Bachelor Thesis

**Czech
Technical
University
in Prague**

F3

**Faculty of Electrical Engineering
Department of Cybernetics**

Automatic kinematic calibration of dual arm manipulator using self-contact and plane constraints

Jakub Rozlivek

Acknowledgements

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 24. května 2019

Abstract

Robots are widely used in various modern manufacturing industry sectors. Their expansion ties up with the requirements of high precision. For that reason, the calibration and its accuracy became important. Traditional calibration procedures involve some form of external metrology systems, but they are costly. Advances in sensor technology make affordable but increasingly accurate devices such as cameras and force sensors available, making it possible to perform automated self-calibration relying on information from these sensors. In this work, we used an industrial robot platform having special end effectors with force sensing at the wrist, fiducial markers, and two external cameras mounted on the robot base to investigate robot kinematic calibration by employing different combinations of calibration approaches: self-observation, self-touch, and planar constraints. A comprehensive dataset has been collected for this purpose and made available to the community. We studied estimation of the end effector parameters as well as calibration of the complete kinematic chain (DH parameters) and we compared our results with ground truth provided by calibration using Leica absolute tracker.

Keywords: dual-arm robot, kinematic calibration, self-calibration, kinematic chains, force sensing, optimization, planar constraints

Supervisor: Mgr. Karla Štěpánová, Ph.D.

Abstrakt

Roboti jsou široce používáni v moderním výrobním průmyslu. Jejich rozšíření jde ruku v ruce s požadavky na vysokou přesnost. Z toho důvodu je kalibrace a její přenost tak důležitá. Tradiční kalibrační procedury zahrnují různé formy externích měřicích zařízení, ale ty jsou drahé. Díky technologickému pokroku se stávají dostupnými pokročilé senzory jako například kamery a silové sensory, které umožňují provádět automatickou sebekalibraci na základě informací z těchto senzorů. V této práci jsme využili průmyslovou robotickou platformu se speciálními koncovými články mající silové senzory na zápěstí, referenční značky a se dvěma externími foto-aparáty upevněnými na společné základně manipulátorů, abychom prozkoumali kinematickou kalibraci robota pomocí různých kombinací kalibračních přístupů - sebepo-zorování, sebedotyk a rovinná omezení. Pro tento účel byl nasbírán a dán volně k dispozici obsáhlý dataset. Studovali jsme odhad parametrů koncového článku stejně jako kalibraci celého kinematického řetězce (DH parametrů) a naše výsledky jsme porovnali s referenčními hodnotami získanými z kalibrace pomocí měřícího zařízení Leica.

Klíčová slova: dvouruký robot, sebekalibrace, kinematické řetězce, snímání síly, optimalizace, rovinná omezení

Překlad názvu: Automatická kinematická kalibrace dvojrukého manipulátoru za pomoci sebedoteku a rovinných omezení

Contents

Figures

Tables

Chapter 1

Introduction

1.1 Motivation

Robots are widely used in various modern manufacturing industry sectors. They replaced humans in repeatable or dangerous tasks. An extension to other sectors or advanced tasks ties up with the requirements of high precision. The robot dimensions can differ from the nominal ones during the years of use. They can be influenced by the stress during transport, deformations after collisions, etc. That is the reason why the calibration and its accuracy became important. The inaccurate calibration causes that the robot does not reach the desired point, but it is shifted by an offset.

However, external metrology systems are costly, e.g., the Leica metrology system used as ground truth in this work costs millions of Czech crowns. The price is too high for most of the companies working with not only industrial robots. There is a possibility to lend it from time to time as we did, but it still brings up problems, e.g., these systems are not able to find out the end effector length and orientation. Another approach is the calibration using cameras: it can estimate the end effector parameters, but the solution can be dependent on camera placement and calibration. One more possibility is the calibration using physical contact and the constraints thus generated. The touching place is usually known and taken as ground truth. We would like to expand the possibilities of robot calibration, so that we can choose the appropriate method depending on the environment, available sensors, etc.

1.2 Goals

The first and main goal is whether we are able to calibrate robot parameters using known or unknown planar constraints. We want to find out whether it is possible to calibrate the robot without the expensive systems using only the robot with its self-contact information, and we want to know the precision of this type of calibration compared to our chosen example of the metrology system – the Leica absolute tracker. The next goal is whether we are able to calibrate without self-observation using cameras and whether their position has to be precalibrated or not. We want to test the suitability of self-touch for the calibration – this can be useful for humanoid robots with artificial skin. We also want to find a suitable calibration method when the end effector is changed. Then we want to find a fast ("daily") calibration method which can be used before the usual tasks on cheap, less accurate, or flexible robots (e.g., some of the collaborative robots or service robots) which are more prone to decalibration. Another goal is the examination if the methods can deal with perturbation of the initial robot model.

▪ **1.3 Related work**

Since our robotic platform has two arms and two cameras similar to eyes, it resembles a humanoid robot, so that we mainly focus on humanoid-like setups that offer rich possibilities for automated self-contained calibration. Most often, the calibration loop is closed through self-observation of the end effector using cameras located in the robot head (open-loop calibration method per [2]). Another family of approaches exploits some form of physical contact of the end effector with the environment, such as fixing the end effector to the ground [3] or more complex setups [4, 5], they all require force sensor or something like that on the part of manipulator.

"Self-touch" constitutes a specific, less common, way of kinematic loop closure. Roncone et al. [6] showed the iCub robot performing autonomous self-touch using a finger on the contralateral arm; Li et al. [7] employed a dual KUKA arm setup with a sensorized "finger" and a tactile array on the other manipulator. Forward kinematics together with skin calibration provide contact position that can then be used for calibration. Since the skin provides a pose measurement rather than constraint, this may fall under open-loop calibration. In this way, one arm of a humanoid can be used to calibrate the other. Khusainov et al. [8] exploit this but using an industrial manipulator to calibrate the legs of a humanoid robot.

Štěpánová and Hoffmann [9] provide a synthetic experiment with self-calibration of a simulated robot based on a real humanoid robot, iCub. The contact is provided by index finger touching a palm of the other hand, thus closing a kinematic loop, while also observing the points with both of its eyes, providing measurements. The calibration problem is approached as an open-loop calibration, considering the touching constraint as a measurement of relative position of the two points that always reports zero displacement. Štěpánová and Hoffmann evaluate the quality of parameters estimation using joint calibration of multiple kinematic chains while utilizing either only the contact information, only the cameras, or both—contact constraint and cameras.

We may recognize pair-wise procedural (sequential) calibration, where a series of calibrations is carried out with a single kinematic chain or sensor being calibrated at a time, and joint calibration, where all sensors or kinematic chains are calibrated using unified error function and optimization.

Birbach et al. [10] evaluate the advantages of joint calibration on multiple sensors (a pair of cameras, a Microsoft Kinect RGBD sensor and an inertial measurement unit) on a humanoid robot. With this method, no external measuring apparatus is needed; instead, internal sensors on the robot are used for calibration. Birbach et al. formulate an error function as a weighted sum of squares over the errors of individual sensors and use Levenberg-Marquardt algorithm to optimize the intrinsic parameters of individual sensors and their position with relation to the robot head. They claim that joint calibration is more efficient than pair-wise procedural calibration, because in the case of pair-wise procedural calibration, inconsistencies in the obtained calibration results may occur, while joint calibration ensures consistent result.

Kinematic calibration using plane constraints was explored by Ikits and Hollerbach [11]

who stated that "Calibration using a planar constraint is the most significant remaining closed-loop approach to be developed." They proposed a new approach of closed-loop calibration using only one endpoint constraint and evaluated primarily in simulation of Puma robot. Zhuang et al. [12] explored multiple variations of plane constraints as well as the option with/without known plane parameters and demonstrate their results also on a PUMA 560 robot. In particular, they show that a single-plane constraint does not necessarily guarantee that all kinematic parameters of the robot will be observable. On the other hand, multiple-plane constraint should be a remedy to this problem and they show that data collected from 3 planes constraints could be equivalent to the data collected from a point measurement device in the case that 1) all three planes are mutually non-parallel; 2) the identification Jacobian of the unconstrained system is nonsingular; and 3) measured points from each individual plane do not lie on a line on that plane.

Joubair at al.[13] show how closed-loop calibration approach using multi-planar constraints can be utilized to significantly improve accuracy of calibration of six-axis serial robot. Zenha et al. [14] provided touch events on multiple planar surfaces having a priori known plane parameters with a simulated iCub robot. From their work, it follows that making the robot explore space by contacts on various surfaces is more effective in reducing its model inaccuracies than a single surface scenario.

Specifically related to the setup used in this work, The CloPeMa robot setup has been previously calibrated using two different methods: measuring Machine (RedCaM) by Beneš et al. [15] and Leica laser tracker by Beneš et al [16]. Petrík and Smutný [1] review the precision of these methods using a linear probe sensor. We will compare our empirical results with theirs. Based on a dataset of 43 different poses with touching end effectors, they calculate the mean error as 0.67 (range 2.92) mm on CAD model, 0.54 (range 2.55) mm on Leica based calibration and 2.45 (range 9.92) mm on RedCaM based calibration.

1.4 Contribution

This thesis directly follows up on [17], continuing to investigate the kinematic calibration of a dual arm Yaskawa Motoman robot platform. The work of Puciow [17] was extended in these main aspects: (1) we modified the experimental setup (end effectors were changed for shorter ones, new cameras were installed and calibrated); (2) we added option to optimize using contact with known or unknown planar constraints; (3) we prepared measurement by an external metrology system – Leica absolute tracker; (4) we extended the data collecting application to collect datasets by horizontal plane touch experiment, vertical plane touch experiment, and the dataset for the calibration by Leica. We collected and published a comprehensive dataset of different setups including end effectors positions and orientations, joints configurations, Leica measurements, photos from XTion and 2 Nikon cameras, end effector contact information during self-touch or touch with the plane constraint. The dataset is available for download in our Google drive [18].

The original experiments using self-touch and self-observation [17] were rerun in the new setup. In addition, new experiments using planar constraints and the Leica tracker were designed and performed. The thus arising new calibration problems were formulated and optimized. We evaluated the calibration by Leica and compared the results with the

nominal DH parameters. We performed calibrations using planar constraints, self-touch, self-observation and their combinations for end effector parameters first, then for all DH parameters of the robot arm as well. We studied if and how the DH parameters are immune to perturbation in different calibration setups. Additionally, we looked into the observability and the identifiability of individual optimized parameters.

In summary, this work presents a unique contribution over [17] and the state-of-the-art in general in that it compares several kinematic calibration methods on a single industrial platform. The methodology draws on [9], which is, however, a simulation study only and with self-touch and self-observation only.

Chapter 2

Materials and methods

2.1 Experimental setup

The experimental setup shown in Figure 2.1 was taken from [17]. The platform consists of two Yaskawa Motoman MA1400 manipulators installed on a common base – a Yaskawa R750 robotic turntable. The special end effectors were connected to manipulators. They were covered by fiducial markers and have force sensors at the wrist. Changes made compared to the previous setup are described in the following sections.
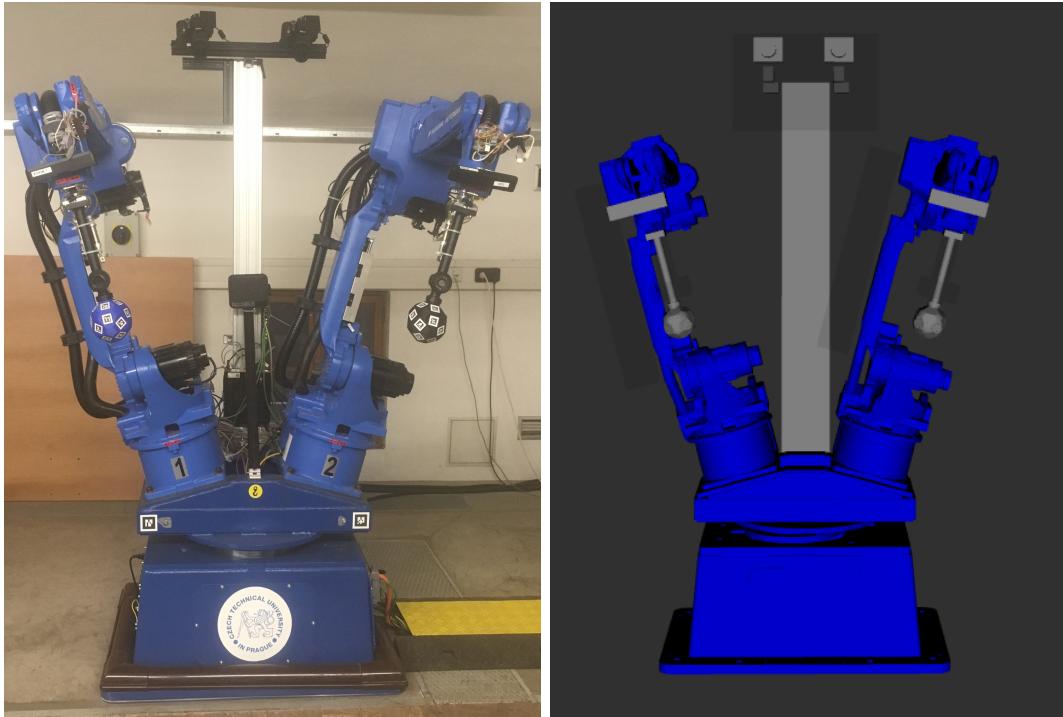
Figure 2.1: The real robot setup (left) with the visualisation (right).

