to leverage context-specific, locally-sourced materials that are inexpensive, abundant, and low in embodied energy. This chapter presents an integrated system for constructing double-faced dry stone walls in situ using the customized autonomous mobile hydraulic excavator HEAP. We outline a process for mapping the environment and localizing and digitizing irregular stones. Using this digitized information, we developed a method to determine the position and orientation of stones to align with a designer-indicated goal surface and conduct grasp- and motion-planning for collision-free placement. We address one of the main limitations of the target pose search presented in Section 6.3 that only allows contacts between two individual objects. The planner presented here can generate interlocking between multiple objects and enables the construction of extended structures like stone walls. As the properties of the materials are unknown at the beginning of construction, and because error propagation can hinder the efficacy of pre-planned assemblies with non-uniform components, the structure is planned on the fly. The desired position of each stone is immediately computed before it is placed, and we account for any settling or unexpected deviations. We demonstrate the applicability of our method by constructing a 3 m tall wall with a total length of 5 m out of 40 stones with masses between 230-1548 kg (see Figure 7.1).

Figure 7.1: The stone wall constructed with the autonomous hydraulic excavator HEAP consists of 40 stones with masses between 230-1548 kg.

7.1 Related Work

Several recent projects in the domain of architectural digital fabrication have made use of nonstandard input materials combined with robotic construction. Such projects have played a substantial role in re-establishing computational design and fabrication as a practice that can not only materialize digital complexity but that can understand and adapt to the existing complexity of the material world. Johns & Anderson [13] provide an overview of recent projects in this general domain, while a number of notable projects deal more specifically with irregular stone-based assembly: "Smart Scrap" uses automotive digitization tools to inventory partially-dressed limestone for facade planning [14], while "Cyclopean Cannibalism" revives ancient polygonal masonry methods through robotic stone-cutting [15]. The latter fits parallel-faced polygonal elements within the constraints of demolition concrete

and stone offcuts, using 73% of the stock material (selected from a more extensive set) after dressing. In comparison, the work presented in this chapter makes use of 100 % of each stone object and uses every stone in the available set.

Towards the assembly of such raw materials, the geometric nesting of unmodified parts has been explored in two dimensions with simulated annealing [114]. Others have demonstrated the potential of shape descriptors for matching 3D shapes to existing geometries for the generation of 3D collages without constraints of fabricability [124] or the reassembly of fractured objects without structural constraints [125].

Similar to our demonstrator presented in Chapter 6, Liu, Choi & Napp [68] perform dry stacking of unmodified stones, using stationary robotic manipulators and RGB-D scanners for the localization of pre-scanned stones. Both projects consider structural stability using a physics simulator and evaluate a cost function to determine the validity of each potential pose. Our work in Chapter 6 is focused on constructing vertical stacks, while the latter can build one-layer thick walls of up to four courses with a 40 % success rate in an open-loop fashion (without accounting for the settling of placed stones). In this work, we overcome the design limitations of such 1- and 2- dimensional structures, presenting a planner capable of building a multiple-layer-thick structure following an arbitrarily specified target geometry. Stones are in-situ scanned on the fly, and the as-built wall is monitored in order to refine the placed-stone poses before the planner is executed again.

7.2 Methodology

7.2.1 Platform

The stone wall was created using HEAP, a highly customized Menzi Muck M545 12t walking excavator developed for autonomous applications and advanced teleoperation (Figure 7.2J). Further details about the platform can be found in Section 4.2. The scale and payload of this system, compared to a human and a conventional industrial manipulator, are displayed in Figure 7.2K.

All software components of the project are written in C++, and the Robot Operating System (ROS) [25] is used to transfer data over the network between the different software nodes distributed on several computers. The ROS master, man-

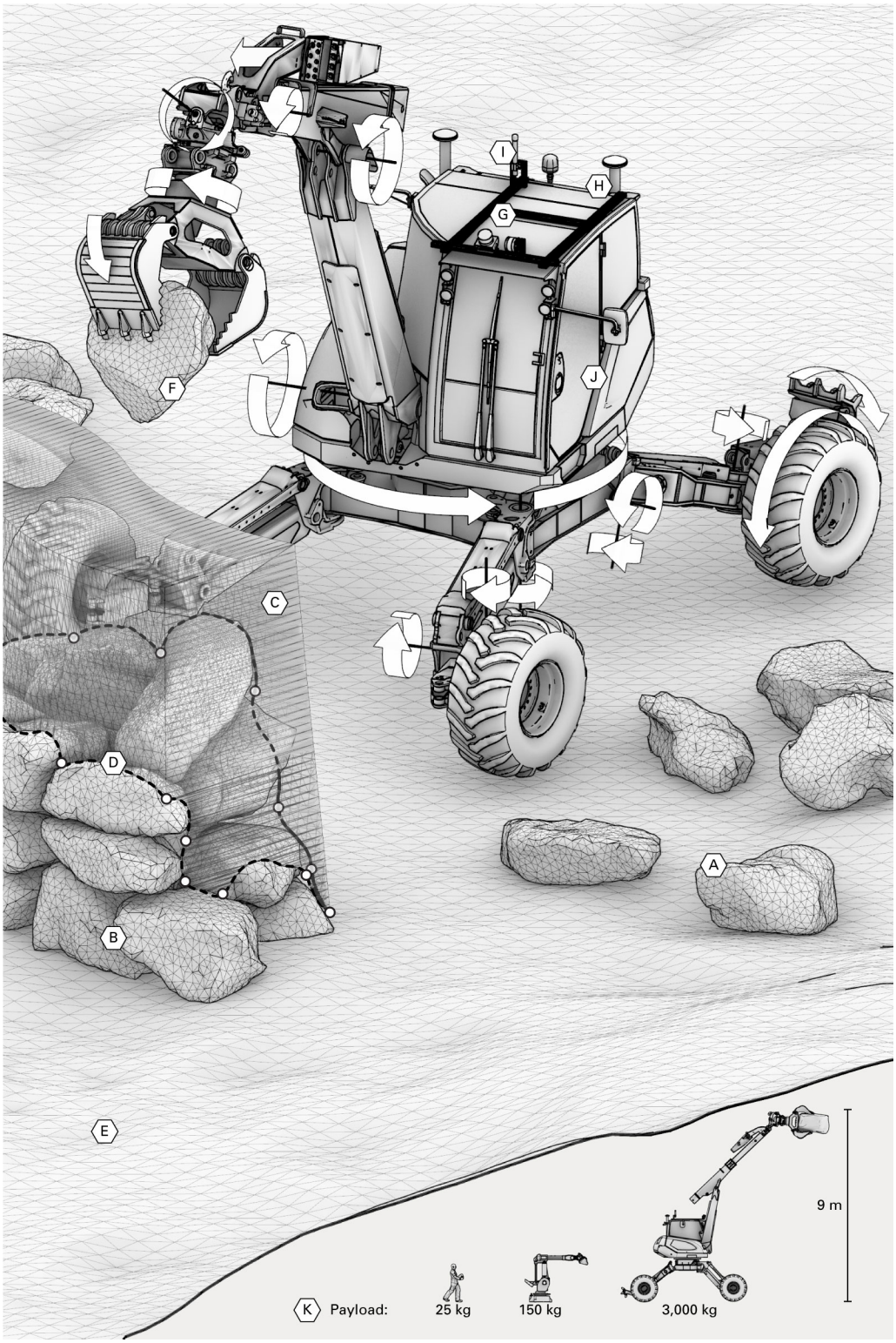


Figure 7.2: Setup Diagram:
A) Available stones B) Constructed wall
C) Search space D) Rim path with keypoints
E) Grid map F) Gripped stone
G) LiDAR sensors H) GNSS antennas
I) Network antenna J) HEAP platform with axes indicated
K) Reference scale and payload comparison with human and industrial robot.

aging the connection between processes, is located on the onboard computer of HEAP.

7.2.2 Stone Localization and Scanning

To reconstruct and localize the stones in the surroundings of the excavator, we use the LiDARs mounted on the cabin roof (Figure 7.2G). Before the construction process begins, the available stones are loosely distributed on the terrain (Figure 7.3). The spreading facilitates detecting single stone instances by first segmenting the ground plane from the LiDAR map and subsequently performing euclidean clustering on the remaining point cloud to distinguish the single objects. These point clusters are used to compute an initial grasp configuration on the not yet reconstructed stone segments (see Chapter 4). Once a given stone is picked up using this relatively low-resolution data, a high-resolution scan of the entire stone is completed while held in the gripper. The following sections describe this stone reconstruction process, the LiDAR-based scene mapping, and the object pose refinement based on the reconstruction and scene map information.

Reconstruction

For the 3D reconstruction, the end effector spins with continuous velocity in front of the LiDAR sensors while holding the stone. With the known geometry of the gripper, we filter out points that belong to the end-effector while accumulating points on the stone. The complete contour of the stone is recorded by concatenating multiple point clouds from different viewing angles and applying Poisson surface reconstruction [126] to generate the surface mesh from the point cloud data.

LiDAR-based Scene Mapping

A precise map of the excavator's whole surroundings (including the as-built structure) is necessary for planning grasp configurations and arm motions, and facilitating the decision of where and how to position stones (see Figure 7.7A). We use the mapping pipeline described in Section 4.3.2 that provides a temporally and spatially consistent map without self-see points on the excavator.

Figure 7.3: Initial state of stones, loosely distributed on the terrain. LiDAR mapping and RANSAC ground segmentation allows for initial detection and grasping for refined scanning in the gripper.

ICP Refinement

Being able to refine the pose of an already reconstructed stone in the environment (Figure 7.7B) or the gripper is an essential capability, as the stone might settle during placement or shift as it is being picked up. For this pose refinement, we first sample a point cloud on the reconstructed mesh of the stone. Using an ICP step, this point cloud is registered to the scene map (if the stone is placed in the environment) or to the accumulated gripper cloud (if the excavator grasps it). To improve the localization, we first segment the ground plane in the vicinity of the stone using RANSAC-based 3D plane fitting. However, additional filtering is necessary as stones might be cluttered and partially occluded, especially if they are constituents of the in-progress wall. Therefore, the scene map is further segmented using Euclidean Clustering, and segments belonging to already refined stones or segments

too far from the assumed pose are removed before the ICP step. The filtering allows for the reliable pose refinement of objects even if the built structure partially occludes them. If the initial orientation of the stone is unknown (e.g., after it has been overturned), the ICP step is performed for multiple initial orientations, and the refined transformation with the highest ICP-score is selected.

7.2.3 Geometric Planning

The iterative selection, positioning and orienting of each stone is determined by a geometric planning software that attempts to construct a double-faced wall, bounded vertically by any user-specified mesh target surface and below by the LiDAR-scanned elevation map of the existing terrain. Given any number of available stones, the software determines the preferred placement from the available solutions. The selected stone and the desired transformation are further processed for grasp planning and physical positioning. Rather than pre-planning the entire wall, the software computes solutions on the fly, stone-by-stone. The incremental planning is done for two reasons:

First, it is impractical to pre-scan, sort, and store hundreds or thousands of stones for sufficiently large constructions when the required space and complexity can instead be minimized if they are salvaged, unearthed, or delivered on-demand, as needed. Computing solutions on the fly allows for a more adaptive construction that is not held up if one pre-planned stone goes missing.