2.1.1 End effector description

The end effectors were changed because the previous ones were prone to oscillations. The new ones preserved the truncated icosahedrons with spherical tiles in place of the pentagonal faces and with ArUco markers located in the hexagonal faces. The ArUco markers give us the self-observation information about the end effector location. The spherical tiles leverage that their distance from the icosahedron center is always equal to the radius of

the icosahedron, so they are used for touching. The icosahedron lies in the axis of the last joint, and its center is 354 mm far from the manipulator tip.

The change compared to the previous end effector [17] is that the bracket was omitted and the beam was extended and made from lead. The total length of the end effector is shorter by approximately 17 centimeters. The reference frame of the end effector lies in the center of the end effector. The contact points on the end effector during the experiments are five spherical tiles in the bottom of both icosahedrons (see Figure 2.2).
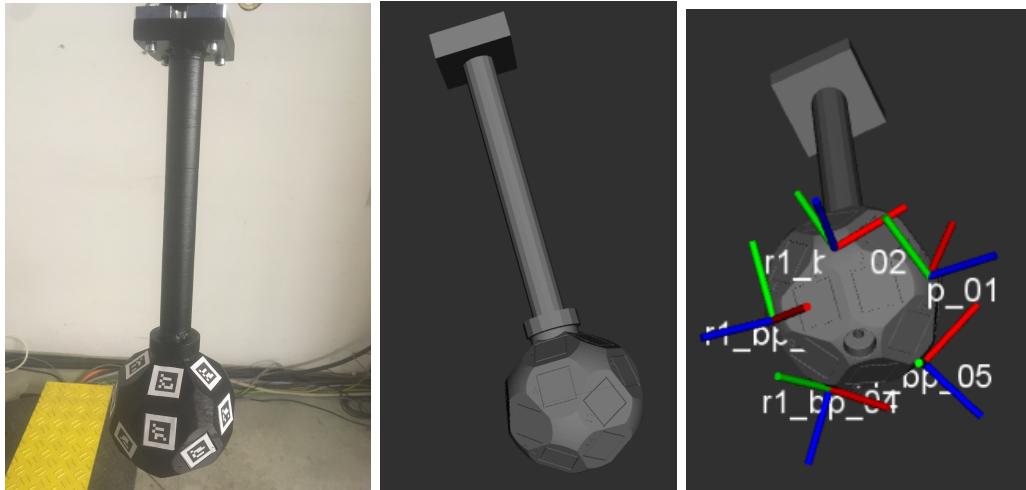
Figure 2.2: New end effector (left), its model (center) and spherical tiles used for touches (right).

■ 2.1.2 Cameras

Cameras Nikon D5300 with new lens AF-P DX NIKKOR 18-55mm f/3.5-5.6G were installed in a side-by-side configuration on top of a vertical beam connected to the mounting adapter, as in [17]. The maximum image size is 6000x4000 Px (24 MPx camera resolution). We set a camera zoom to cover all end effector touches with table and touches of two end effectors. The cameras are not able to keep the settings of focus after a restart. That is the most visible difference between old lenses and the new ones. So we created a python script (*config_utility.py*) to find the best focus value for each camera. The script can be found in [19] in folder *CameraConfig*. These values are used by a focus stepper motor to set the same focus every time. Optimal focus values (the pulse length to the focus ring motor to cause a relative movement) are 960 [20] for both cameras.

■ 2.1.3 ArUco markers

As in [17], ArUco markers were used for pose estimation of the end effector. According to [21], ArUco markers are square markers composed of an outside black frame and an inner black and white matrix, containing the marker identification. The marker size determines the size of the internal matrix. They are used to detect a chosen object and obtain its position and orientation related to the camera.

We use OpenCV Aruco module to detect the markers in the camera images, allowing markers of various sizes to be used. A set of markers that are considered in a specific application is called dictionary – the list of binary codifications of each of its markers. Every marker known to the dictionary has a unique ID number - the marker index inside the dictionary. The OpenCV aruco module contains some predefined dictionaries covering a range of different dictionary sizes and marker sizes. We used a dictionary with 1024 distinct markers having a marker size 5×5 called "DICT_ARUCO_ORIGINAL".

Faces on both icosahedrons have the same range of numbers from 1 to 20. So we used one more digit ahead of these two to distinct the icosahedrons. These three digits constitute icosahedron face ID. IDs are from 101 to 120 on the right icosahedron and 201 to 220 on the left one, respectively.

∎ 2.1.4 Robot dimensions

The base reference frame definition is — z-axis is identical with the rotation axis of the turntable, x-axis is in the direction from the right to left arm and y-axis makes a right-hand coordinate system with x and z. The Denavit-Hartenberg (DH in what follows) convention was used to describe the parameters of the manipulator.
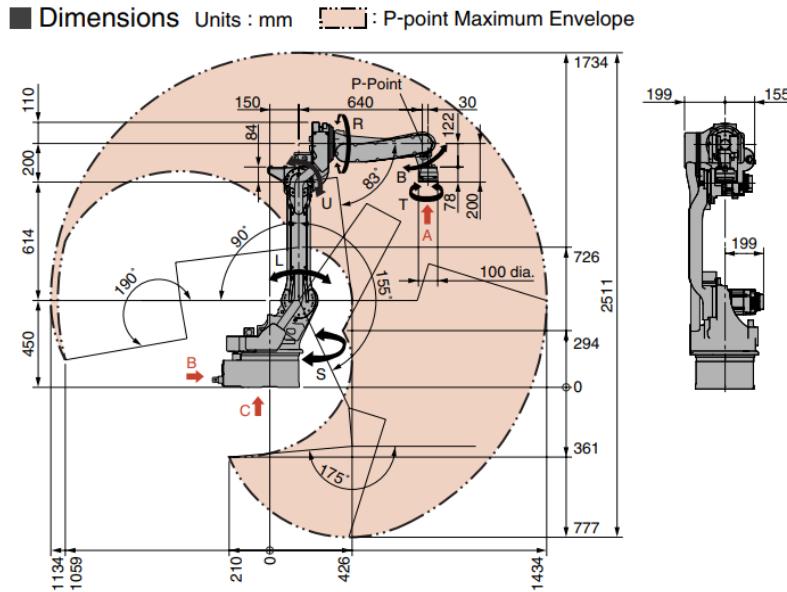
Figure 2.3: Dimensions and workspace of Motoman MA1400 taken from [22].

Figure 2.3 shows the nominal dimensions of the MA1400 manipulator. These nominal dimensions were used to get the DH parameters of the manipulator. In the previous work [17], it was decided to represent the mounting of the manipulators as another link in the DH parameters description of both manipulators. Because of that, two new virtual links were created so that the point, where the transformation would place the new reference frame, can be reached with given orientation. Then another set of DH parameters was added to represent the transformation from the tip link to the end effector. These DH parameters were changed compared to [17] because the new end effectors were installed

(see Section 2.1.1). The camera poses were converted into DH parameters as well and the values can be found in Table 2.1. The gained DH parameters are used as initial values for the calibration.

		Camera 1					Camera 2		
i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]	i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
RC1	0.1602	1.9042	-2.4396	-3.0171	LC1	0.1602	1.9042	2.4396	-0.1245
RC2	0	-0.5118	0	0.0953	LC2	0	-0.5118	0	3.0463

Table 2.1: The DH parameters of camera chains, the gray parameters were not calibrated unless otherwise stated.

Table 2.2 shows the complete DH parameter description of both arms. As can be seen, after merging with the transformation matrix, the turntable joint is different for each manipulator, because the rotation axes of the first joints are different. Both joints, however, share the same joint coordinate.

		Manipulator 1 (right arm)					Manipulator 2 (left arm)		
i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]	i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
TT1	0	-0.2630	$\pi/12$	$-\pi/2$	TT2	0	-0.2630	$\pi/12$	$\pi/2$
S1	0.15	1.4159	$-\pi/2$	0	S2	0.15	1.4159	$-\pi/2$	π
L1	0.614	0	π	$-\pi/2$	L2	0.614	0	π	$-\pi/2$
U1	0.2	0	$-\pi/2$	0	U2	0.2	0	$-\pi/2$	0
R1	0	-0.64	$\pi/2$	0	R2	0	-0.64	$\pi/2$	0
B1	0.03	0	$\pi/2$	$-\pi/2$	B2	0.03	0	$\pi/2$	$-\pi/2$
T1	0	0.2	0	0	T2	0	0.2	0	0
EE1	0	0.354	0	0	EE2	0	0.354	0	0

Table 2.2: The complete DH parameter description of both arm kinematic chains, the gray parameters were not calibrated unless otherwise stated.

As was already mentioned, both robot arms are represented by individual kinematic chains described by DH parameter convention. Both kinematic chains start in the base frame (BF) and the first joint - the turntable joint is reached by the identity transformation matrix. Both chains have a common rotation of the first joint and mounting of each manipulator on the turntable is different, so we divided turntable into two separated links. For that reason, the turntable DH parameters are not the same. Both arms have six driven joints, marked with letters "S", "L", "U", "R", "B" and "T". Joint S connects the turntable (TT) with link S of the robot, and the link T is connected with the end effector link (EE). The mentioned end effector link was added to the end of each kinematic chain, and it represents the transformation to the center of the icosahedron, our end effector. It is connected only by a virtual joint.

Two other kinematic chains are containing only the turntable link (C1) and a camera link (C2). So the turntable is divided into not two but four links. The camera links are connected to turntable links by a virtual joint.
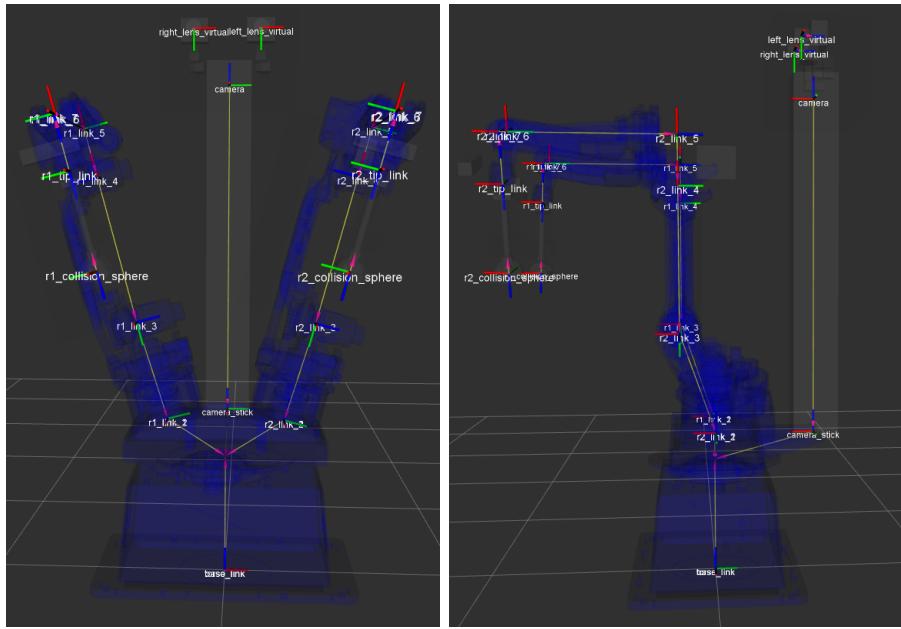
Figure 2.4: Location of robot reference frames.