Second, the absolute rigidity of stone makes for significant error propagation that would result in undesirable deviation from the desired geometry and a higher probability of structural collapse in turn. Any minute difference in the scanned geometry can cause unexpected collisions or settle during placement, and the current setup allows for these changes to be automatically accounted for in the next solution.

Finding a suitable position of even one stone in a relatively small area can be a computationally heavy problem. The irregular nature of the stones and substrate geometry makes it generally difficult to apply stochastic solvers or continuous optimization methods as there are many local optima The software should generally find a valid solution in less time than it takes for the excavator to physically locate, grasp, and place a stone, such that the finding of solutions does not significantly deter progress. At present, the solver takes approximately one minute per stone on

a laptop running Ubuntu 18.04 with an Intel i7-8750H 4.1 GHz Processor and 16GB RAM.

Much like the algorithm presented by Liu, Choi & Napp [68], we use a number of heuristics derived from conventional stone masonry techniques to reduce the size of the solution space. However, our approach differs significantly in the specification and sequencing of these heuristics: we use both the geometric properties of the stones and the constraints of the designer-specified goal surface to inform an algorithm that begins with fast computational checks and increases in complexity as the solution space is substantially reduced.

For any given stone placement, the following high-level goals are established: determine a structurally stable solution that minimizes the volume directly under the stone in question (Figure 7.4A) and also minimizes the volume between the exposed stone face and the designated geometry of the wall surface (Figure 7.4B). To achieve these objectives, we implement the following subroutines:

Stone Attributes

At all times, the software holds an inventory of stones that have been scanned. Upon receipt of new information from the scanning node, the stone mesh is immediately cleaned by removing any degenerate faces and disjoint remnants that might remain from the Poisson reconstruction. The mass properties are computed using the method described by Bächer *et al.* [127], and each stone is repositioned such that its centroid is at the origin, and its principal axes (computed using PCA on the mesh vertices) are aligned with the local stone frame. An iterative implementation of the Variational Shape Approximation method [128] is used to determine approximate partitioned face surfaces on the stones given an allowable surface normal deviation per clustered region of mesh triangles. From these partitioned regions, a face-adjacency matrix is generated, stone edges and edge midpoints are identified. The mean weighted normal of each region represents the normal direction of the partitioned face surface (Figure 7.4C).

Search Space

A volumetric search space (Figure 7.2C) is defined as the region that is bounded by the designated outer wall surfaces on all sides and above, and below by the ground and the upper surface of the in-progress wall. While this upper surface

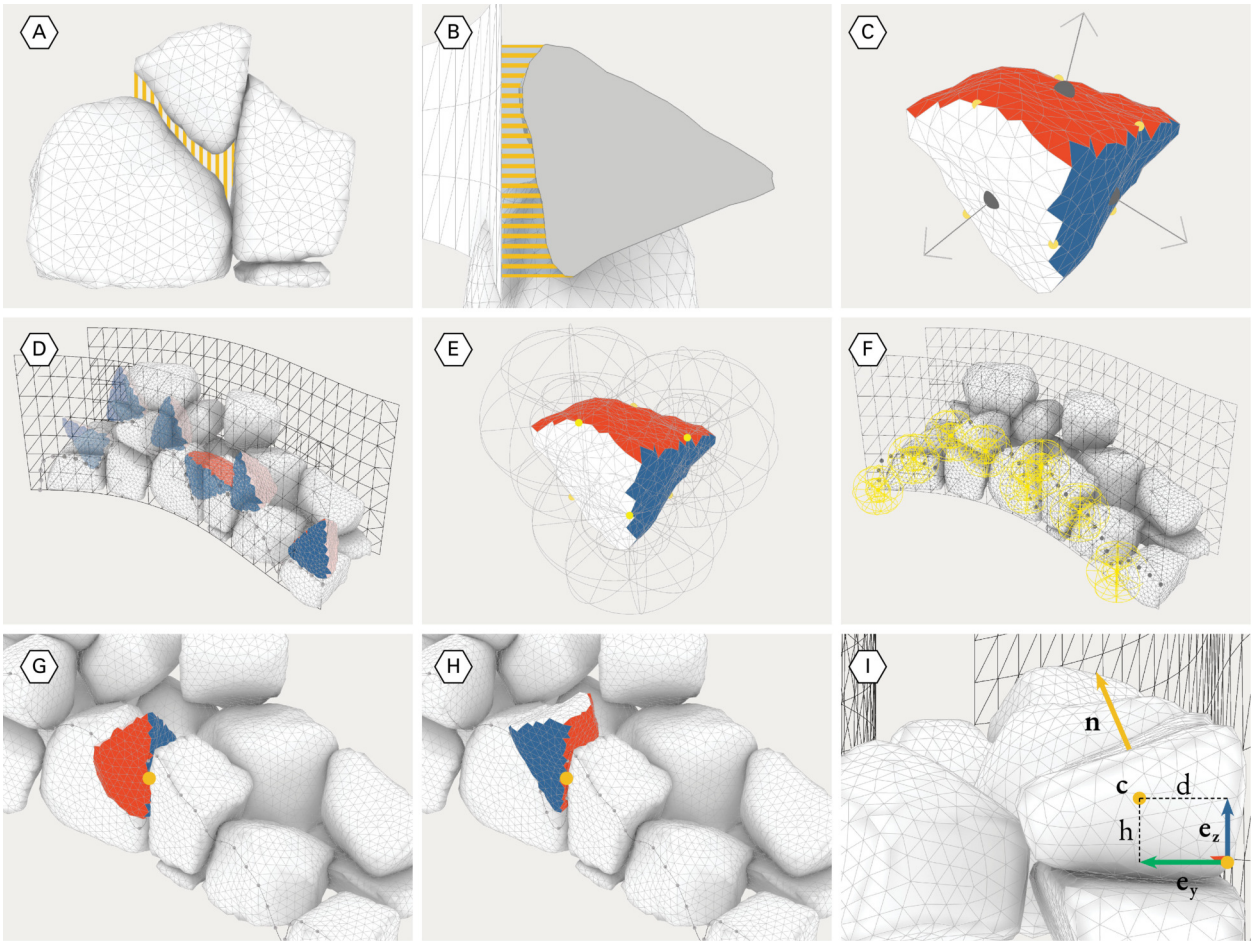


Figure 7.4: Geometric planning solver goals and methods: A) Minimize volume under stone, B) Minimize volume between stone face and goal surface, C) Compute faces and edges, D) Sample along rim path, E) Compute descriptors at edge midpoints F) Descriptors at rim path points. G-H) Two possible orientations for each partial match I) Stability and upper normal direction factors.

is, in reality, generally composed of many disjointed stones, it is approximated as a single continuous surface using a simulated LiDAR scanner on the known mesh data of the ground and placed stones. The point cloud from the simulated LiDAR is then meshed using Poisson reconstruction. Thus, both the empty ground and the upper bounds of a half-built wall are treated similarly to the next stone's foundation. The intersection between this reconstructed "ground" surface and the target wall-face surfaces defines the rim path, which we identify as the upper perimeter edge of the structure at any given state (the contour where the top surface of the as-built structure intersects with the vertical goal wall-surfaces)(Figure 7.2D).

An even sampling along the rim path serves as a one-dimensional list of possible placement positions. This list is further reduced to only include lower points within

a given distance of identified keypoints (Figure 7.2D) along this path. We define *keypoints* as any point along the rim path that lies either between two stones, at the corners or ends of the designated structure, or where one stone meets the ground. This filtering process ensures that the wall is generally built up evenly and reduces the likelihood of running joints that weaken the bonding of the wall.

Fit Estimation

Given the known edge-midpoints, edge directions, and face-normal-vectors on the stones, and the normal vectors of the goal-wall surface at each point on the rim path, it is trivial to determine the transformations necessary to place any specified stone on this path – such that a given stone edge is tangent to the rim path and the stone face is aligned with the wall surface. While each combination of path-point and stone-edge could be sampled (Figure 7.4D), this still represents a substantially large solution space if the wall is long and there are multiple known stones in the available inventory. Just as a stone-mason would likely only try placing stones that look like they will fit (instead of brute force sampling every possible position), we first check if a stone is geometrically similar to the region in question using rotationally invariant shape descriptors before attempting any more involved processing (Figures 7.4E-F, 7.5). Specifically, we use the FPFH descriptor [90] at each edge midpoint and at each rim path vertex coupled with a k-Nearest Neighbor (KNN) search to find the k-closest geometric matches at each included point on the rim path (k≈10). In this demonstrator, the KNN search typically reduced over a million possible combinations of stone edges with rim path points to some thousands. Because of the rotationally invariant nature of such shape descriptors, each match must consider both possible orientations of an edge with the rim path (Figures 7.4G, 7.4H).

Stability Heuristics

Prior to running any detailed physics simulations, we remove potential candidates that are likely unstable or likely to reduce future stability, informed by guidance from stone masonry literature. In dry stone walls, it is generally advised not to place stones in such an orientation that their thinnest dimension defines the depth into the wall—thus acting as a thin veneer that is more likely to fall away from the wall [129]. We generalize this requirement as needing to meet a minimum

Figure 7.5: Elevation detail of constructed wall. Rotationally invariant shape descriptors allow for a tight fit by initially matching available stone surfaces to corresponding negative geometry of the wall.

ratio between the horizontal (d) and vertical (h) distance measured between the specified point on the rim path and the stone centroid (Figure 7.4I):

$$d/h > 0.5. \tag{7.1}$$

As the construction process uses a double-faced method, the structure benefits from having an inward slope at the top of each placed stone – such that any settling

is minimized and supported by stones on the opposing side of the wall [1].[1] For each intended position, we take an even sampling of the exposed stone surface using a raycasting method and compute the mean top-exposed surface normal $(\boldsymbol{n}_{\text{top}})$ from the stone's normal direction at each hit point. We then verify that $\boldsymbol{n}_{\text{top}}$ is pointing into the wall and that the stone surface is not excessively steep by

$$\boldsymbol{e}_y \cdot \boldsymbol{n}_{\text{top}} >= 0 \quad and \quad \boldsymbol{e}_z \cdot \boldsymbol{n}_{\text{top}} > 0.7, \tag{7.2}$$

where $\boldsymbol{e}_y$ is the unitized inward-facing surface normal of the target wall surface projected onto the horizontal plane, and $\boldsymbol{e}_z$ is a unit vector pointing against the direction of gravity (Figure 7.4I).

ICP Refinement

Once the pool of potential matches has been reduced by the heuristics mentioned above, each candidate is refined using ICP to better align it with the surrounding stones and goal wall surface.