2.1.5 Leica absolute tracker

We decided to choose a coordinate measuring machine as our reference - ground truth. That machine was a Leica Absolute Tracker AT402 (Leica tracker in what follows) from Leica Geosystems. Leica tracker is a portable 3D absolute tracker for fully guided measurement processes [23]. The absolute distance meter has a resolution of 0.1 microns and a maximum uncertainty of 10 microns over a full 160 m radial volume. The maximum uncertainty is known as maximum permissible error, which is defined in the ISO-VIM [24] as "extreme value of measurement error, with respect to a known reference quantity value, permitted by specifications or regulations for a given measurement, measuring instrument, or measuring system". It means all measured values are guaranteed to be below.

According to [25] the tracker sends a modulated and polarised infrared light beam to a retro-reflective target held against the object to be measured. A spherically mounted retro-reflector is used as a retro-reflective target. Light reflected off the target retraces its path, re-entering the tracker at the same position it left. As light re-enters the tracker, it goes to an absolute distance meter that measures the distance from the tracker to the reflector. The laser tracker contains two angular encoders for measuring the angular orientation of the tracker's two mechanical axes. The center of the retro-reflector then can be precisely calculated from these two angles and the measured distance. Because of the spherical design, the center of the retro-reflector is always at a fixed offset distance to any surface being measured, so the coordinates of surfaces are easily obtained. The tracker collects 3D (only position, not orientation) coordinate data displayed within a coordinate system tied to the tracker. The advantage of an absolute distance measurement used by Leica tracker over incremental distance measurement is the ability simply to point the beam at the target and shoot, and the distance to the target is measured automatically, even if the beam has previously been broken.

The Leica tracker was situated outside the robot workspace to secure the minimal distance from retro-reflector (1.5 m). The retro-reflector has to be in a magnetic holder where it stays motionlessly. We attached the holder to the end effector beam approximately 25 cm from the last joint. The holder had to be added to the collision model as well. Figure 2.5 shows the holder on the beam and its collision model created by a block. Since the retro-reflector worked as end effector in a robot calibration by Leica tracker, the end effector link (LR) represents the transformation to the retro-reflector. The position and orientation of the retro-reflector were measured and converted to DH parameters, which can be found in Table 2.3.
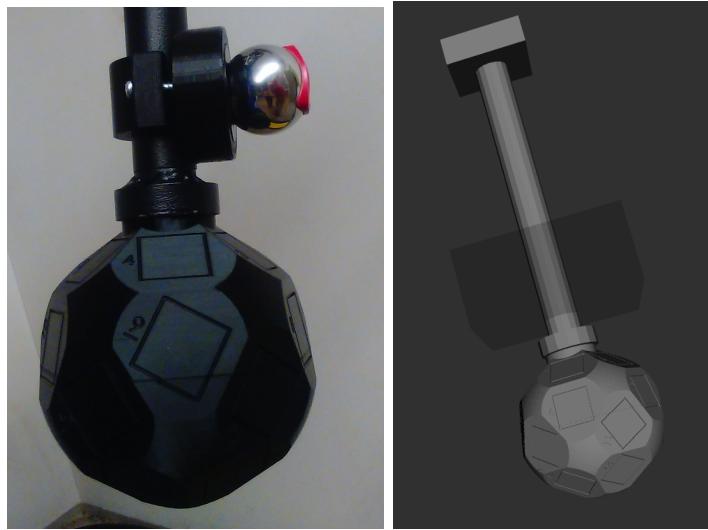
Figure 2.5: Leica ball reflector on the holder (left) and its simple collision model (right).

	Leica - right hand					Leica - left hand			
i	a_i **[m]**	d_i **[m]**	α_i **[rad]**	θ_i **[rad]**	i	a_i **[m]**	d_i **[m]**	α_i **[rad]**	θ_i **[rad]**
LR1	0.02	0.25	0	$\pi/2$	LR2	0.02	0.25	0	$\pi/2$

Table 2.3: The DH parameters of Leica retro-reflector, the gray parameters were not calibrated unless otherwise stated.

2.2 Camera calibration

It was necessary to perform calibration of camera intrinsic parameters. This was preceded by their focusing. At first, a lens was set to autofocus, and a focus motor was set to a zero value. Then the focus value was changed to the optimal value, and the stepper motor set a requested focus for the camera. After focusing the lens focus mode was changed to *manual*, not to alter intrinsic parameters while using. The same approach was applied to both cameras.

The calibration of intrinsic parameters was carried out with a calibration dot pattern. The dataset used for calibration was composed of 22 pattern images. Each of the captured images had a different position and rotation to achieve better calibration results. After a

left camera was calibrated, the same process was done with the right camera. New camera matrices K and the vectors of distortion coefficients $\vec{d}$ are defined as

$$
K_R = \begin{bmatrix} 8185.397 & 0.000 & 2009.318 \\ 0.000 & 8170.401 & 2963.960 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}, K_L = \begin{bmatrix} 8110.478 & 0.000 & 1949.921 \\ 0.000 & 8098.218 & 2991.727 \\ 0.000 & 0.000 & 1.000 \end{bmatrix},
$$

$$
\vec{d_R} = \begin{bmatrix} -0.021 & -0.206 & -0.002 & -0.001 & 0.719 \end{bmatrix}^T,
$$

$$
\vec{d_L} = \begin{bmatrix} -0.023 & -0.213 & -0.001 & -0.001 & 0.662 \end{bmatrix}^T,
$$

the distortion coefficients are sorted as they are returned from OpenCV camera calibration functions [26].

2.3 Forward kinematics

The robot DH parameters from 2.1.4 are used for forward kinematic transformation. The transformation in each joint is described by 4 parameters a_i, d_i, α_i, θ_i. Transformation matrix T_i^{i-1} from link $i-1$ to link i can be computed by multiplying two sub-transformation matrices.

$$
T_{id}^{i-1} = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i) & 0 & 0 \\ sin(\theta_i) & cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_i^{i\alpha} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & cos(\alpha_i) & -sin(\alpha_i) & 0 \\ 0 & sin(\alpha_i) & cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}
$$

The resulting transformation matrix is

$$
T_i^{i-1} = T_{int}^{i-1} T_i^{int} = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i)cos(\alpha_i) & sin(\theta_i)sin(\alpha_i) & a_icos(\theta_i) \\ sin(\theta_i) & cos(\theta_i)cos(\alpha_i) & -cos(\theta_i)sin(\alpha_i) & a_isin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}
$$

The transformations between frames are the same as in [17]. The transformation from the icosahedron center (EE) to the base frame is obtained as:

$$
T_{EE1}^{BF} = T_{TT1}^{BF} T_{S1}^{TT1} T_{L1}^{S1} T_{U1}^{L1} T_{R1}^{U1} T_{B1}^{R1} T_{T1}^{B1} T_{EE1}^{T1} \tag{2.3}
$$

for the right arm and

$$
T_{EE2}^{BF} = T_{TT2}^{BF} T_{S2}^{TT2} T_{L2}^{S2} T_{U2}^{L2} T_{R2}^{U2} T_{B2}^{R2} T_{T2}^{B2} T_{EE2}^{T2} \tag{2.4}
$$

for the left arm. The transformations from the cameras to the base frame are

$$
T_{camera1}^{BF} = T_{TT3}^{BF} T_{camera1}^{TT3} \quad \text{and} \quad T_{camera2}^{BF} = T_{TT4}^{BF} T_{camera2}^{TT4} \tag{2.5}
$$

for the right and left camera, respectively. Using transformation T_{EE1}^{BF} or T_{EE2}^{BF}, we can transform a marker M_i from the icosahedrons to the base frame:

$$
T_{Mi}^{BF} = T_{EE1}^{BF} \cdot T_{Mi}^{EE1} \quad \text{and} \quad T_{Mi}^{BF} = T_{EE2}^{BF} \cdot T_{Mi}^{EE2} \tag{2.6}
$$

for the right and left icosahedron, respectively. Positions of the markers in the icosahedron frame are fixed. Now we can transform the marker to the camera frame using $T_{BF}^{camera1}$ or $T_{BF}^{camera2}$:

$$
T_{Mi}^{camera1} = T_{BF}^{camera1} \cdot T_{Mi}^{BF} \quad \text{and} \quad T_{Mi}^{camera2} = T_{BF}^{camera2} \cdot T_{Mi}^{BF}. \tag{2.7}
$$

2.4 Optimization problem formulation

We consider the open-chain calibration as an estimation of a parameter vector

$$\phi = \{[a_1, ..., a_n], [d_1, ..., d_n], [\alpha_1, ..., \alpha_n], [o_1, ..., o_n]\}$$

with $i \in N$, where $N = \{1, .., n\}$ is a set of indices identifying individual links; a_i, d_i and α_i are the first three parameters of the DH formulation of the link i and o_i is the offset of θ_i, last DH parameter. The estimation depends on the dataset D defined as

$$D = \{[\xi_1, ..., \xi_M], [\vec{z}_1, ..., \vec{z}_M], [\theta_{1,1}, ..., \theta_{1,N}, \theta_{2,1}, ..., \theta_{M,N}]\}, \tag{2.8}$$

where $j \in 1..M$ is an index of a particular dataset point, M is the overall number of points in the dataset, ξ_j is information about the place and the sensor of the measurement, $\vec{z}_j$ is one particular measurement vector and $\theta_{j,i}$ is the DH parameter of the link i in the configuration in which the measurement $\vec{z}_j$ was taken. All other necessary parameters for estimation such as camera calibration, fixed transformations to markers, fixed DH parameters or other properties of the robot are supposed as constant and denoted as ζ. We often estimate a subset of DH parameters only, assuming that the others are known, so our calibration enables us to treat a single DH parameter as fixed ($\in \zeta$) or to be optimized ($\in \phi$).

The calibration by minimizing the objective function $f(\phi, D, \zeta)$ can be formulated as

$$\phi^* = \arg\min_{\phi} f(\phi, D, \zeta), \tag{2.9}$$

$$f(\phi, D, \zeta) = \|g(\phi, D, \zeta)\|^2 = \sum_{i=1}^{M} g(\phi, D_i, \zeta)^2, \tag{2.10}$$

where M is the number of robot configurations and corresponding end effector positions used for calibration.

The notation is taken from [17].

2.4.1 Calibration using Leica tracker

The objective function is formulated as the error of distances:

$$\vec{g}_l(\phi, D, \zeta) = \left[\vec{p}(\phi, D_1, \zeta), ..., \vec{p}(\phi, D_M, \zeta)\right], \tag{2.11}$$

where the function $p(\phi, D_i, \zeta)$ computes the distance of the transformed point from Leica tracker and the point gained from the forward kinematics in the configuration given by the dataset point D_i, where $i \in B$, $B \subset A$ is a subset of the dataset points where the pose of the robot is unique.

As can be seen, we need to compare two different sets of 3D points. Set $\vec{p_1}$ contains points from Leica tracker in its coordinate system and set $\vec{p_2}$ contains points in base coordinate system computed from a joint configuration using forward kinematics. The relation between corresponding points in sets is generally

$$p_{2,i} = Rp_{1,i} + T + N_i. \tag{2.12}$$

We used an algorithm for finding least-squares solution of R and T [27]. It is an non-iterative algorithm using the singular value decomposition of a $3{\times}3$ matrix. First step of algorithm is to calculate centers for both sets as

$$p_c = \frac{1}{N}\sum_{i=1}^{N} p_i. \tag{2.13}$$

Then $\vec{q}$ is computed from the center as

$$q_i = p_i - p_c, \tag{2.14}$$

from these points the matrix H is gained as

$$H = \sum_{i=1}^{N} q_i q_i^T. \tag{2.15}$$

Then the singular value decomposition is calculated

$$H = USV^T \tag{2.16}$$

Next step is computing the matrix X as

$$X = VU^T \tag{2.17}$$

Afterwards the determinant of X is checked. If det(X) is 1 then R = X, on the other hand if det(X) is -1, the algorithm fails. Now we are able to get the translation vector T from Equation 2.12 using centroids p_{1c} and p_{2c} as

$$T = p_{2c} - Rp_{1c}. \tag{2.18}$$

Once we have both R and T, we transform points from Leica tracker using Equation 2.12. These transformed points are used in our objective function.