Physics

Following ICP refinement, the remaining solutions are sorted by their ICP-score and (if specified) reduced to a preset maximum number of solutions. These are then simulated for stability with the Bullet physics engine [130]. Solutions that do not reach equilibrium within a fixed simulated-world-time period or within a distance moved are ruled out.

Scoring

After the physics simulation step, the remaining stones are sorted using the combined parameters from Figures 7.4A and 7.4B. The gap volume is calculated with a raycasting technique measuring the displacement of hits measured from the local plan- and elevation- projection planes with and without the placed stone (and considering the known stone volume). The best match is then sent for physical

1 Note that this inward slope differs from the inward slope employed by Liu, Choi & Napp [68]: in our use and the masonry literature, the inward slope is a sectional property of a double-faced wall. Liu et al. use it to mean a lowered center of mass between the end stones of a given bond, as observed across the front elevation, thereby creating a sagging bond in a single-layered structure.

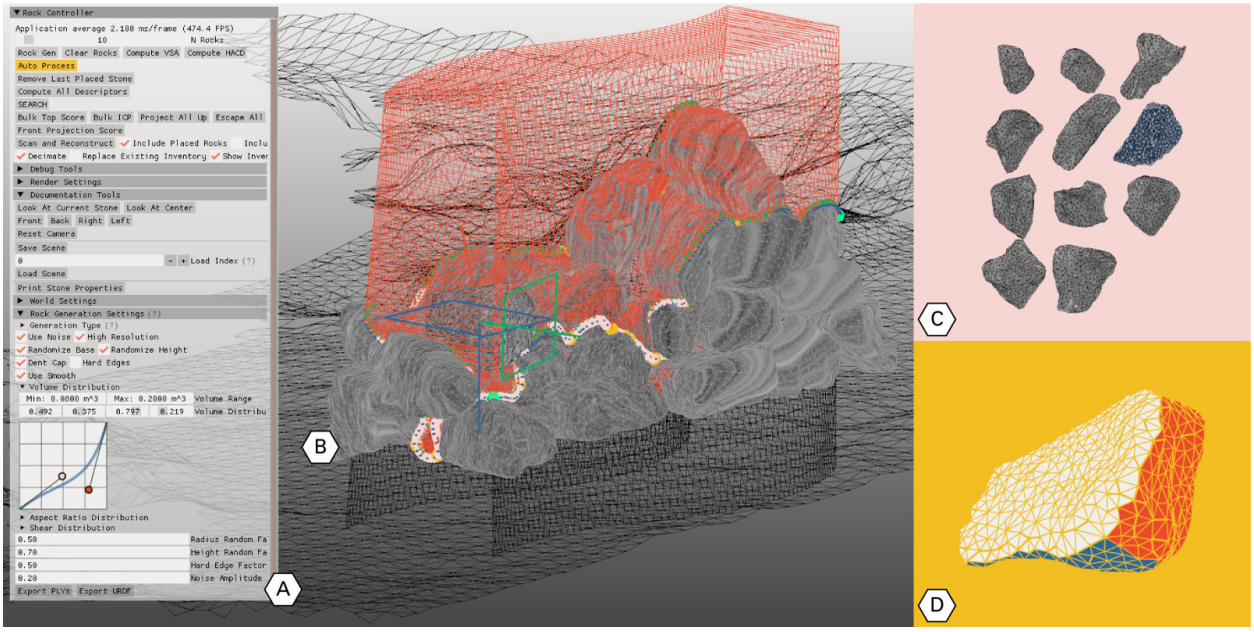


Figure 7.6: Screen capture of geometric planning software. A) Imgui Interface, B) Active environment with search space, solutions, and placed stones, C) Available scanned stones in inventory, D) Selected Stone Viewer with VSA faces.

placement. A provided Imgui interface [131] allows for alternative high-scoring solutions to be manually selected if desired.

The geometric planning software makes extensive use of libigl [132] for mesh processing, PCL for point cloud operations [32], and openframeworks [133] for visuals (Figure 7.6).

7.2.4 Grasp and Placement Execution

The goal of the grasp pose planning is to find possible grasping configurations that allow the excavator to pick the desired stone and place it at the planned location without collisions with the existing wall (Figure 7.7E). A grasp configuration is defined as the 6 DoF pose of the excavator's gripper, where contact with the desired stone can be performed. In order to find a possible grasp configuration that respects the collision constraints, we sample a large number of grasp hypotheses on the stone of interest in the map point cloud (see Section 4.4.2). Those grasp hypotheses are validated for collisions by intersecting a polyhedral gripper model with the map cloud and searching for inliers. Note that both the grasp configuration and the corresponding placement configuration must be verified to be without collisions (Figure 7.7D/F). Due to occlusions, the map point cloud may contain holes

Figure 7.7: Grasp planning and placement of a stone to its desired location (red): (A) Collision and localization point cloud, (B) Stones localized in point cloud, (C) Spline motion trajectory, (D, Green) Collision free grasp hypotheses, (F, Red) Grasp hypotheses with collisions, (E) Selected grasp configuration, (G) The collision point cloud is augmented with point clouds of already localized stones.

that can result in undetected collisions. In order to prevent these situations, we augment the map point cloud with point clouds generated from the meshes of already localized objects (Figure 7.7G). After ruling out colliding configurations, the remaining grasp poses are evaluated for force closure and scored by task-specific criteria (alignment to ground, grasp encompassment, and distance to stone CoM) to obtain the final grasp (see Section 4.4.3).

A motion planner is used to generate a spline trajectory for reaching the desired grasp pose and moving from the grasp pose to the placement pose (Figure 7.7C). The approach direction during placement is chosen by raycasting to provide the highest margin to obstacles. Figure 7.7 shows the spline trajectory (C, blue) for moving the grasped stone from the ground to its desired placement pose (E, red).

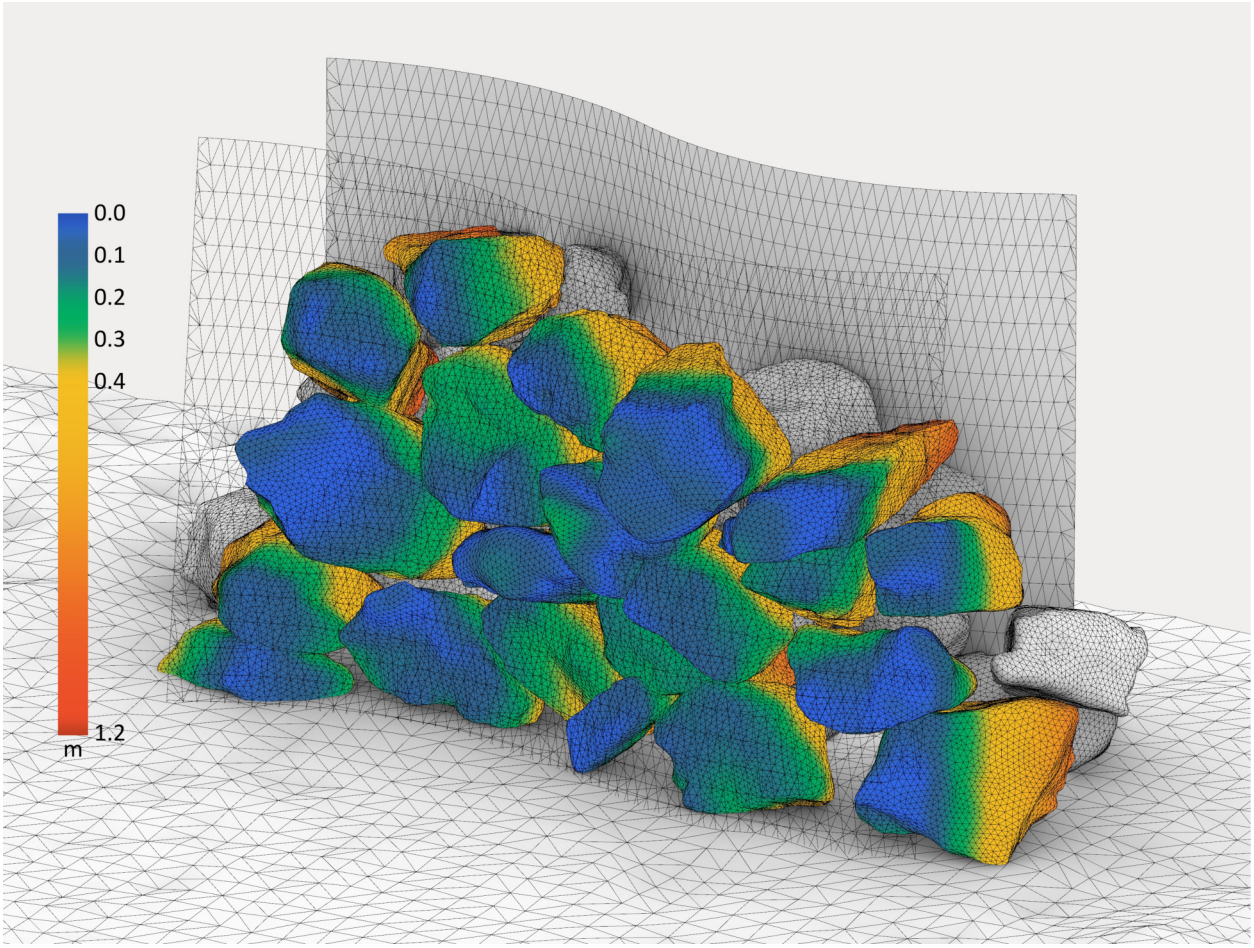


Figure 7.8: Digital model of constructed wall with goal surfaces and indicated distance between placed stones and one target surface.

Each time a stone is grasped or released, its pose with respect to the map or end-effector is updated with an ICP refinement step using its previous location as an initial guess (Section 7.2.2).

7.3 Experiment

The localization, scanning, geometric- and motion- and planning routines developed in this work were used to construct a large-scale (3 m high and 5 m long) test wall using 40 available gneiss stones[2] with an average mass of 757 kg (range: 230-1584 kg). To our knowledge, at construction time, it represented both the greatest height and number of components of any masonry structure constructed au-

2 Gneiss is a locally abundant metamorphic rock often used for architectural applications, landscape construction, and retaining walls.

Figure 7.9: Side elevation of constructed wall conveying double-faced structure and near-vertical orientation.

tonomously from irregular stones. It was only surpassed later by the structures presented in this thesis with improved manipulation (e.g., reorientation) and planning capabilities. However, it is essential to note that the geometry of the available material heavily influences such qualifications, and one might much more easily construct a many-bonded wall with ashlar-like stones. Despite the relatively large size of the available stones,[3] the outer face of the as-built structure was measured to be generally within 10 cm of the intended target surface (Figure 7.8). Figure 7.9 shows the side elevation of the completed structure, and the front elevation can be seen in Figure 7.1.