■ 2.4.2 Calibration with planar constraints

For this optimization problem where planar constraints are used, we formulated the objective function as the error distances between contacts and a single or multiple fitted planes:

$$\vec{g_p}(\phi, D, \zeta) = \left[\vec{C_1}(\phi, D_1, \zeta),\ \vec{C_2}(\phi, D_2, \zeta),\ \vec{C_3}(\phi, D_3, \zeta) \right], \tag{2.19}$$

where $\vec{C_j}(\phi, D_j, \zeta)$ is a N-element vector of the error of distances between touches from a j-th dataset and a j-th estimated plane. The ϕ denotes the parameter vector defined in Equation 2.4, D_i indicates the part of dataset defined in Equation 2.8 and ζ wraps up all other necessary parameters.

The plane distance error is defined as

$$(\boldsymbol{\rho}^*, d^*, \boldsymbol{\phi}^*) = \arg\min_{\boldsymbol{\rho}, d, \boldsymbol{\phi}} \sum_{m=1}^{N} ||\boldsymbol{n}\boldsymbol{p}_m(\boldsymbol{\phi}) + d||^2, \tag{2.20}$$

where N is a number of measured points, ϕ is the estimated parameter vector corresponding to the kinematic parameters of the touching arm, $n = [a\ b\ c]$ is the plane normal, and d is the distance of the plane from the origin. The plane parameters (ρ and d) are estimated at each iteration of the optimization process based on current point coordinates estimates using singular value decomposition or least squares method. The point p_m is given by forward kinematics from dataset point $D_{i,m}$, where $m \in B$, $B \subset A$ is a subset of the dataset points where the pose of the robot is unique. Without loss of generality, we assume that the contacts were provided by the end effectors centers instead of the spherical tiles.

■ Fitting plane to measured points

An approach we selected assumes that the points satisfy the general plane equation

$$ax + by + cz + d = 0, \tag{2.21}$$

where (a, b, c) is a plane normal vector. There are two implemented ways to compute a plane equation - singular value decomposition (SVD) and least squares fitting (LSQ) [28].

Singular value decomposition of matrix A is defined by an equation

$$A = USV^T, \tag{2.22}$$

where S is a diagonal matrix having singular values of matrix A on its diagonal, U and V are orthogonal and contain left and right singular vectors of matrix A, respectively.

The points are converted to homogeneous coordinates for SVD. Then a points center is estimated, and all points are centered. Then Matlab function svd is called. The singular vector corresponding to the least singular value is set as a normal of the plane. Parameter d in Equation 2.21 is calculated from

$$d = -ax_0 - by_0 - cz_0, \tag{2.23}$$

where (x_0, y_0, z_0) are coordinates of the points center.

The other possibility is the least squares method. We want to minimize a vertical distance from the plane using this method. Equation 2.21 can be written as

$$-cz = ax + by + d, \tag{2.24}$$

without loss of generality, assume that $c = -1$, then

$$z = ax + by + d. \tag{2.25}$$

Matrix form of this equation is:

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ ... & ... & ... \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} z_0 \\ z_1 \\ ... \\ z_n \end{bmatrix}, \tag{2.26}$$

$$P \begin{bmatrix} a \\ b \\ d \end{bmatrix} = Q. \tag{2.27}$$

A system of equations is over-determined because there are more than three points in the dataset, so left pseudo-inverse should be used to solve the system.

$$\begin{bmatrix} a \\ b \\ d \end{bmatrix} = (P^T P)^{-1} P^T Q \qquad (2.28)$$

A plane is computed in a new function *getPlane*, this function can be found in [19] in folder *matlab_models/motoman-new-calib/model_fx*. Joints angles from the dataset are taken, and the positions of the points are computed from forward kinematic transformation matrices.

Since fitting a vertical plane to points using least squares method does not work properly (see Figure 2.6), we chose to compute planes using SVD in the whole program.
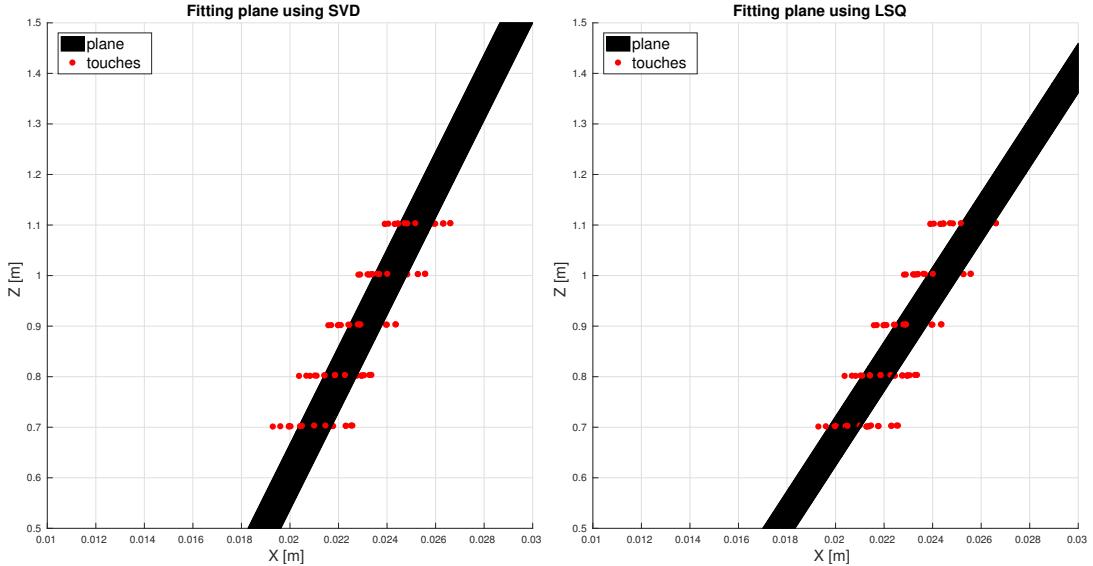
Figure 2.6: Comparison of fitting plane methods - SVD (left) and LSQ (right).

2.4.3 Other calibration approaches

These optimization problems are defined in [17], therefore only brief descriptions are provided below with the equations rewritten from [17].

Calibration using self-observation

The objective function is formulated as the distance between projected markers and their pixel coordinates in taken images:

$$\vec{g_{so}}(\phi, D, \zeta) = \left[\vec{P}(\phi, D_1, \zeta) - \vec{z}(D_1), ..., \ \vec{P}(\phi, D_M, \zeta) - \vec{z}(D_M) \right], \qquad (2.29)$$

where $\vec{P}(\phi, D_i, \zeta)$ projects the marker from dataset point D_i where $i \in \{1, ..., M\}$, M is the length of the dataset D. The $\vec{z}(D_i)$ is the actual marker position in the camera image.

15

We applied the calibrated camera model in order to obtain projected marker positions in each of the robot cameras. First, we need to obtain the marker position in camera frame:

$$
\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = T_{Mi}^{camera1} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},
$$

where $[x_c, y_c, z_c, 1]^T$ are the homogeneous coordinates of marker position in the camera frame. From this point, we are able to compute the 2-D point projection to the camera plane. We applied a standard pinhole camera model on this point and transformed the 3D point in camera frame ($[x_c, y_c, z_c]$) into image coordinates $[u, v]$ (the 2D plane of the camera):

$$
\begin{aligned}
x_c' &= x_c/z_c \\
y_c' &= y_c/z_c \\
r &= \sqrt{x'^2 + y'^2} \\
x_c'' &= x_c'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x_c' y_c' + p_2(r^2 + 2x_c'^2) \\
y_c'' &= y_c'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y_c'^2) + 2p_2 x_c' y_c' \\
\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= K \begin{bmatrix} x_c'' \\ y_c'' \\ 1 \end{bmatrix}
\end{aligned}
\tag{2.30}
$$

where K, k_1, k_2, k_3 and p_1, p_2 are camera intrinsic parameters obtained by camera calibration described in Section 2.2.

The actual marker position means the center of ArUco marker. The OpenCV function provides only the coordinates of all four marker corners with the distinction of whole pixels. The center is calculated as it was in [17], where the assumption that the center lies on the intersection of both diagonals was used. The error made by it is less than the distinction of OpenCV function [17].

■ Calibration using self-touch

The objective function is formulated as the error of distances of touching end effectors:

$$
\vec{g_{st}}(\phi, D, \zeta) = [c(\phi, D_1, \zeta) - q(\zeta), ..., \ c(\phi, D_K, \zeta) - q(\zeta)]
\tag{2.31}
$$

where the function $c(\phi, D_i, \zeta)$ computes the distance of the icosahedron centers in the configuration given by the dataset point D_i, where $i \in B$, $B \subset A$ is a subset of the dataset points where the pose of the robot is unique. The $q(\zeta)$ is the distance of the icosahedrons centers. This distance, marked as q, is equal to one diameter, because both icosahedrons have identical shape. Due to certain reasons [17] the icosahedron diameter of 116 mm is taken as a ground truth. Nevertheless, our optimization program is prepared to work with $q \in \phi$.

16
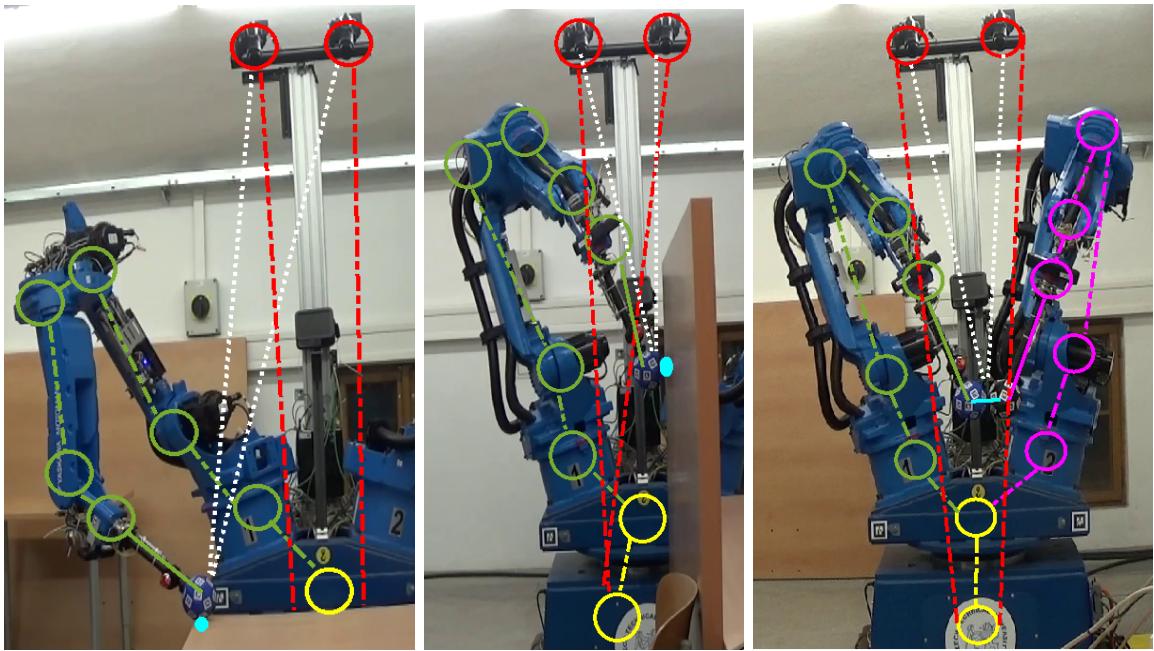
Figure 2.7: Illustration of calibration using different chains and constraints — horizontal plane contact (left), vertical plane contact (center), self-touch (right). All chains originate in a common Root (bottom yellow circle). The left and right arm chains are drawn in purple and green respectively. The eye chains are drawn in red. White lines denote projection into the cameras. The cyan marks indicate the distance between end effectors centers (one diameter) or between the end effector and a plane (zero).

■ 2.4.4   Combination of individual objective functions

It is possible to put different optimization problems together and minimize them at one time. Then the overall objective function is generally defined as

$$\vec{g}(\phi, D, \zeta) = \left[\vec{g_{so}}(\phi, D, \zeta), k \cdot \vec{g_{st}}(\phi, D_{st}, \zeta), k \cdot \vec{g_p}(\phi, D_p, \zeta), k \cdot \vec{g_l}(\phi, D, \zeta) \right], \qquad (2.32)$$

where datasets for self-touch D_{st} differ from datasets for planar constraints D_p and k is the scale factor, to compare the distance errors given in meters with the projection errors in pixels. The value of k was set in the same way as it was in [17]. The average distance between markers and cameras is approximately 2 m. Since the focal length is about 8000 pixels for both cameras, a 1 m displacement causes a 4000 px displacement in the image. Also, there is more than one marker for each pose, it means we should increase a weight of the touching and Leica errors. We assume that there are 20 markers for each pose. For that reason, the scale factor k has a value of $20 \cdot 4000 = 80000$. Now we have a converted distance unit from meters to an equivalent of pixels so that we can optimize them all at once. Figure 2.7 shows connections of different calibration chains and constraints.

■ 2.5   Non-linear least squares optimization

We used Matlab Optimization Toolbox for solving our optimization problem. This toolbox offers ready-to-use optimization function *lsqnonlin* a nonlinear least-squares solver. As this

work follows on from [17], for solving the calibration task we have chosen to use the same method – Levenberg-Marquardt algorithm instead of the default Trust-Region-Reflective algorithm, furthermore, Birbach et al. [10] use it in their work, and it is only an improved version of Gauss-Newton algorithm proposed by Hollerbach et al. [2].

According to [28] Levenberg-Marquardt algorithm is a numerical optimization method used for solving non-linear least-squares problems. It belongs to iterative line descent algorithms. These algorithms find a free local minimum of an objective function from an initial point using a variable direction and a variable step size α.

The Levenberg-Marquardt algorithm interpolates between the Gauss–Newton algorithm and the method of gradient descent. Gradient descent method uses negative function gradient at the current point as a descending direction to find a local minimum [28]. The main advantage of this method is its reliability because the direction always descends, but on the other hand, it suffers a drawback of slow convergence. Iteration step for this method is

$$\vec{x}_{k+1} = \vec{x}_k + \alpha_k(-\nabla f(\vec{x}_k)). \tag{2.33}$$

Gauss–Newton algorithm is a variation of Newton's optimization method for finding a minimum of a sum of squared function values [28]. Comparing to standard Newton's method Gauss-Newton algorithm can be used only for least-squares nonlinear functions. This disadvantage is balanced by the fact that the second derivation (the Hessian) of the objective function, which is needed in Newton's method and can be challenging to compute, is not required. Another advantage of this algorithm is its fast convergence, on the other hand, if the initial value is not close to minimum of function or the inappropriate step size is chosen, the algorithm can diverge. Update step for this method can be obtained from linear least squares equation

$$\vec{g}'(\vec{x}_k)^T \vec{g}'(\vec{x}_k)(\vec{x} - \vec{x}_k) = -\vec{g}'(\vec{x}_k)^T \vec{g}(\vec{x}_k) \tag{2.34}$$

using pseudoinverse, so the iteration step is given as

$$\vec{x}_{k+1} = \vec{x}_k - \alpha_k \vec{g}'(\vec{x}_k)^+ \vec{g}(\vec{x}_k). \tag{2.35}$$

The Levenberg-Marquardt algorithm eliminates the drawbacks of both described algorithms by merging them together [28]. The iteration step of this method extends the iteration step of Gauss-Newton algorithm so it is given as

$$\vec{x}_{k+1} = \vec{x}_k - (\vec{g}'(\vec{x}_k)^T \vec{g}'(\vec{x}_k) - \mu_k \mathbf{I})^{-1} \vec{g}'(\vec{x}_k)^T \vec{g}(\vec{x}_k), \tag{2.36}$$

where μ_k is a damping parameter and $\mu_k > 0$. The added part $-\mu_k \mathbf{I}$ makes the Levenberg-Marquardt algorithm akin to one of the two other methods. For μ_k very high the term $(\vec{g}'(\vec{x}_k)^T \vec{g}'(\vec{x}_k) - \mu_k \mathbf{I})^{-1}$ approximates $\mu_k^{-1} \mathbf{I}$ and the algorithm behaves as gradient descent with step size μ_k^{-1}. On the other hand for μ_k very low the term $(\vec{g}'(\vec{x}_k)^T \vec{g}'(\vec{x}_k) - \mu_k \mathbf{I})^{-1}$ approaches $(\vec{g}'(\vec{x}_k)^T \vec{g}'(\vec{x}_k))$ and the algorithm behaves as Gauss-Newton algorithm. During each iteration is possible to choice between gradient descent's stability far from optimum and Gauss-Newton algorithm's fast convergence close to optimum by changing a value of μ_k using multiplication and division by a constant c.

The Levenberg-Marquardt algorithm starts with an initial point $\vec{x_0}$ and an initial damping parameter μ_0. Then algorithm repeats these iteration steps:

- Update the value $\vec{x}_{k+1} = \vec{x}_k - (\vec{g}\,'(\vec{x}_k)^T \vec{g}\,'(\vec{x}_k) - \lambda_k I)^{-1} \vec{g}\,'(\vec{x}_k)^T \vec{g}(\vec{x}_k)$

- Next check the accept condition $f(\vec{x}_{k+1}) < f(\vec{x}_k)$. If satisfied, set $\lambda_{k+1} = \frac{\lambda_k}{c}$ else revert $\vec{x}_{k+1} = \vec{x}_k$ and set $\lambda_{k+1} = c\lambda_k$

- Then check the termination criterion, e.g. number of iterations, size of the step. If satisfied, end computation.

2.6 Observability and identifiability

According to [29] the observability index measures the quality of the dataset based on the identification Jacobian matrix **J**, which represents the sensitivity of minimized values to the change of individual parameters. Menq and Borm [30] proposed a measure marked as O_1, Driels and Pathre [31] proposed a measure marked as O_2 and Nahvi and Hollerbach proposed measures O_3 [32] and O_4 [33]. All these measures can be computed from the singular value decomposition of **J**. They are defined as:

$$O1 = \frac{(\sigma_1 \sigma_2 ... \sigma_m)^{1/m}}{\sqrt{n}}, \quad O_2 = \frac{\sigma_{min}}{\sigma_{max}}, \quad O_3 = \sigma_{min}, \quad O_4 = \frac{\sigma_{min}^2}{\sigma_{max}},$$

where σ_i is i-th singular number, m is the number of independent parameters to be identified and n is the number of calibration configurations.

The identification Jacobian matrix itself shows us the identifiability of individual optimized parameters – $\mathbf{J}(i,j) = \frac{\partial X_i}{\partial \phi_j}$, where X_i is a distance or a projection error and ϕ_j is the parameter to be estimated. If a matrix column related to a parameter consists only of zeros, the parameter is unidentifiable leading to the unobservable problem (the minimal singular number is zero). According to [2] an unidentifiable parameter means that the experimental setup does not allow them to be identified, not that they are intrinsically identifiable. The identifiability can be improved by adding additional sensors to the setup.

2.7 Evaluation

As it was in [17] we randomly split the poses from the dataset on traning and testing parts for each repetition separately. The training one contains 70 percent of poses rounded below and the testing one the rest of them, all dataset points of the same pose are in the same part of the dataset. The parameters were optimized on the training dataset. The testing one was used to compare parameters quality of each execution. We chose root-mean-square (RMS) error for that task. We computed RMS error for each optimization problem, equation for problem from Section 2.4.3 are rewritten from [17]. The error of the marker position over dataset D is given as

$$RMS_{so} = \sqrt{\sum_{j=1}^{2} \frac{1}{M} \sum_{i=1}^{M} (p_j(\phi, D_i, \zeta) - z_j(D_i))^2} = \sqrt{\frac{1}{M} \|\vec{P}(\phi, D, \zeta)\|^2}, \tag{2.37}$$

where M is a number of points in dataset D, ϕ is parameter vector and ζ defines all other necessary parameters, $p_j(\phi, D_i, \zeta) - z_j(D_i)$ is difference between j-th coordinate of the projection of the marker given by the dataset point D_i, where $i \in A$, $A = \{1, ..., M\}$ into the camera plane and the actual position in the camera from the dataset point. The RMS error of the self-touch distances over dataset D is given as

$$RMS_{st} = \sqrt{\sum_{i=1}^{K} \frac{1}{K}(c(\phi, D_i, \zeta) - q(\zeta))^2} = \sqrt{\frac{1}{K}\|\vec{C}(\phi, D, \zeta)\|^2}, \tag{2.38}$$

where K is a number of poses, the function $c(\phi, D_i, \zeta)$ computes the distance of the icosahedron centers in the configuration given by the dataset point D_i, where $i \in B$, $B \subset A$ is a subset of the dataset points where the pose of the robot is unique. The $q(\zeta)$ is the distance of the icosahedrons centers. The RMS error of distances between contacts and a single or multiple fitted planes is given as:

$$RMS_{planes} = \sqrt{\sum_{j=1}^{3} \sum_{i=1}^{L_j} \frac{1}{L_j}\|C_j(\phi, D_{j,i}, \zeta)\|^2}, \tag{2.39}$$

where L_j is a number of poses for j-th dataset and C_j is a vector of distances from j-th plane. To compare different executions we have to sum everything up as:

$$RMS_{total} = \sqrt{RMS_{so}^2 + RMS_{st}^2 + RMS_{planes}^2}. \tag{2.40}$$

Chapter 3

Data acquisition and dataset description

3.1 Dataset collection programs

The collision model had to be created for each experiment separately. In Figure 3.1 different objects of a table shape and green color can be seen. They are used as visual representations of the individual planar constraint. The collision between these green objects and the manipulator is prohibited. It means that the path of the robot is planned with respect to these restrictions so that the trajectories and joint distribution can vary a lot between experiments. The only exception is the collision between the object and robot end effector in the specific phase of each touching process defined below.
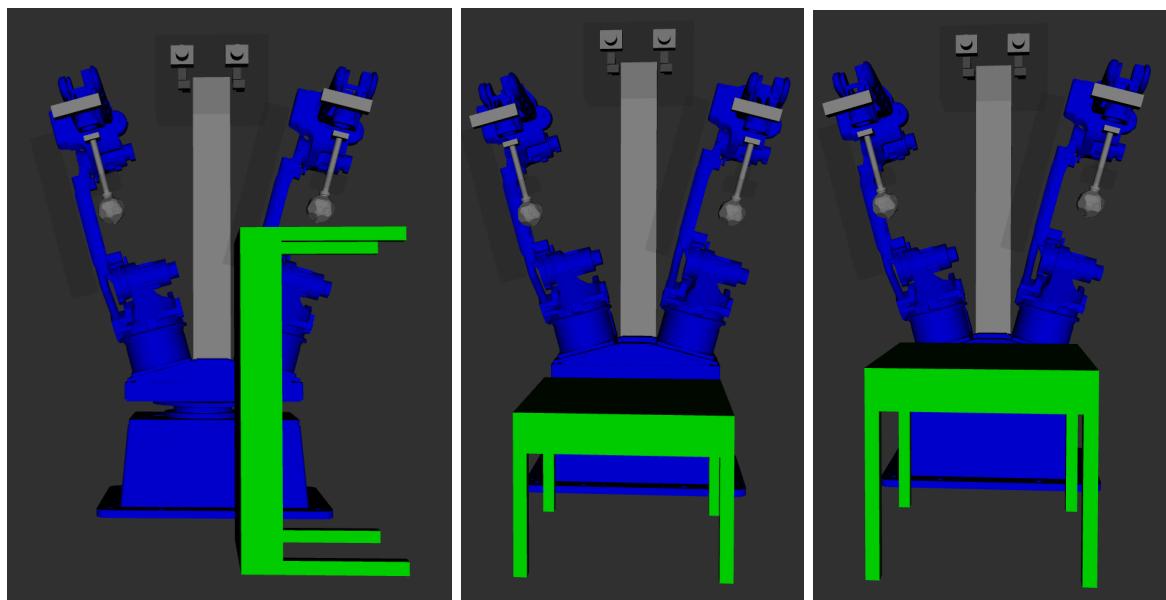
Figure 3.1: Collision models for different planar constraints.

Dataset collection programs are based on a simple GUI applet to collect the dataset from [17]. The experiment consists of touch on every node in a specified grid. Each one is provided in one of the five possible orientations of the end effector.

Each touching process consists of three or four phases depending on the experiment. In the first phase, the robot right arm moves at a speed of 0.7 meters per second to a point close to the required position in the chosen orientation. The part, when the left one moves at a speed of 0.7 meters per second to the required position, follows in case of the self-touch program. Then the right one starts going to the desired position at a speed of 0.1 meters

per second until the collision with the plane or left arm is detected. The deceleration here is necessary for the end effector not to oscillate and to avoid a measure distortion. The collision is enabled only in this phase. Once the end effector touches by any of his spherical tiles in place of the pentagonal faces, arm stops moving and waits for cameras taking a photo. The last phase means the slow return to the position from the end of the first phase. Afterward, the robot chooses the next coordinates and orientation and starts the first phase again.