While the geometric planner allows for solutions to be found with any number of stones in the inventory, it predictably finds better correspondences if there are more stones to choose from (at the computational expense of increased search space). For a much larger construction, one would likely scan and build in batches, keeping tens of digitized stones in the inventory and refilling it when no solution can be found within some tolerance requirements. For this experiment, there was a fixed number of available stones (40), and they were each scanned and added to the digital inventory before the construction of the wall commenced. The pre-scanning accelerated construction and provided a wide variety of stones for the initial planning steps. The planning and execution were performed sequentially, meaning that after each placement, the stone pose was refined and updated in the geometric planning before initiating the next update step. Figure 7.10 shows intermediate steps of the wall construction with two and twenty placed stones.

7.4 Improvements

Several improvements are intended, increasing process robustness and allowing for additional design constraints to influence the solver. While this initial prototype was imagined as a section of a longer wall (Figures 7.11, 7.12), future developments will resolve the particularities of terminating details at wall ends and corners: the use of alternating stretchers and headers with returned faces that both strengthen the structure and clearly define the *arris* (edge where surfaces meet) [135].

At the time of constructing the presented wall, the autonomous reorientation of stones (see Section 5.3) was not yet available. On the occasion that placement solu-

3 These are classified as 'medium-coarse boulders' on the modified Udden–Wentworth grain-size scale [134].

Figure 7.10: Wall construction in progress with two (above) and twenty (below) placed stones out of the 40 used in this experiment.

tions could not be grasped in the current stone position, an alert was generated that conveyed the need for manual intervention. The stone had to be flipped manually before relocalizing it autonomously and continuing construction.

The drawback of using LiDARs for reconstruction and ICP refinement is the need always to move the sensor to accumulate scans into the complete 3D model: this is time-consuming and renders the current sensors ill-equipped for dynamic tracking during stone flipping. This task will be made more feasible with the imminent integration of additional sensors, including RGB cameras. By texturing the stone models [136], it is possible to extract visual features and use them for recognizing a stone instance or tracking its motion [137]. Textured stone models will not only better facilitate dynamic tasks such as flipping, but could also be considered as a component in the design process. Provided that a structure contained enough stones to make a pattern legible, it would be possible to incorporate variable distributions of color, scale, or surface shapes to meet additional design goals beyond the global shape of the wall.

Whereas we currently rely on physics simulation for stability assessment, we will consider structural validation by physically probing the as-built construction in future work. The excavator – equipped with pressure sensors and an additional force-torque sensor – would serve as both a construction tool and testing device, using applied loads to verify the stability of the structure and measure deformations.

7.5 Discussion on Sustainability

This research demonstrates a robotically-enabled process for creating structures from a wide range of input and target geometries. It allows digital design goals to be integrated with existing ancient construction methods for double-faced, free-standing dry stone walls built from unprocessed or minimally processed natural stone. In contrast to cut stone (and even more so to concrete), these materials exhibit extremely low embodied energy because there is minimal transportation required[4] and because they do not need energy-intensive cutting or post-processing. While the embodied energy of stone construction varies regionally, post-processing (sawing, chiseling) is a significant contributor to the overall environmental impact of masonry architecture. Despite an increase in the thickness of the built structure

4 For this experiment, the raw stones were delivered by truck from a quarry located 40 km from the test field.

Figure 7.11: In progress wall with overlay of potential extension of structure, generated by allowing the solver to continue with additional digital stone models.

Figure 7.12: Constructed wall with figure (left) for scale and indicated potential extensions.

when compared to cut stone, building with raw stone can be environmentally advantageous [138].

Rather than modifying individual stones, this method focuses on cleanly fitting the exposed surfaces of stones along with the outer two faces of a given wall section while tolerating necessary gaps between stones within the hidden space of the poché. Even in a well-constructed dry stone wall, these gaps can account for 20-40% of the constructed volume [139, 140]. They are essential in enabling the use of entirely irregular and indeterminate source materials: a dry stone wall is thus "defined by the spaces between the stones as much as it is by the stones themselves" [141]. While such gaps can be filled in with smaller 'chinking' stones [1] or with mortar, they can also be left as supplemental habitats for flora and fauna to thrive [142].

Dry stone construction has diminished in popularity in the last century, despite the environmental, economic, and aesthetic advantages of building with inexpensive, local, and natural materials. Increasing labor costs combined with comparatively inexpensive and simple-to-install mass-manufactured building components have rendered irregular stone construction infeasible in many situations [143, 144]. By synthesizing knowledge from manual dry stone masonry handbooks [1, 129, 135] together with new methods of computational design and robotic construction, this work aspires to reactivate irregular stone construction while enabling new modes of design expression and large-scale fabrication.

7.6 Summary

In this chapter, we introduce a platform and algorithm for the autonomous construction of large-scale dry stone walls in situ, with the aid of a customized robotic excavator – addressing the challenge of using unprocessed and locally available construction material. We outlined the complete perception process from mapping, digitizing, and localizing irregular stones. A core element of this work is the planning algorithm that determines the position and orientation of the stones to align with a target wall shape coming from the landscape design. The planner determines stable poses by combining heuristics from masonry textbooks with iterative geometric shape matching and physics-based settling. This planner enables the sound construction of extended structures as it generates interlocking between multiple objects. Finally, we demonstrate the applicability of the presented method

by constructing a 3-m-tall double-faced wall out of 40 stones with an average weight of 760 kg.

Part IV

CONCLUSIONS AND OUTLOOK

8

Conclusion and Outlook

In this dissertation, we have studied autonomous robotic assembling under uncertainties in unstructured real-world applications. We focused on the task of assembling objects diverse in shape without further modification or usage of adhesives. As an example of digital fabrication, this task is right on the intersection between robotics, including manipulation, navigation, sensing, control, and structural planning, including architectural design, structural analysis, and assembling algorithms, encouraging interdisciplinary collaborations. We divided this challenging problem into three parts:

- **Part I:** Development and control of a compliant mobile manipulation platform designed for autonomous mission
- **Part II:** Perception of object instances in the environment and manipulation of them
- **Part III:** Implementation of the proposed tools to create structurally stable dry-stone assemblies

In the first part, we discuss the robotic fundamentals necessary to navigate and control a mobile manipulator interacting with an unstructured environment. We show a suitable mobile platform capable of running autonomous missions. It has a compliant arm that allows to shape contact dynamics and is robust against unexpected contacts and disturbances. The second part covers the perception and manipulation of previously unseen large-scale objects to place them in confined spaces, including grasping and reorientation. The last part includes the structural planning to create dry-stone assemblies and the monitoring of the object pose to inform the assembly planner about the current construction state. Finally, the approaches presented in this thesis for manipulating irregularly shaped objects enabled the construction of the world's first large-scale stone wall using an autonomous excavator. We manipulated more than a hundred stones that each weighs several hundred kilograms and have a unique and highly diverse geometry (see Figure 8.1). In the

Figure 8.1: The applicability of the proposed grasp approaches in mapping, manipulation, and assembly planning is directly shown in the construction of a large-scale double-faced dry-stone wall with dimensions approx. 10 m x 4 m x 2 m (W x H x D).

remainder of the conclusion, we discuss each chapter's contributions and highlight future directions.

8.1 Mobile Manipulator for Interaction Tasks

We presented a mobile manipulation platform capable of autonomously executing manipulation tasks relying solely on its onboard sensing. The system consists of a standard four-wheeled skid steer platform and a custom-built 6 DoF robot arm composed of SEAs. The platform is capable of autonomous navigation and obstacle detection to reach desired target locations safely. For arm control, we combine model-based feedback linearization with low-impedance joint stabilization, enabling not only to move the end-effector precisely but also to measure and control contact forces and torques accurately during an interaction. The low-impedance manipulation control proved essential for accomplishing a complex interaction task like inserting a wrench on a valve stem. It helps mitigating misalignments of the end-effector that can hardly be estimated from a vision system like a monocular camera. A mission task dispatcher introduces autonomy to the system, activating

the required software modules and monitoring the progress and success, making it resilient to failures and demonstrating the ability of a mobile robot to perform successful manipulation in a demanding real-world application.

The robot arm introduced in this chapter was used further by our team and others for developing control algorithms that make use of its compliance. While we focused in this thesis on exploiting the dynamics of the actuators for accurate dynamic manipulation and disturbance rejections (see Chapter 3), we also present a learning-based method for improving tracking with compliant actuators in [145]. In this work, we utilize a Gaussian Process trained offline to estimate the mismatch between the actual and estimated model and combine it with an EKF filter to estimate the residual model mismatch online to provide offset-free tracking. In [146], we present a contact-implicit optimization approach that uses the high dynamics and the impact robustness of the arm to tackle various dynamic pushing problems.

8.2 Actuator-Aware Model Predictive Control for Dynamic Manipulation

Compliant manipulators are composed of materials or use actuation modes that are flexible and soft, allowing them to exploit the interaction between the robot and the environment. The inherent softness of these systems provides adaptability, robustness, and safety, which is especially desirable for assembling tasks of objects with uncertainty in pose and shape. However, the combination of compliance with accurate and dynamic motion is still an unsolved challenge in robotics as the elastic elements introduce unwanted intrinsic oscillatory dynamics to the system, causing underactuation, and reduce the system's natural frequency. Furthermore, the actuators are often designed to have low friction and damping to improve efficiency, aggravating the issue. In our work, we presented an MPC formulation that incorporates the dynamics of compliant actuators to tackle this problem and perform accurate task-space tracking. We derived in detail how different control approaches alter the natural stiffness of the system and show that the presented MPC formulation alters the natural compliance least. In experiments, we demonstrated that the proposed controller is robust against unmodeled end-effector inertia. By including the actuator dynamics, we improve the high-frequency robustness and directly address the issue of the limited actuator bandwidth as the real physical spring acts

as a proportional gain to drive the link to its desired position, generating smoother and physically more consistent motions. To the best of our knowledge, this is the first time that a task-space MPC formulation including the actuator dynamics is applied to a multi-joint robot with compliant actuators.

Our proposed method to include the actuator dynamics in the control formulation is not limited only to a specific kind of actuation like SEAs. For example in the future, we are planning to port the actuator-aware MPC formulation to the excavator's arm. The hydraulic actuators exhibit complex nonlinear dynamics due to hydraulic coupling between the actuators, cylinder friction, control input deadzones, and delays. The idea is to learn an actuator model based on measurements collected during operation [147] and use it in the receding horizon controller to track the end-effector more accurately.

8.3 Large-Scale Object Mapping, Segmentation, and Manipulation

To bring our work to an architectural scale, we presented an integrated perception and grasp pose planning system for autonomous manipulation of large-scale irregular objects with a robotic excavator. The online laser mapping system provides a consistent map over an extended time horizon to segment new object instances and to localize objects that moved. A segment-based registration scheme allows fusing mapping data acquired from heterogeneous external sensors (e.g., a drone-borne camera) into the LiDAR-based map. The robust and collision-aware grasp planning method allows picking irregular objects in slightly cluttered and occluded scenes. We demonstrate our autonomous mapping, segmentation, and grasp planning in real-world experiments and use them in actual applications to build wall-like irregular dry-stone assemblies on an architectural scale (see Chapter 5, 7). The achieved autonomy for an architectural-scale construction task is unprecedented, and the developed tools are not committed to the presented application only. They can be employed with different planners for autonomous assembling and demolition.

A limitation of the current system is that the stones need to be initially spread for accurate segmentation. While this is still a realistic scenario that can be achieved by careful unloading of the material on-site, we aim at handling even more generic and practical situations in the future. Therefore, an important area of research will focus on identifying individual object instances in more challenging settings, such as piles of stones. For this, current work investigates the addition of a camera on

the excavator's cabin or arm, which could potentially map regions of the scene that are not visible by the onboard LiDARs and enable more advanced segmentation techniques leveraging both texture and geometry.

8.4 Grasp Pose Planning and Object Reorientation

Focusing on the assembling of complicated structures, we showed a grasp pose sampling pipeline suited for manipulating objects in cluttered and obstructed scenes. An essential part of this work is to assess the feasibility of a grasp considering the collision constraints at pick and place and to plan a reorientation sequence if necessary, which is unique for objects with such irregular shapes and large sizes. The proposed approach enabled constructing the world's first large-scale dry-stone wall using an autonomous excavator and stones with a unique and highly diverse geometry, weighing more than a ton on average. However, the manipulation approach itself is task agnostic and could be extended, besides landscape construction, to industrial assembling or household and service robotics. To further improve construction process reliability, the assembly planner could already consider the current object pose for finding possible placement poses to avoid time-consuming reorientation and reduce the number of grasp attempts.

It is not realistic to achieve a perfect success rate on the first grasp attempt for a complex task like collision-free assembly with irregularly shaped objects. Therefore, we plan to improve the autonomy of the grasping process with an enhanced recovery strategy, inspired by how a human operator grasps irregular objects: slightly adapting the grasp during gripper closing instead of performing complete re-localization and re-grasping upon detection of slippage or dropping.

8.5 Vertical Stone Towers

In the chapter about vertical stone tower creation, we investigated integrating a pose searching algorithm that considers structural stability using a physics engine. We evaluated the method with an autonomous system: a table-top setup consisting of a fixed-base robot arm equipped with a 3-finger gripper and end-effector mounted depth camera. A vision pipeline successfully determined precise locations of known stone models in the scene and the gripper. Our simulation frame-

work provided us with the next best target locations to one of the identified stones on the already built stack based on a heuristic cost function. We integrated a motion planning library to grasp and place the stones, and despite the greedy nature of our next stone placement strategy, we showed that our system could successfully stack up to four stones.

While this setup successfully proved that pose search in simulation, using a heuristic cost function, generates solutions that can be executed to stack stones in a laboratory setup, two significant limitations prevent the direct transfer to a real-world scenario. First, our pipeline requires accurate stone models beforehand, to localize them in the scene and for the pose search using the physics engine. Second, the planning is limited to what target shapes can be achieved. As the pose search only looks for solutions along a thrust line in contact with a single object, it is impossible to create extended structures or follow an external target shape. In [68], our proposed pose search is enhanced to allow constructing spatially extended single-layer walls, showing impressive results in a similar table-top setup. On the other hand, we considered this approach more proof-of-concept that the physics simulation results can be transferred to the real world and focused on an approach that can create desired target designs by combining physics settlement with geometric alignment (see Chapter 7).

8.6 Autonomous Dry Stone

Finally, we brought together all developed tools in our work about autonomous dry stone and introduced a platform for autonomous construction of large-scale dry stone walls on site. We outlined the complete process from mapping, digitizing, and localizing irregular stones with the aid of a customized robotic excavator. A core element of this work is the planning algorithm that determines the position and orientation of the stones to align with a target wall shape coming from the landscape design. The planner determines stable poses by combining heuristics from masonry textbooks with iterative geometric shape matching and physics-based settling.

In the end, we achieved with this method our goal of constructing spatially extended structures composed of irregularly shaped objects in complex contact situations. Furthermore, the addition of a high-payload, fully-mobile platform has greatly expanded the 'machinic morphospace' [148] of robotic construction tools

on-site, allowing us, in turn, to navigate better the constrained 'material morphospace' of complex and nonstandard found objects [149]. The process can significantly reduce the embodied energy of architectural construction in specific contexts and can be extended to work with a range of source materials, including reclaimed demolition debris. By realizing a double-faced wall on an architectural scale, we demonstrate the applicability of automated stone masonry for future construction processes.

8.7 Outlook

This thesis addresses fundamental prerequisites for bringing autonomous mobile robots from clearly defined industrial applications to the unstructured and cluttered scene of a construction site. We already demonstrated the ability to build in real-world structures on an impressive scale. However, many more aspects require consideration before autonomous construction robots become widespread in reality. First of all, the production rate should be increased to match or exceed human operation and exploit the machine's full capabilities. While all the single steps for reconstructing, localizing, grasping, and placing an object are automated, each placement sequence is still initiated manually. For a system to work in a remote place where only limited or no surveillance is feasible, it needs further effort in improving the robustness and resilience against unexpected events and disturbances. However, it is not the intention that construction robots replace human operators one-to-one. Instead, they should go beyond the capabilities of today's manually operated machines by allowing unprecedented design and improving sustainability by using on-site material and increasing productivity. In a first step, one could also imagine that the provided tools can help teleoperate a machine in an abstract and intuitive way. E.g., through an augmented reality device, segmented objects can be selected for grasping, and the machine is executing it autonomously. While these are complex engineering problems, we believe that the robotics community will solve them to a satisfying extent within the next few years.

To handle extensive landscaping projects in the future, we have to be economical and strategic with computational power and data storage. The total generated amount of data from object modeling and assembly planning might overwhelm the resources. We envision using a database to store all the incoming data and only provide the required sub-selection of it to the grasp and assembly planning.

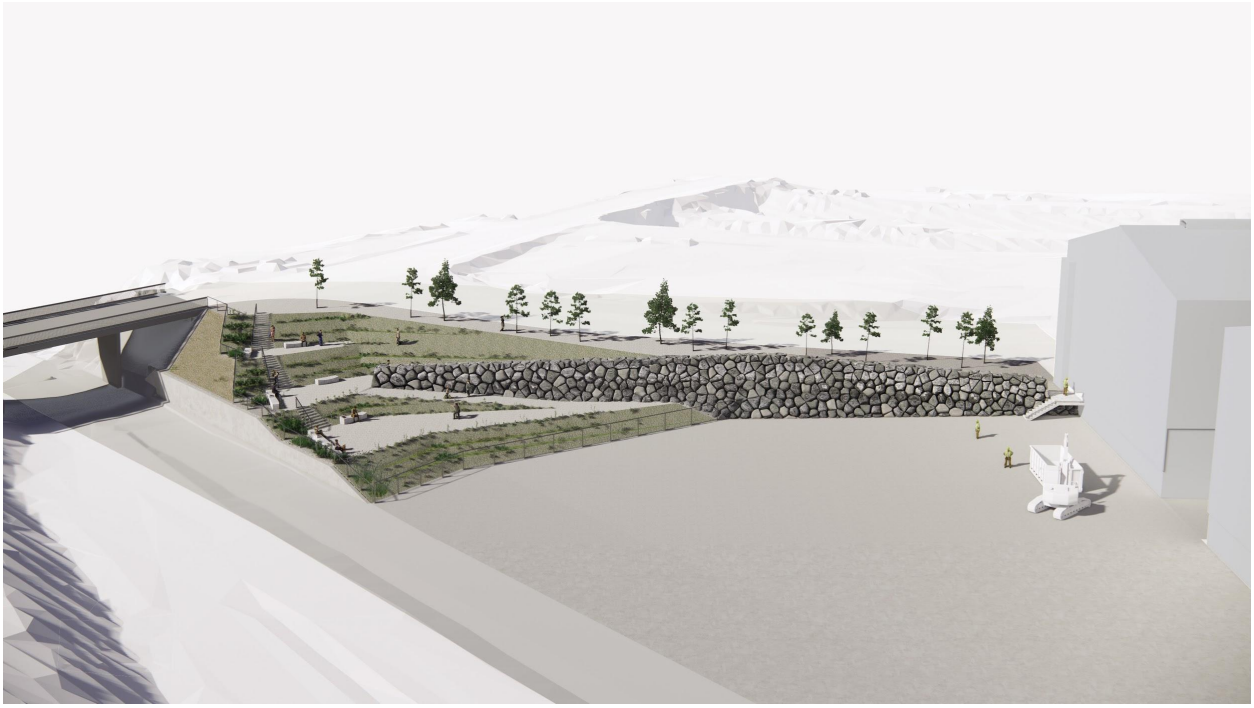


Figure 8.2: In collaboration with Eberhard Bau AG, a permanent demonstrator of approx. 80 m including dry stone elements will be erected in 2021 with the autonomous excavator HEAP.

A further advantage of the database is that multiple autonomous agents can share information and simultaneously work on extended structures. Another aspect that deserves to be taken into account for constructing spatially extended structures is on-site logistics. The assembly planner should be informed to favor objects in close vicinity of the robot and the desired placement pose to minimize robot motion. Furthermore, if multiple robots build on the same structure, the tasks must be distributed and coordinated between them.

In collaboration with Eberhard Bau AG, our autonomous excavator HEAP will be in operation during spring and summer 2021 to create the world's first permanent retaining wall at the construction debris recycling plant EbiMik[1] in Oberglatt, Zürich, Switzerland, using the methods for object manipulation and assembly planning presented in this thesis. The dry stone wall consists of approximately 900 stones and concrete rubbles and will be part of a publicly available space informing about digital fabrication (see Figure 8.2). This extensive project aims to validate and improve our presented methods in terms of reliability and construction pace,

1 Eberhard Bau AG, https://eberhard.ch/ebimik/ (Accessed: 2021-04-15)

monitor the relevant data to assess its sustainability, and gain information about the process's robustness and the possibility to transfer the technology to industry.

A

Appendix

A.1 Actuator-Aware Model Predictive Control for Dynamic Manipulation

A.1.1 Inverse Dynamics PID Gains

Gain	Value
$\boldsymbol{k}_P$	$\begin{pmatrix} 150 & 0 & 0 & 0 & 0 & 0 \\ 0 & 150 & 0 & 0 & 0 & 0 \\ 0 & 0 & 150 & 0 & 0 & 0 \\ 0 & 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 50 \end{pmatrix}$
$\boldsymbol{k}_I$	$\begin{pmatrix} 15 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$
$\boldsymbol{k}_D$	$\begin{pmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Table A.1: Inverse dynamics PID gains.