▇ 3.1.1 Recording data

Once the contact is detected, all data for this pose are captured. Namely, images from both cameras, image from XTion camera on the last robot link, joint configurations of both arms, data from Leica measurement (where available), force information during collisions and positions of detected ArUco markers. Firstly cameras take photos, which are then sent by a ROS topic to the OpenCV function *detectMarkers* to find markers with IDs from 101 to 120 and from 201 to 220 in the taken images. The function returns the coordinates of all four corners of the found marker. From these coordinates, the center of each marker is computed and logged in the dataset. Photos are saved, and their names are added to the dataset. The joint configuration of robot is acquired from ROS as well as the positions of the icosahedrons by forward kinematics, and both are appended to the dataset.

For recording data measured by Leica tracker, we used an application named AT402 Tracker pilot on a notebook prepared to communicate with the tracker. The program offers a measuring mode called stable probing. This mode allows measurement of a position automatically when the robot arm stays in the same location for a while. It is necessary only to set how long a reflector has to be in the same place to start measuring.

Unfortunately, we were able to use this mode only in the Leica tracker experiment (see Section 3.4.1), other experiments contain more than one pause in each touching process, so the program would not recognize when to measure and when not. Due to that, we had to record data manually because the application was not synchronized with our dataset collection programs. So we waited until the cameras take the photo and at that moment we clicked to collect data from Leica tracker. The collecting lasted around a second during which the position of Leica ball reflector was repeatably measured. After that, the mean coordinates and its uncertainty were stored with additional information as the temperature, humidity, pressure, and time stamp. At the end of the experiment, the data were exported to the csv file for further use.

▇ 3.2 Dataset structure

We updated two dataset structures used in [17]. Original dataset, the one produced by the dataset collection program, is expanded by a few additional values collected in the updated version of the program. Then the dataset was merged with the data gained from the Leica tracker measurement (data description in Section 2.1.5). Transformed dataset, the one prepared for the calibration script, was extended by Leica tracker data as well. Matlab script called *convert_leica_dataset.m*, that converts datasets from original to transformed,

was updated to work with new versions of structures and can be found in [19].

3.2.1 Original dataset structure

The dataset consists of the actual state of the state machine (usually this value is zero), assumed pose of both icosahedron centers, the joint configuration of the robot, the magnitude of force measured by both force sensors during and before the contact, the names of saved camera images, the coordinates of the projections of every marker into each of the cameras, the mean position of Leica ball reflector in Leica tracker coordinate system and its expanded uncertainty (described in Section 2.1.5) and Leica tracker measurement timestamp. The dataset is available for download in our Google drive [18].

The projections are sorted in ascending order by marker ID (described in Section 2.1.3) for the right camera image and then for the left camera image. If a marker was not found in the image, its coordinates are denoted as (NaN, NaN). If data were not measured or captured by Leica tracker, all linked values are indicated as NaN. The structure of each line in the dataset is as follows (new values are bold).

index	value description	unit
1	**Actual state of the state machine**	-
2 - 4	Position of the right icosahedron: X, Y, Z	mm
5 - 8	Orientation of the right icosahedron: W, X, Y, Z (quaternion)	-
9 - 11	Position of the left icosahedron: X, Y, Z	mm
12 - 15	Orientation of the left icosahedron: W, X, Y, Z (quaternion)	-
16 - 28	Joint configuration of the robot: turntable, S1, L1, U1, R1, B1, T1, S2, L2, U2, R2, B2, T2	rad
29	Peak force (magnitude) during collision on right arm	N
30	Peak force (magnitude) during collision on left arm	N
31	**Peak force (magnitude) before collision on right arm**	N
32	**Peak force (magnitude) before collision on left arm**	N
33	Name of the image file from right camera	-
34	Name of the image file from left camera	-
35 - 114	40 × positions of the marker centers (IDs 101 to 120, 201 to 220) in the right camera: u, v	px
115 - 194	40 × positions of the marker centers (IDs 101 to 120, 201 to 220) in the left camera: u, v	px
195 - 197	**Position of Leica ball reflector in Leica tracker coordinate system: X, Y, Z**	mm
198	**Uncertainty U95 of Leica ball reflector position in Leica tracker coordinate system**	mm
199 - 201	**Uncertainty U95 of Leica ball reflector position in Leica tracker coordinate system for individual axis: X, Y, Z**	mm
202	**Leica measurement timestamp in format DD.MM.YYYY hh:mm:ss.sss**	-

■ 3.2.2 Transformed dataset structure

As it was written in [17] the transformed dataset contains only the markers that were actually detected. Due to optimization reasons one line relates to one data point. It consists of a number uniquely defining the robot pose in the experiment, a face number of the detected marker, the number of the arm (1 for right or 2 for left), the number of the camera (1 for right or 2 for left), the coordinates of the projected point, the current robot joint configuration, the mean position of Leica ball reflector in Leica tracker coordinate system and its expanded uncertainty. If data were not captured by Leica tracker, all linked values are indicated as NaN. The structure of one line in the dataset is as follows (new values are bold).

index	value description	unit
1	Pose ID: monotonic integer	-
2	Face ID: 1 to 20	-
3	Arm number: 1 or 2	-
4	Camera number: 1 or 2	-
5 - 6	Position of the marker center in the camera: u, v	px
7 - 19	Joint configuration of the robot: turntable, S1, L1, U1, R1, B1, T1, S2, L2, U2, R2, B2, T2	rad
20 - 22	**Position of Leica ball reflector in Leica tracker coordinate system: X, Y, Z**	m
23	**Uncertainty U95 of Leica ball reflector position in Leica tracker coordinate system**	m
24 - 26	**Uncertainty U95 of Leica ball reflector position in Leica tracker coordinate system for individual axis: X, Y, Z**	m

■ 3.3 Problems during collecting datasets

A few problems occurred during datasets collection, which delayed the process of datasets completion. First of them was a persisting problem with cameras. When the collecting application was not closed correctly, the cameras stopped to respond to messages generated by libgphoto2 [17] and could not be seen by the computer as well. We had to restart them either by a switch on the device or by a power outage which was our preferred option because it was more accessible than the switch and there was no danger in shifting the camera while pushing the switch.

The problem causing that the application was not closed properly was that the motion planning framework crashed unexpectedly during the experiment. The least inappropriate moment when that happened was during the contact with a plane or the other arm. Then the robot had to be manually moved out from the collision space. During the manual manipulation with the robot, the robotic arm twice collided with the objects in the workspace.

The other problems did not influence the experiment itself but made the datasets incomplete. We were able to collect data by Leica tracker only in one orientation for each experiment because the reflector had to be oriented to the tracker all the time. However, the

reflector oriented the tracker did not guarantee the position would be logged by Leica tracker. A few positions in each dataset were not captured because the position measurement was unstable - it means the arm did not stay in the pose for enough time. On the other hand, all of these invalid poses were logged at least with the information from cameras and the joint configuration.

The last problem was that joint limits and collision restrictions caused that the datasets did not contain all points and all orientations. There were a few points in a grid where the robot arm was not able to move in the desired orientation, or the trajectory was not collision-less, so the orientation or the whole point were skipped and could not be included in a dataset.

3.4 Individual experiments

The Leica ball-reflector was located on the right arm in all experiments except of left-hand Leica tracker experiment, regardless the Leica tracker measured or not. Leica tracker saved the information about the environment for the experiments with Leica capturing positions. All touching experiments ran twice to compare repeatability. Dimensions of grids are described in a robot base coordinate system.

3.4.1 Leica tracker
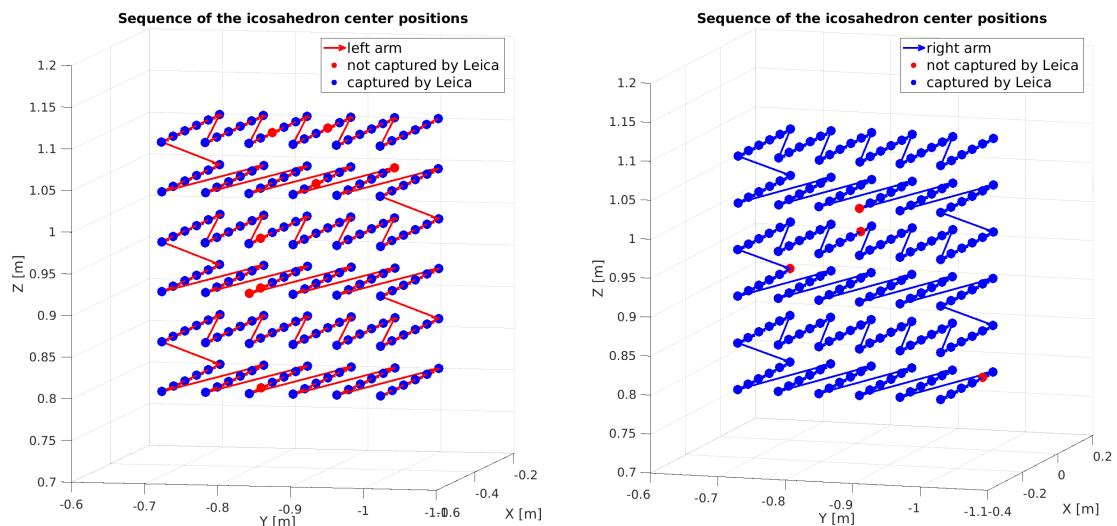
Figure 3.2: Positions of icosahedrons centers in the Leica datasets with the information whether the pose was logged by Leica or not.

Each arm had its own experiment, so the second one was hidden outside of cameras view not to affect the experiment. The Leica ball-reflector was placed on the arm moving in the experiment. These experiments differ from others. There are no collisions to log, so the robot's move does not have to be decelerated and can move directly to the desired position in chosen orientation at a speed of 0.7 meters per second. Once the location is reached, the

arm stops moving and waits for cameras taking a photo. Afterward, the robot chooses the next coordinates with the same orientation and starts moving there. The orientations of the end effectors on both arms were chosen to keep the reflector visible to the Leica tracker. On the right arm, the reflector was rotated to values (85, -90, 0) in configuration ZYZ of Euler's angles and on the left one the reflector was rotated to (85, 90, 0) in the same configuration. One orientation was enough because Leica tracker does not measure the orientation 2.1.5. The selected arm was moving in a $6 \times 6 \times 6$ grid. The grid's dimensions were from -0.4 m to 0.1 m along the x-axis, from -1.0 m to -0.7 m along the y-axis, from 0.8 m to 1.1 m along the z-axis. The grid was identical for both arms.

Leica tracker did not capture eight positions for the left-hand experiment, so the dataset contains 208 poses, and the temperature was from 22.7 to 22.8 °C, pressure 988 mBar and humidity 27 %. For the right-hand experiment, Leica did not log four poses, so 212 poses were saved to the dataset and the temperature was from 23.8 to 24.0 °C, pressure 989 mBar and humidity 26 %.

3.4.2   Horizontal plane contact
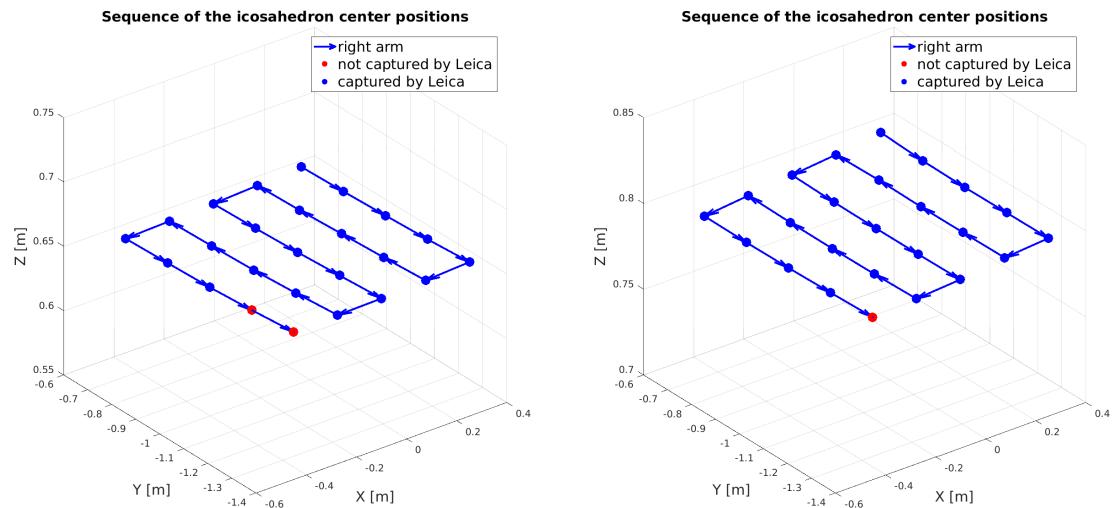
Figure 3.3: Positions of icosahedrons centers in the horizontal plane contact datasets (the lower table is the left one) with the information whether the pose was logged by Leica tracker or not.