A.1.2 MPC gains

$\boldsymbol{Q}$	MPC $\boldsymbol{\tau}$	MPC $\boldsymbol{v}_m$
$\boldsymbol{Q}_{\text{pos}}$	$300 \cdot \boldsymbol{I}_3$	$1600 \cdot \boldsymbol{I}_3$
$\boldsymbol{Q}_{\text{o}}$	$50 \cdot \boldsymbol{I}_3$	$9600 \cdot \boldsymbol{I}_3$
$\boldsymbol{Q}_{\boldsymbol{w}}$	$0.01 \cdot \boldsymbol{I}_6$	$1.5 \cdot \boldsymbol{I}_6$
$\boldsymbol{Q}^{\text{f}}_{\text{pos}}$	$20 \cdot \boldsymbol{I}_3$	$25 \cdot \boldsymbol{I}_3$
$\boldsymbol{Q}^{\text{f}}_{\text{o}}$	$1 \cdot \boldsymbol{I}_3$	$0.5 \cdot \boldsymbol{I}_3$
$\boldsymbol{Q}^{\text{f}}_{\boldsymbol{w}}$	$0.01 \cdot \boldsymbol{I}_6$	$1.7 \cdot \boldsymbol{I}_6$
$\boldsymbol{R}_{\boldsymbol{\nu}}$	$10^{-3} \cdot \begin{pmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix}$	$10^{-3} \cdot \begin{pmatrix} 1.25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.5 \end{pmatrix}$

Table A.2: Weighting Matrices of the MPC controllers

Bibliography

1. Vivian, J. *Building Stone Walls* 114 (Storey Publishing, LLC, 1976).
2. Melenbrink, N., Werfel, J. & Menges, A. On-site autonomous construction robots: Towards unsupervised building. *Automation in Construction* **119**, 103312 (2020).
3. Dörfler, K., Sandy, T. & Giftthaler, M. in *Robotic Fabrication in Architecture, Art and Design 2016* (eds Reinhardt, D., Saunders, R. & Burry, J.) June, 204 (Springer International Publishing, Cham, 2016).
4. Fastbrick Robotics. *Hadrian X* https://www.fbr.com.au/view/hadrian-x. Accessed: 2021-04-15.
5. Keating, S. *A Compound Arm Approach to Digital Construction* (eds McGee, W. & Ponce de Leon, M.) **March 2014**, 99 (Springer International Publishing, Cham, 2014).
6. Jud, D., Leemann, P., Kerscher, S. & Hutter, M. Autonomous Free-Form Trenching Using a Walking Excavator. *IEEE Robotics and Automation Letters* **4**, 3208 (2019).
7. Thangavelu, V., da Silva, M. S., Choi, J. & Napp, N. *Autonomous Modification of Unstructured Environments with Found Material* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), 7798.
8. Saboia, M., Thangavelu, V. & Napp, N. Autonomous multi-material construction with a heterogeneous robot team. *Robotics and Autonomous Systems* **121**, 103239 (2019).
9. Fujisawa, R., Nagaya, N., Okazaki, S., Sato, R., Ikemoto, Y. & Dobata, S. Active modification of the environment by a robot with construction abilities. *ROBOMECH Journal* **2**, 9 (2015).
10. Sanders, G. & Larson, W. Progress Made in Lunar In Situ Resource Utilization under NASA's Exploration Technology and Development Program. *Earth and Space 2012*, 457 (2012).

11. Eckman, E., Peck, M. A. & Napp, N. *Lunar Infrastructure via Microscale Regolith Assembly* in *AIAA Scitech 2021 Forum* (American Institute of Aeronautics and Astronautics, Reston, Virginia, 2021), 1.

12. Petersen, K. H., Napp, N., Stuart-Smith, R., Rus, D. & Kovac, M. A review of collective robotic construction. *Science Robotics* **4**, eaau8479 (2019).

13. Johns, R. L. & Anderson, J. S. *Interfaces for Adaptive Assembly* in *ACADIA // 2018: Recalibration. On Imprecisionand Infidelity* (CUMINCAD), 126.

14. De Boer, J., Klinger, K., Vermillion, J., Greenberg, B., Hittler, G. & Perry, K. *Smart Scrap* http://i-m-a-d-e.org/?p=2201. Accessed: 2021-04-15.

15. Clifford, B., McGee, W. & Muhonen, M. Recovering Cannibalism in Architecture with a Return to Cyclopean Masonry. *Nexus Network Journal* **20**, 583 (2018).

16. Bodie, K., Bellicoso, C. D. & Hutter, M. *ANYpulator: Design and Control of a Safe Robotic Arm* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2016), 1119.

17. Bähnemann, R., Pantic, M., Popović, M., Schindler, D., Tranzatto, M., Kamel, M., Grimm, M., Widauer, J., Siegwart, R. & Nieto, J. The ETH-MAV Team in the MBZ International Robotics Challenge. *Journal of Field Robotics* **36**, 78 (2019).

18. Chen, X., Kimura, K., Mizohana, H., Zhao, M., Shi, F., Chaudhary, K., Chan, W. P., Nozawa, S., Kakiuchi, Y., Okada, K. & Inaba, M. *Development of task-oriented high power field robot platform with humanoid upper body and mobile wheeled base* in *IEEE/SICE International Symposium on System Integration (SII)* (IEEE, 2016), 349.

19. Shaqura, M. & Shamma, J. S. *A novel gripper design for multi hand tools grasping under tight clearance constraints and external torque effect* in *IEEE International Conference on Mechatronics and Automation (ICMA)* (IEEE, 2017), 840.

20. Gandin, S. *Object Detection e Visual Servoing per applicazioni robotiche di grasping e manipolazione* Master's thesis (University of Padua, 2017).

21. Engemann, H., Wiesen, P., Kallweit, S., Deshpande, H. & Schleupen, J. in *Ferraresi C., Quaglia G. (eds) Advances in Service and Industrial Robotics. RAAD 2017. Mechanisms and Machine Science* 389 (Springer International Publishing, 2018).

22. Carius, J., Wermelinger, M., Rajasekaran, B., Holtmann, K. & Hutter, M. *Autonomous Mission with a Mobile Manipulator—A Solution to the MBZIRC* in *Field and Service Robots (FSR)* ("Springer International Publishing, 2018), 559.

23. Castaman, N., Tosello, E., Antonello, M., Bagarello, N., Gandin, S., Carraro, M., Munaro, M., Bortoletto, R., Ghidoni, S., Menegatti, E. & Pagello, E. RUR53: an unmanned ground vehicle for navigation, recognition, and manipulation. *Advanced Robotics* **35**, 1 (2020).

24. Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C. D., Tsounis, V., Hwangbo, J., Bodie, K., Fankhauser, P., Bloesch, M., Diethelm, R., Bachmann, S., Melzer, A. & Hoepflinger, M. *ANYmal - a highly mobile and dynamic quadrupedal robot* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2016), 38.

25. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. & Ng, A. Y. *ROS: an open-source Robot Operating System* in *ICRA workshop on open source software* **3** (2009), 5.

26. Pomerleau, F., Colas, F., Siegwart, R. & Magnenat, S. Comparing ICP variants on real-world data sets. *Autonomous Robots* **34**, 133 (2013).

27. Pomerleau, F., Krusi, P., Colas, F., Furgale, P. & Siegwart, R. *Long-term 3D map maintenance in dynamic environments* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), 3712.

28. Krüsi, P., Furgale, P., Bosse, M. & Siegwart, R. Driving on Point Clouds: Motion Planning, Trajectory Optimization, and Terrain Assessment in Generic Nonplanar Environments. *Journal of Field Robotics* **34**, 940 (2017).

29. Krüsi, P., Bücheler, B., Pomerleau, F., Schwesinger, U., Siegwart, R. & Furgale, P. Lighting-invariant Adaptive Route Following Using Iterative Closest Point Matching. *Journal of Field Robotics* **32**, 534 (2015).

30. Skiena, S. Dijkstra's algorithm. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley*, 225 (1990).

31. Fox, D., Burgard, W. & Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine* **4**, 23 (1997).

32. Rusu, R. B. & Cousins, S. *3D is here: Point Cloud Library (PCL)* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2011), 1.

33. Fischler, M. A. & Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **24**, 381 (1981).

34. Hearst, M., Dumais, S., Osuna, E., Platt, J. & Scholkopf, B. Support vector machines. *IEEE Intelligent Systems and their Applications* **13**, 18 (1998).

35. Dalal, N. & Triggs, B. *Histograms of Oriented Gradients for Human Detection* in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* **1** (IEEE, 2005), 886.

36. Joachims, T. *Making Large-Scale SVM Learning Practical* tech. rep. (TU Dortmund, 1998).

37. Flandin, G., Chaumette, F. & Marchand, E. *Eye-in-hand/eye-to-hand cooperation for visual servoing* in *IEEE International Conference on Robotics and Automation (ICRA)* **3** (IEEE, 2000), 2741.

38. Davies, E. R. *Machine vision: theory, algorithms, practicalities* (Elsevier Science, 2004).

39. Suzuki, S. & Be, K. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing* **30**, 32 (1985).

40. Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 1330 (2000).

41. Corke, P. I. in, 1 (1993).

42. Bashir, F. & Porikli, F. in *Proceedings of the 9th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)* 7 (2006).

43. Buss, S. R. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation* **17**, 16 (2004).

44. Gawel, A., Siegwart, R., Hutter, M., Sandy, T., Blum, H., Pankert, J., Kramer, K., Bartolomei, L., Ercan, S., Farshidian, F., Chli, M. & Gramazio, F. *A Fully-Integrated Sensing and Control System for High-Accuracy Mobile Robotic Building Construction* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2019), 2300.

45. Palli, G., Melchiorri, C. & De Luca, A. *On the Feedback Linearization of Robots with Variable Joint Stiffness* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2008), 1753.

46. Petit, F., Daasch, A. & Albu-Schaffer, A. Backstepping Control of Variable Stiffness Robots. *IEEE Transactions on Control Systems Technology* **23**, 2195 (2015).

47. Sirouspour, M. & Salcudean, S. Nonlinear control of hydraulic robots. *IEEE Transactions on Robotics and Automation* **17**, 173 (2001).

48. Khaligh, Y. S. & Namvar, M. *Adaptive control of robot manipulators including actuator dynamics and without joint torque measurement* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2010), 4639.

49. Farshidian, F., Jelavic, E., Winkler, A. W. & Buchli, J. *Robust whole-body motion control of legged robots* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2017), 4589.

50. Vorndamme, J., Schappler, M., Todtheide, A. & Haddadin, S. *Soft robotics for the hydraulic atlas arms: Joint impedance control with collision detection and disturbance compensation* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2016), 3360.

51. Della Santina, C., Bianchi, M., Grioli, G., Angelini, F., Catalano, M., Garabini, M. & Bicchi, A. Controlling Soft Robots: Balancing Feedback and Feedforward Elements. *IEEE Robotics & Automation Magazine* **24**, 75 (2017).

52. Keppler, M., Lakatos, D., Ott, C. & Albu-Schaffer, A. Elastic Structure Preserving (ESP) Control for Compliantly Actuated Robots. *IEEE Transactions on Robotics* **34**, 317 (2018).

53. Angelini, F., Santina, C. D., Garabini, M., Bianchi, M., Gasparri, G. M., Grioli, G., Catalano, M. G. & Bicchi, A. Decentralized Trajectory Tracking Control for Soft Robots Interacting With the Environment. *IEEE Transactions on Robotics* **34**, 924 (2018).

54. Keppler, M., Lakatos, D., Ott, C. & Albu-Schaffer, A. *Elastic Structure Preserving Impedance ($ES\pi$) Control for Compliantly Actuated Robots* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* **34** (IEEE, 2018), 5861.

55. Keppler, M., Lakatos, D., Werner, A., Loeffl, F., Ott, C. & Albu-Schaffer, A. *Visco-Elastic Structure Preserving Impedance (VESπ) Control for Compliantly Actuated Robots* in *European Control Conference (ECC)* (IEEE, 2018), 255.

56. Hutter, M., Remy, C., Hoepflinger, M. A. & Siegwart, R. Y. *HIGH COMPLIANT SERIES ELASTIC ACTUATION FOR THE ROBOTIC LEG SCARL ETH* in *Field Robotics* (WORLD SCIENTIFIC, 2011), 507.

57. Nakanishi, J., Cory, R., Mistry, M., Peters, J. & Schaal, S. Operational Space Control: A Theoretical and Empirical Comparison. *The International Journal of Robotics Research* **27**, 737 (2008).

58. Diehl, M., Bock, H., Diedam, H. & Wieber, P.-B. in *Fast Motions in Biomechanics and Robotics* 65 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006).

59. Giftthaler, M., Neunert, M., Stauble, M., Buchli, J. & Diehl, M. *A Family of Iterative Gauss-Newton Shooting Methods for Nonlinear Optimal Control* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), 1.

60. Farshidian, F., Neunert, M., Winkler, A. W., Rey, G. & Buchli, J. *An efficient optimal planning and control framework for quadrupedal locomotion* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017), 93.

61. Diehl, M., Bock, H. G. & Schlöder, J. P. A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control. *SIAM Journal on Control and Optimization* **43**, 1714 (2005).

62. Bell, B. M. CppAD: a package for C++ algorithmic differentiation. *Computational Infrastructure for Operations Research* **57**, 10 (2012).

63. Siciliano, B., Sciavicco, L., Villani, L. & Oriolo, G. *Robotics: modelling, planning and control* 632 (Springer Science & Business Media, 2010).

64. Farshidian, F., Kamgarpour, M., Pardo, D. & Buchli, J. Sequential Linear Quadratic Optimal Control for Nonlinear Switched Systems. *IFAC-PapersOnLine* **50**, 1463 (2017).

65. Grandia, R., Farshidian, F., Dosovitskiy, A., Ranftl, R. & Hutter, M. Frequency-Aware Model Predictive Control. *IEEE Robotics and Automation Letters* **4**, 1517 (2019).

66. Ellis, G. *Control System Design Guide: Using Your Computer to Understand and Diagnose Feedback Controllers* (Butterworth-Heinemann, 2012).

67. Saboia da Silva, M., Thangavelu, V., Gosrich, W. & Napp, N. *Autonomous Adaptive Modification of Unstructured Environments* in *Robotics: Science and Systems XIV* (Robotics: Science and Systems Foundation, 2018).

68. Liu, Y., Choi, J. & Napp, N. *Planning for Robotic Dry Stacking with Irregular Stones* in *Field and Service Robotics* (eds Ishigami, G. & Yoshida, K.) (Springer Singapore, Singapore, 2021), 321.

69. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. & Leonard, J. J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics* **32**, 1309 (2016).

70. Mendes, E., Koch, P. & Lacroix, S. *ICP-based pose-graph SLAM* in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (IEEE, 2016), 195.

71. Droeschel, D., Schwarz, M. & Behnke, S. Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner. *Robotics and Autonomous Systems* **88**, 104 (2017).

72. Shan, T. & Englot, B. *LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), 4758.

73. Furrer, F., Novkovic, T., Fehr, M., Gawel, A., Grinvald, M., Sattler, T., Siegwart, R. & Nieto, J. *Incremental Object Database: Building 3D Models from Multiple Partial Observations* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), 6835.

74. Mccormac, J., Clark, R., Bloesch, M., Davison, A. & Leutenegger, S. *Fusion++: Volumetric Object-Level SLAM* in *International Conference on 3D Vision (3DV)* (IEEE, 2018), 32.

75. Grinvald, M., Furrer, F., Novkovic, T., Chung, J. J., Cadena, C., Siegwart, R. & Nieto, J. Volumetric Instance-Aware Semantic Mapping and 3D Object Discovery. *IEEE Robotics and Automation Letters* **4**, 3037 (2019).

76. Dube, R., Gawel, A., Sommer, H., Nieto, J., Siegwart, R. & Cadena, C. *An online multi-robot SLAM system for 3D LiDARs* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2017), 1004.

77. Dube, R., Dugas, D., Stumm, E., Nieto, J., Siegwart, R. & Cadena, C. *SegMatch: Segment based place recognition in 3D point clouds* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017), 5266.

78. Forster, C., Pizzoli, M. & Scaramuzza, D. *Air-ground localization and map augmentation using monocular dense reconstruction* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2013), 3971.

79. Gawel, A., Dube, R., Surmann, H., Nieto, J., Siegwart, R. & Cadena, C. *3D registration of aerial and ground robots for disaster response: An evaluation of features, descriptors, and transformation estimation* in *IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)* (IEEE, 2017), 27.

80. Holz, D., Ichim, A. E., Tombari, F., Rusu, R. B. & Behnke, S. Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D. *IEEE Robotics & Automation Magazine* **22**, 110 (2015).

81. Ten Pas, A., Gualtieri, M., Saenko, K. & Platt, R. Grasp Pose Detection in Point Clouds. *The International Journal of Robotics Research* **36**, 1455 (2017).

82. Nikolic, J., Rehder, J., Burri, M., Gohl, P., Leutenegger, S., Furgale, P. T. & Siegwart, R. *A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), 431.

83. Qin, T., Li, P. & Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics* **34**, 1004 (2018).

84. Schonberger, J. L. & Frahm, J.-M. *Structure-from-Motion Revisited* in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), 4104.

85. Grisetti, G., Kümmerle, R., Stachniss, C. & Burgard, W. A Tutorial on Graph-Based SLAM. *IEEE Intelligent Transportation Systems Magazine* **2**, 31 (2010).

86. Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J. & Dellaert, F. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research* **31**, 216 (2012).

87. Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P. & Frenkel, A. *On the segmentation of 3D LIDAR point clouds* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2011), 2798.

88. Zhong, Y. *Intrinsic shape signatures: A shape descriptor for 3D object recognition* in *IEEE International Conference on Computer Vision (ICCV) Workshops* (IEEE, 2009), 689.

89. Tombari, F., Salti, S. & Di Stefano, L. *Unique Signatures of Histograms for Local Surface Description* in *Computer Vision – ECCV 2010* (eds Daniilidis, K., Maragos, P. & Paragios, N.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010), 356.

90. Rusu, R. B., Blodow, N. & Beetz, M. *Fast Point Feature Histograms (FPFH) for 3D registration* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2009), 3212.

91. Guo, Y., Sohel, F., Bennamoun, M., Lu, M. & Wan, J. Rotational Projection Statistics for 3D Local Surface Description and Object Recognition. *International Journal of Computer Vision* **105**, 63 (2013).

92. Jelavic, E. & Hutter, M. *Whole-Body Motion Planning for Walking Excavators* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2019), 2292.

93. Dario Bellicoso, C., Gehring, C., Hwangbo, J., Fankhauser, P. & Hutter, M. *Perception-less terrain adaptation through whole body control and hierarchical optimization* in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (IEEE, 2016), 558.

94. Dobashi, H., Hiraoka, J., Fukao, T., Yokokohji, Y., Noda, A., Nagano, H., Nagatani, T., Okuda, H. & Tanaka, K.-i. Robust grasping strategy for assembling parts in various shapes. *Advanced Robotics* **28**, 1005 (2014).

95. Prattichizzo, D. & Trinkle, J. C. in *Springer Handbook of Robotics* 955 (Springer International Publishing, Cham, 2016).

96. Bohg, J., Morales, A., Asfour, T. & Kragic, D. Data-Driven Grasp Synthesis—A Survey. *IEEE Transactions on Robotics* **30**, 289 (2014).

97. Ten Pas, A. & Platt, R. in *Robotics Research* September, 307 (Springer, Cham, 2018).

98. Kappler, D., Bohg, J. & Schaal, S. *Leveraging big data for grasp planning* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), 4304.

99. Herzog, A., Pastor, P., Kalakrishnan, M., Righetti, L., Bohg, J., Asfour, T. & Schaal, S. Learning of grasp selection based on shape-templates. *Autonomous Robots* **36**, 51 (2014).

100. Borst, C., Fischer, M. & Hirzinger, G. *Grasping the dice by dicing the grasp* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* **3** (IEEE, 2003), 3692.

101. Wan, W., Igawa, H., Harada, K., Onda, H., Nagata, K. & Yamanobe, N. A regrasp planning component for object reorientation. *Autonomous Robots* **43**, 1101 (2019).

102. Lozano-Perez, T. & Kaelbling, L. P. *A constraint-based method for solving sequential manipulation planning problems* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2014), 3684.

103. Hou, Y., Jia, Z. & Mason, M. T. Reorienting Objects in 3D Space Using Pivoting. *arXiv preprint arXiv:1912.02752*, 1 (2019).

104. Chavan-Dafle, N., Mason, M. T., Staab, H., Rossano, G. & Rodriguez, A. *A two-phase gripper to reorient and grasp* in *IEEE International Conference on Automation Science and Engineering (CASE)* (IEEE, 2015), 1249.

105. Dafle, N. C., Rodriguez, A., Paolini, R., Tang, B., Srinivasa, S. S., Erdmann, M., Mason, M. T., Lundberg, I., Staab, H. & Fuhlbrigge, T. *Extrinsic dexterity: In-hand manipulation with external forces* in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), 1578.

106. Shi, J., Woodruff, J. Z., Umbanhowar, P. B. & Lynch, K. M. Dynamic In-Hand Sliding Manipulation. *IEEE Transactions on Robotics* **33**, 778 (2017).

107. Hou, Y., Jia, Z., Johnson, A. M. & Mason, M. T. in *Algorithmic Foundations of Robotics XII* 464 (Springer, Cham, 2020).

108. Stoll, G. *Kostenrelevante Faktoren auf Trockenmauerbaustellen*

109. Livesley, R. K. Limit analysis of structures formed from rigid blocks. *International Journal for Numerical Methods in Engineering* **12**, 1853 (1978).

110. Livesley, R. K. A computational model for the limit analysis of three-dimensional masonry structures. *Meccanica* **27**, 161 (1992).

111. Block, P., Ciblac, T. & Ochsendorf, J. Real-time limit analysis of vaulted masonry buildings. *Computers & Structures* **84**, 1841 (2006).

112. Whiting, E., Ochsendorf, J. & Durand, F. Procedural modeling of structurally-sound masonry buildings. *ACM SIGGRAPH Asia* **28**, 1 (2009).