The horizontal plane contact experiments consist of touch on every node in a 5×5 grid on the xy-plane. Only the right robot arm is used for contact with a plane. The left one is hidden outside of cameras view not to affect the experiment. The table center is located a meter in front of the robot base in the x-axis and lies on the y-axis. Two tables were used first one's height was set to the value approximately 0.55 m, and the grid's dimensions were from -0.4 m to 0.3 m along the x-axis, from -1.35 m to -0.65 m along the y-axis. The second one was higher, and its height was set to the value approximately 0.70 m, and the grid was the same as before. The heights are calibrated as well, so the exact values were not necessary.

In the beginning, the experiment for the lower table was prepared. The first orientation of end effector was chosen to keep the reflector visible to the Leica tracker. The spherical tile *r1_bp_02* was rotated to values (0, 185, 52) in configuration XYZ of Euler's angles, so the tile touches the plane. The last two positions of this experiment were not captured by Leica tracker, so 23 poses were logged in the dataset. The temperature was 24.2 °C, pressure 983 mBar and humidity 30 %.

Then we continued with the other spherical tiles, so their orientations were set. These orientations are in Table 3.1 marked as "No" in the first column. Three positions in all four and one in three orientations were not logged, so the dataset contains $21 \cdot 4 + 1 = 85$ records.

Once the experiments with the lower table were finished, the tables were changed, and we resumed with the first orientation on the higher table. Leica tracker did not capture the last position, so 24 poses were logged in the dataset. The temperature was from 23.9 to 24.1 °C, pressure 980 mBar and humidity from 30 to 31 %. The orientations of the other spherical tiles stayed the same as they were in the experiment with the lower table (see Table 3.1). Two positions in all orientations were not reachable by the arm, so the dataset contains $23 \cdot 4 = 92$ records. The horizontal plane contact dataset contains 108 records for the lower table, 116 for the higher table, so 224 records in total.

Leica tracker	spherical tile	XYZ angles
Yes	*r1_bp_02*	(0, 185, 52)
No	*r1_bp_01*	(0, 180, -20)
No	*r1_bp_03*	(0, 180, 124)
No	*r1_bp_04*	(0, 180, -164)
No	*r1_bp_05*	(0, 180,-92)

Table 3.1: XYZ Euler's angles for spherical tiles in the horizontal plane contact experiments.

3.4.3 Vertical plane contact

The vertical plane contact experiment consists of touch on every node in a 5×5 grid on the yz-plane. As it was in horizontal plane contact experiment, only a right robot arm is used for contact with a plane. The left one is hidden outside of cameras view not to affect the experiment. The normal vector of the plane is parallel with the y-axis of the robot base, and a plane vector parallel with the x-axis of the robot base lies sixteen centimeters in the x-axis direction. The grid's dimensions were from -1.2 m to -0.7 m along the y-axis, from 0.7 m to 1.1 m along the z-axis.

The first orientation of end effector was chosen to keep the reflector visible to the Leica tracker. The spherical tile *r1_bp_05* was rotated to values (-108, 90, 0) in configuration ZYZ of Euler's angles, so the tile touches the plane. All positions were captured by Leica tracker, so all 25 poses were logged in the dataset. The temperature was from 23.4 to 23.7 °C, pressure 978 mBar and humidity from 30 to 32 %.

Then we continued with the other spherical tiles, so their orientations were set. These orientations are in Table 3.2 marked as "No" in the first column. The arm was not able to
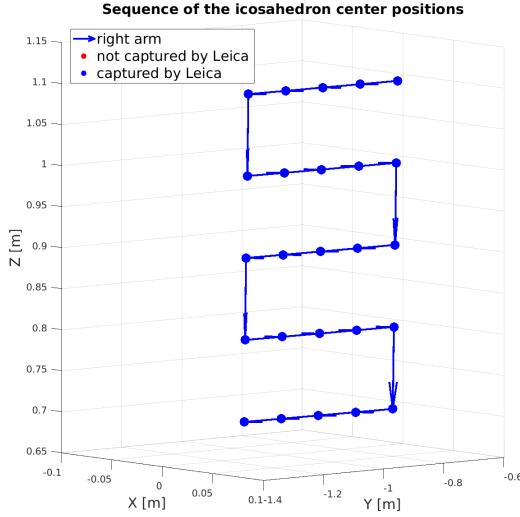
Figure 3.4: Positions of icosahedrons centers in the touching vertical plane dataset with the information whether the pose was logged by Leica tracker or not.

reach only one orientation of one position, so the dataset contains $24 \cdot 4 + 3 = 99$ records. The vertical plane contact dataset contains 124 records in total.

Leica tracker	spherical tile	ZYZ angles
Yes	*r1_bp_05*	(-108, 90, 0)
No	*r1_bp_01*	(180, 90, 0)
No	*r1_bp_02*	(108, 90, 0)
No	*r1_bp_03*	(36, 90, 0)
No	*r1_bp_04*	(-36, 90, 0)

Table 3.2: ZYZ Euler's angles for spherical tiles in the vertical plane contact experiments.

3.4.4 Self-touch

The self-touch experiment consists of touch on every node in a 4×4 grid on the xy-plane in two heights. The lower one has 0.8 m along the z-axis, and the grid's dimensions were from -0.3 to 0.2 m along the x-axis and from -1.1 to -0.6 m along the y-axis. The second one has 1 m along the z-axis, and the grid's dimensions were from -0.3 to 0.1 m for orientation and from -0.3 to 0.2 m else along the x-axis and from -1.1 to -0.6 along the y-axis.

The first orientation of the right arm end effector was chosen to keep the reflector visible to the Leica tracker. The spherical tile *r1_bp_05* was rotated to values (-108, 90, 0) in configuration ZYZ of Euler's angles, so the tile is oriented to touch the left-arm end effector. Then one spherical tile on the left end effector was chosen and rotated to be hit by the *r1_bp_05*. The selected tile was *r2_bp_01*, and the rotation was (0, -90, 0) in the same configuration. The temperature was from 24.2 to 24.4 °C, pressure 985 mBar and humidity from 29 to 30 %. All positions were captured by Leica tracker, so all 16 poses were logged in the dataset for both heights.

28
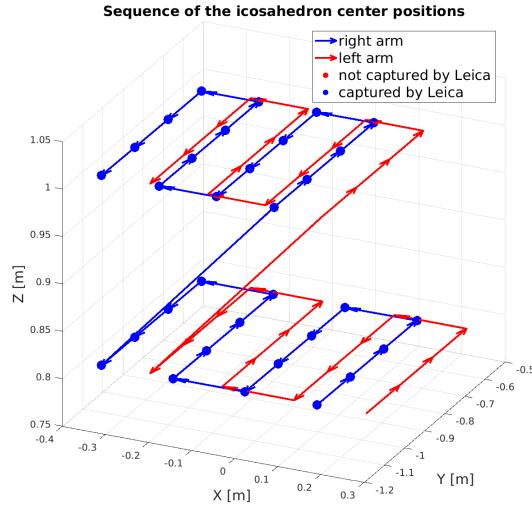
Figure 3.5: Positions of icosahedrons centers in the self-touch dataset with the information whether the pose was logged by Leica tracker or not.

Then we continued with the other orientations, so eight from 24 combinations of pairs of the spherical tiles were chosen. These combinations are in Table 3.3 marked as "No" in the first column. For the lower height, two orientations of one position were not logged in both repetitions, and two orientations in another position were not registered only in the second run, so the dataset contains $15 \cdot 8 + 6 = 126$ records for the first run and $14 \cdot 8 + 2 \cdot 6 = 124$ for the second one. For the upper one, four orientations of one position were not captured only in the first run, so the dataset contains $15 \cdot 8 + 4 = 124$ records for the first run and $16 \cdot 8 = 128$ for the second one. The self-touch dataset contains 534 records in total.

Leica tracker	right arm spherical tile	right arm ZYZ angles	left arm spherical tile	left arm ZYZ angles
Yes	*r1_bp_05*	(-108, 90, 0)	*r2_bp_01*	(0, -90, 0)
No	*r1_bp_01*	(180, 90, 0)	*r2_bp_05*	(72, -90, 0)
No	*r1_bp_02*	(108, 90, 0)	*r2_bp_02*	(-72, -90, 0)
No	*r1_bp_03*	(36, 90, 0)	*r2_bp_05*	(72, -90, 0)
No	*r1_bp_04*	(-36, 90, 0)	*r2_bp_02*	(-72, -90, 0)
No	*r1_bp_01*	(180, 90, 0)	*r2_bp_03*	(-144, -90, 0)
No	*r1_bp_02*	(108, 90, 0)	*r2_bp_04*	(144, -90, 0)
No	*r1_bp_03*	(36, 90, 0)	*r2_bp_04*	(144, -90, 0)
No	*r1_bp_04*	(-36, 90, 0)	*r2_bp_03*	(-144, -90, 0)

Table 3.3: Combinations of spherical tiles with ZYZ Euler's angles in the self-touching experiments.

3.4.5 Repeatability measurement

Repeatability measurement experiments were performed for each of the setup (self-touch, horizontal plane, and vertical plane) to evaluate the repeatability of the measurement.

Each experiment consists of 19 or 20 repetitions of touches in 2 different positions. The orientations were taken from previous experiments to keep the reflector visible to the Leica tracker. Number of repetitions, points of contact (in the robot base coordinate system) and additional information about the environment (temperature, humidity, and pressure) from Leica tracker for each setup can be found in Table 3.4.

setup	# of repetitions	first point	second point	temperature [$°C$]	humidity [%]	pressure [mBar]
Self touch	20	x = 0.225 y = -0.85 z = 0.8	x = -0.025 y = -0.85 z = 1	23.6 - 23.8	27 - 28	988
horizontal plane	19	x = 0.3 y = -0.825 z = 0.67	x = -0.4 y = -0.825 z = 0.67	23.1 - 23.3	28	988
vertical plane	19	x = 0.17 y = -0.825 z = 1.1	x = 0.17 y = -0.95 z = 0.7	24.4 - 24.7	29 - 30	987

Table 3.4: Information about repeatability experiment for each setup. First and second point are the coordinates of contact points in the robot base coordinate system.