113. Nielsen, S. A. & Dancu, A. Fusing design and construction as speculative articulations for the built environment. *Future of Architectural Research*, 65 (2015).

114. Lambert, M. & Kennedy, P. Using Artificial Intelligence to Build with Unprocessed Rock. *Key Engineering Materials* **517**, 939 (2012).

115. Cholewiak, S. A., Fleming, R. W. & Singh, M. Perception of physical stability and center of mass of 3-D objects. *Journal of Vision* **15**, 13 (2015).

116. Battaglia, P. W., Hamrick, J. B. & Tenenbaum, J. B. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences of the United States of America* **110**, 18327 (2013).

117. Ko, M.-C. *Algorithms and Automated Material Handling Systems Design for Stacking 3D Irregular Stone Pieces* PhD thesis (Texas A&M University in, 2011).

118. Sujan, V., Dubowsky, S. & Ohkami, Y. *Design and implementation of a robot assisted crucible charging system* in *IEEE International Conference on Robotics and Automation (ICRA)* **2** (IEEE, 2000), 1969.

119. Chen, H. & Bhanu, B. 3D free-form object recognition in range images using local surface patches. *Pattern Recognition Letters* **28**, 1252 (2007).

120. Furrer, F., Fehr, M., Novkovic, T., Sommer, H., Gilitschenski, I. & Siegwart, R. in *Field and Service Robots (FSR)* 145 (Springer International Publishing, Cham, 2018).

121. Alliez, P., Saboret, L. & Salman, N. Point set processing. *CGAL User and Reference Manual* **3** (2010).

122. Karavelas, M. 2D segment Delaunay graphs. *CGAL User and Reference Manual. CGAL Editorial Board* **4** (2007).

123. Chitta, S., Sucan, I. & Cousins, S. MoveIt! [ROS Topics]. *IEEE Robotics & Automation Magazine* **19**, 18 (2012).

124. Gal, R., Sorkine, O., Popa, T., Sheffer, A. & Cohen-Or, D. *3D collage* in *International symposium on Non-photorealistic animation and rendering - NPAR '07* (ACM Press, New York, New York, USA, 2007), 7.

125. Huang, Q.-X., Flöry, S., Gelfand, N., Hofer, M. & Pottmann, H. Reassembling fractured objects by geometric matching. *ACM Transactions on Graphics* **25**, 569 (2006).

126. Kazhdan, M., Bolitho, M. & Hoppe, H. *Poisson Surface Reconstruction* in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (Eurographics Association, 2006), 61.

127. Bächer, M., Whiting, E., Bickel, B. & Sorkine-Hornung, O. Spin-It: Optimizing Moment of Inertia for Spinnable Objects (supplementary material). *ACM Trans. Graph.* **33** (2014).

128. Cohen-Steiner, D., Alliez, P. & Desbrun, M. Variational shape approximation. *ACM Transactions on Graphics* **23**, 905 (2004).

129. *Dry Stone Walls: Basics, Construction, Significance* (ed Environmental Action Foundation) 472 (Scheidegger & Spiess, 2019).

130. Coumans, E. *Bullet Physics Simulation* in *ACM SIGGRAPH Courses* (Association for Computing Machinery, 2015), 1.

131. Cornut, O. *Dear ImGui: Bloat-Free Immediate Mode Graphical User Interface for C++ with Minimal Dependencies* https://github.com/ocornut/imgui. Accessed: 2021-04-15.

132. Jacobson, A., Panozzo, D., *et al. Libigl: A Simple C++ Geometry Processing Library* https://libigl.github.io/. Accessed: 2021-04-15.

133. Lieberman, Z., Watson, T., Castro, A. & openFrameworks Community. *openFrameworks* https://openframeworks.cc/. Accessed: 2021-04-15.

134. Blair, T. C. & McPherson, J. G. Grain-size and textural classification of coarse sedimentary particles. *Journal of Sedimentary Research* **69**, 6 (1999).

135. Cramb, I. *The Art of the Stonemason* 174 (Alan C. Hood & Co., Chamersburg, 1992).

136. Waechter, M., Moehrle, N. & Goesele, M. *Let There Be Color! Large-Scale Texturing of 3D Reconstructions* in *Computer Vision – ECCV* (eds Fleet, D., Pajdla, T., Schiele, B. & Tuytelaars, T.) (Springer International Publishing, 2014), 836.

137. Marchand, E., Uchiyama, H. & Spindler, F. Pose Estimation for Augmented Reality: A Hands-On Survey. *IEEE Transactions on Visualization and Computer Graphics* **22**, 2633 (2016).

138. Ioannidou, D., Zerbi, S. & Habert, G. When more is better – Comparative LCA of wall systems with stone. *Building and Environment* **82**, 628 (2014).

139. Villemus, B. *Etude des murs de soutènement en maçonnerie de pierres sèches* Doctoral Thesis, Civil Engineering (L'institut National Des Sciences Appliquees De Lyon, 2004), 247.

140. Mundell, C. *Large Scale Testing of Drystone Retaining Structures* Doctoral Thesis, Department of Architecture & Civil Engineering (University of Bath, 2009).

141. Snow, D. *Listening to Stone: Hardy Structures, Perilous Follies, and Other Tangles with Nature* (Artisan).

142. Witschi, F. in *Dry Stone Walls: Basics, Construction, Significance* (ed Environmental Action Foundation) 1st ed., 329 (Scheidegger & Spiess).

143. Bätzing, W. in *Dry Stone Walls: Basics, Construction, Significance* (ed Environmental Action Foundation) 1st ed., 379 (Scheidegger & Spiess).

144. Ioannidou, D., Zerbi, S., García de Soto, B. & Habert, G. Where does the money go? Economic flow analysis of construction projects. *Building Research & Information* **46**, 348 (2018).

145. Carron, A., Arcari, E., Wermelinger, M., Hewing, L., Hutter, M. & Zeilinger, M. N. Data-Driven Model Predictive Control for Trajectory Tracking With a Robotic Arm. *IEEE Robotics and Automation Letters* **4**, 3758 (2019).

146. Sleiman, J.-P., Carius, J., Grandia, R., Wermelinger, M. & Hutter, M. *Contact-Implicit Trajectory Optimization for Dynamic Object Manipulation* in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2019), 6814.

147. Egli, P. & Hutter, M. *Towards RL-Based Hydraulic Excavator Automation* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020).

148. Menges, A. in *Rob | Arch 2012* 28 (Springer, Vienna).

149. Johns, R. L. & Foley, N. *Bandsawn Bands* in *Robotic Fabrication in Architecture, Art and Design 2014* (eds McGee, W. & Ponce de Leon, M.) (Springer International Publishing, Cham, 2014), 17.

Curriculum Vitae

Martin Wermelinger

born August 23, 1990 in Switzerland

Education

2016 – 2021	Doctoral studies at the Robotic Systems Lab of Prof. M. Hutter at the Institute of Robotics and Intelligent Systems, Department of Mechanical and Process Engineering, ETH Zürich, Switzerland
2013 – 2015	Master of Science in Mechanical Engineering, ETH Zürich, Switzerland
2010 – 2013	Bachelor of Science in Mechanical Engineering, ETH Zürich, Switzerland

Work Experience

03/2016 – 06/2021	PhD Student and Research Assistant, Robotic Systems Lab, ETH Zürich, Switzerland
02/2017 – 02/2020	Lecturer *Programming for Robotics – ROS*, ETH Zürich, Switzerland
11/2015 – 03/2021	Research Assistant, Robotic Systems Lab, ETH Zürich, Switzerland
11/2014 – 01/2015	Intern, Helbling Technik AG, Zürich, Switzerland
09/2013 – 12/2013	Teaching Assistant *Technische Mechanik*, ETH Zürich, Switzerland
09/2012 – 12/2012	Teaching Assistant *Mechanik 3*, ETH Zürich, Switzerland

Publications

This section provides a list of all papers that were published or submitted during these doctoral studies. Please find an up-to-date list on Google Scholar.

Published articles in peer-review journals:

- Carius, J., Wermelinger, M., Rajasekaran, B., Holtmann, K. & Hutter, M., Deployment of an autonomous mobile manipulator at MBZIRC. *Journal of Field Robotics*, **35** 8, pp.1342-1357 (2018).
- Carron, A., Arcari, E., Wermelinger, M., Hewing, L., Hutter, M. & Zeilinger, M. N. Data-Driven Model Predictive Control for Trajectory Tracking With a Robotic Arm. *IEEE Robotics and Automation Letters* **4**, 3758 (2019).
- Johns, R. L., Wermelinger, M., Mascaro, R., Jud, D., Gramazio, F., Kohler, M., Chli, M. & Hutter, M. Autonomous Dry Stone: On-Site Planning and Assembly of Stone Walls with a Robotic Excavator. *Construction Robotics* **4** 3, pp.127-140 (2020).
- Mascaro*, R., Wermelinger*, M., Hutter, M., & Chli, M. Towards automating construction tasks: Large-scale object mapping, segmentation, and manipulation. *Journal of Field Robotics* **38** 5, pp.684-699 (2021).
- Wermelinger, M., Johns, R., Gramazio, F., Kohler, D., & Hutter, M. Grasping and Object Reorientation for Autonomous Construction of Stone Structures. *IEEE Robotics and Automation Letters* **6**, 5105, (2021).
- Jud, D., Kerscher, S., Wermelinger, M., Jelavic, E., Egli, P., Leemann, P., Hottiger, G. and Hutter, M. HEAP - The Autonomous Walking Excavator. *Automation in Construction* **129**, 103783, (2021).

(* contributed equally)

Relevant conference contributions:

- Furrer*, F., Wermelinger*, M., Yoshida*, H., Gramazio, F., Kohler, M., Siegwart, R., & Hutter, M. *Autonomous Robotic Stone Stacking with Online next Best Object Target Pose Planning* In *IEEE international conference on robotics and automation* (*ICRA*), pp. 2350-2356. IEEE, 2017.
- Wermelinger*, M., Furrer*, F., Yoshida*, H., Gramazio, F., Kohler, M., Siegwart, R., & Hutter, M. *Greedy Stone Tower Creations with a Robotic Arm* In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence* (*IJCAI-18*), pp. 5394-5398. Lawrence Erlbaum Associates, 2018.
- Carius, J., Wermelinger, M., Rajasekaran, B., Holtmann, K. and Hutter, M., *Autonomous Mission with a Mobile Manipulator – A Solution to the MBZIRC* In *Field and Service Robotics*, pp. 559-573, Springer, 2018.
- Sleiman, J.P., Carius, J., Grandia, R., Wermelinger, M. & Hutter, M., *Contact-Implicit Trajectory Optimization for Dynamic Object Manipulation*. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*), pp. 6814-6821, IEEE, 2019.

(* contributed equally)

Awards:

- *Best Student Paper Award* at ICRA 2017
  For the paper: *Autonomous Robotic Stone Stacking with Online next Best Object Target Pose Planning*.