■ 3.5 Joint angle distribution

Figure 3.6 shows the distributions of joint angles on the right arm. The upper one compares the vertical plane contact experiment with the horizontal plane one. We can see that the angles of the joint R1 are different for both setups; on the other hand, the other joints have similar values. The lower figure compares the self-touch setup with the combination of all planar setups. It can be concluded, that the planar setups have richer joint angle distribution, and only a few angles are purely for self-touch.
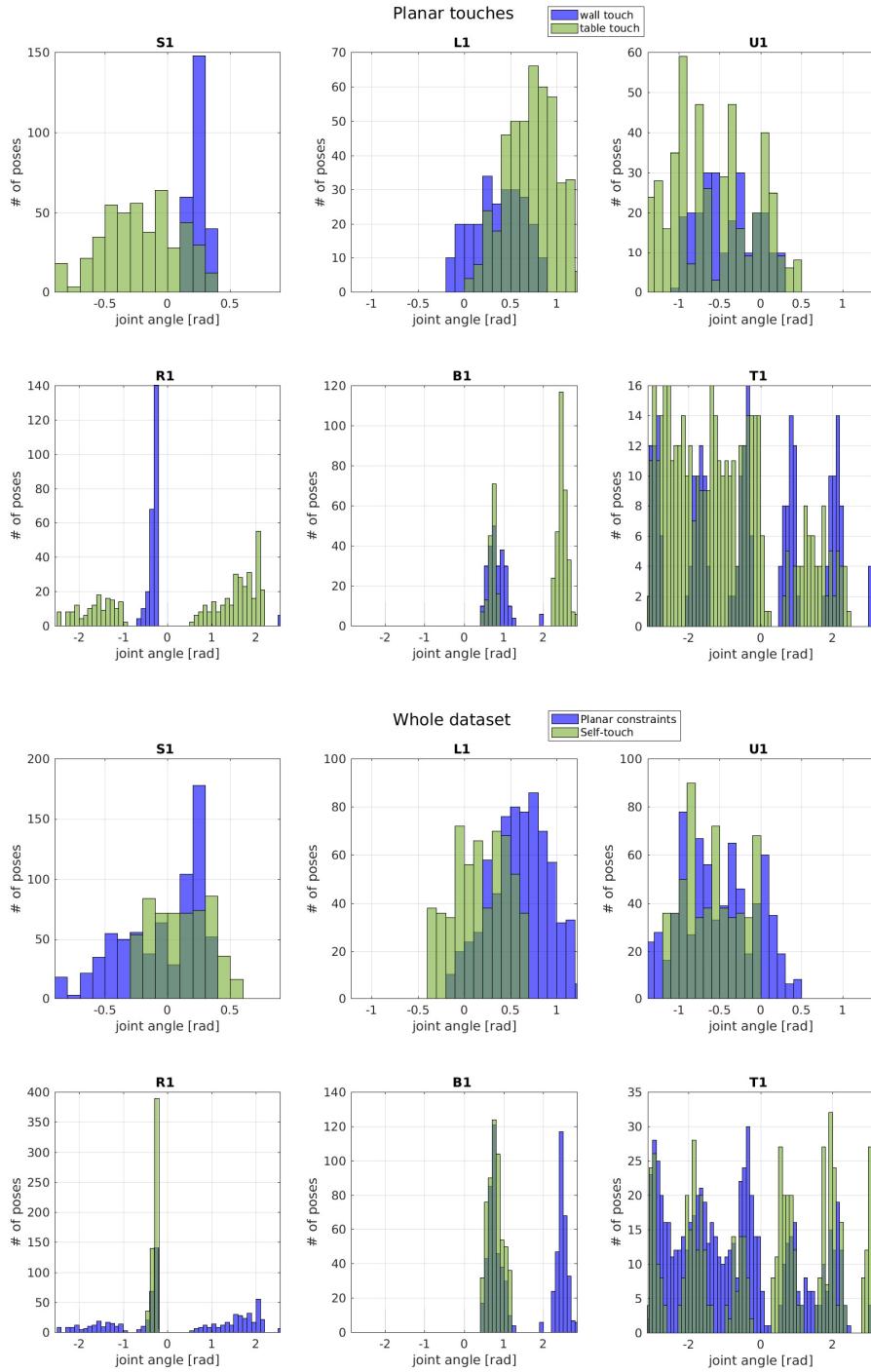
Figure 3.6: Distributions of joint angles across the different setups.

31

Chapter 4

Results

The parameter corrections show how the found DH parameters differ from the nominal ones. The corrections are visualized in box plots. The box plots show the median (the central red line), the bottom and top edges of the box indicating the 25th and 75th percentiles, respectively. The outliers are plotted individually using the "+" symbol. The error graphs show mean values and the vertical error bars symbolize max and min values (not the standard deviation) over usually ten repetitions. Most of the calibrations were made on combinations of datasets which can be found below.

datasets	label
vertical plane	VP
lower horizontal plane	LH
higher horizontal plane	HH
lower and higher horizontal planes	BH
vertical plane and lower horizontal plane	LHVP
vertical plane and higher horizontal plane	HHVP
vertical plane, lower horizontal plane and higher horizontal plane	AP
self-touch	ST
vertical plane, lower horizontal plane, higher horizontal plane and self-touch	ALL

Table 4.1: Datasets combinations for calibration sets and their labels in graphs.

4.1 Leica tracker calibration

We wanted to know how the robot dimensions differ from the nominal ones after years of use. Therefore we took the nominal DH parameters as starting parameters, and we calibrated them using the Leica tracker. We were not able to calibrate DH parameters of our custom end effector because the retro-reflector was attached to the end-effector and became a new end-effector itself. The DH parameters of the retro-reflector were optimized too. However, as it can be seen in Figure 4.1, DH parameters of the retro-reflector are badly identifiable. On top of that, the parameter α_{LR} was omitted from optimization to keep it zero, since tip link and end effector z-axes are parallel. The observability was computed from the identification Jacobian matrix. The values can be found in Table 4.2.

The calibration was repeated 100 times for each arm; the parameters corrections for both arms can be seen in Figure 4.3. The red dots in the same figure show the corrections of

arm	O_1 $[10^4]$	O_2 $[10^{-14}]$	O_3 $[10^{-4}]$	O_4 $[10^{-18}]$
right	2.8771	1.9471	3.9774	7.7442
left	3.3710	0.74269	2.1558	1.6011

Table 4.2: Values of different observation indices for Leica tracker calibration.

the parameters with the smallest RMS error on a testing dataset. Most of the selected DH parameters lie between 25th and 75th percentile of estimated parameter values (some are median values). We can notice, that the corrections for the right arm have significantly smaller distortion. The green dots mark the differences between nominal values and the values from the calibration mentioned in [1].

Figure 4.2 which shows the RMS errors after calibration, has a small range of values for both arms. Since a few parameters corrections were far from the median, we decided to compare the RMS errors over the whole dataset. We compared the DH parameters, which had the smallest RMS error on the testing dataset (RMS_{min}) with the DH parameters, which had median corrections (RMS_{med}). The errors for the right and the left arm separately are:

$$RMS_{min,right} = 0.076033 \ mm, \quad RMS_{med,right} = 0.076297 \ mm$$
$$RMS_{min,left} = 0.077601 \ mm, \quad RMS_{med,left} = 0.078151 \ mm.$$

The difference between errors is less than 1 % for both arms, so we decided to choose median parameters (can be found in Table 4.3) for further comparison.
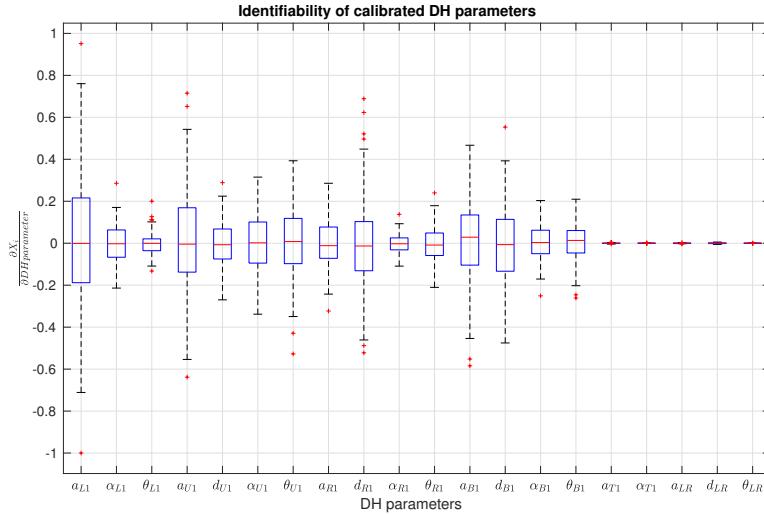
Figure 4.1: Identifiability of all calibrated DH parameters on the right arm (similar for left one) in Leica tracker calibration.
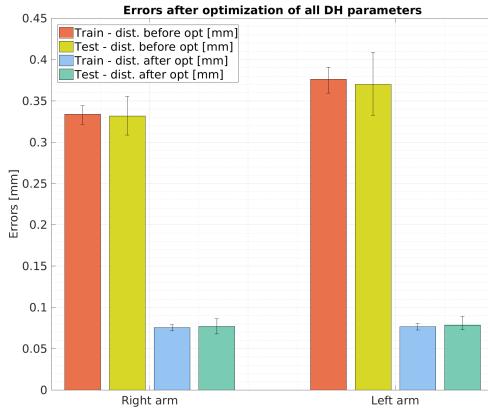
Figure 4.2: Errors before and after calibration of the all DH by Leica tracker. We show absolute errors of distances between sets of points in mm over training and testing dataset for each arm.
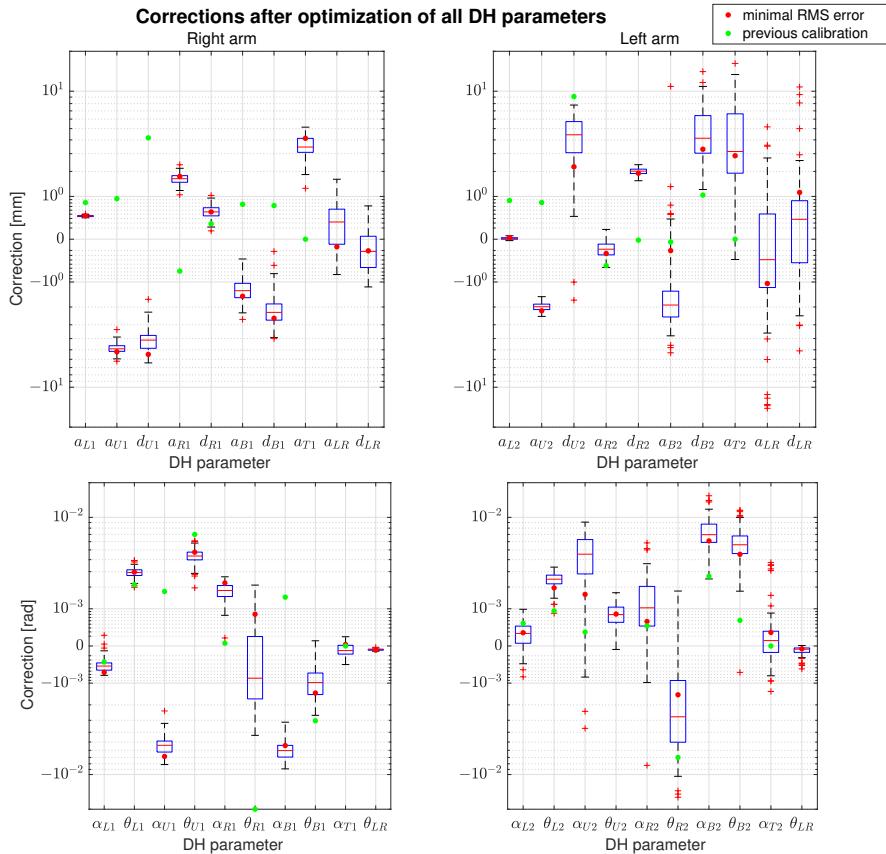
Figure 4.3: Corrections of DH parameters over 100 repetitions of calibration by Leica tracker, right arm (left) and left arm. The red dots correspond the corrections with the smallest RMS error. The green dots show the differences of the calibrated values mentioned in [1] from nominal ones.

35

Manipulator 1 (right arm)

i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.571 (0)
S1	0.15 (0)	1.416 (0)	-1.571 (0)	0 (0)
L1	0.6145 (0.0005)	0 (0)	3.141 (-0.0005)	-1.568 (0.0029)
U1	0.1951 (-0.0049)	-0.0041 (-0.0041)	-1.576 (-0.0054)	0.0044 (0.0044)
R1	0.0017 (0.0017)	-0.6394 (0.0006)	1.573 (0.0018)	-0.0008 (-0.0008)
B1	0.0287 (-0.0013)	-0.0023 (-0.0023)	1.565 (-0.006)	-1.572 (-0.001)
T1	0.0035 (0.0035)	0.2 (0)	-0.0001 (-0.0001)	0 (0)
EE1	0 (0)	0.354 (0)	0 (0)	0 (0)

Manipulator 2 (left arm)

i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
TT1	0 (0)	-0.263 (0)	-0.2618 (0)	-1.571 (0)
S1	0.15 (0)	1.416 (0)	-1.571 (0)	0 (0)
L1	0.614 (0)	0 (0)	3.142 (0.0003)	-1.568 (0.0025)
U1	0.198 (-0.002)	0.0044 (0.0044)	-1.566 (0.0045)	0.0008 (0.0008)
R1	-0.0002 (-0.0002)	-0.638 (0.002)	1.572 (0.001)	-0.0027 (-0.0027)
B1	0.0281 (-0.0019)	0.0041 (0.0041)	1.578 (0.007)	-1.565 (0.0056)
T1	0.0031 (0.0031)	0.2 (0)	0.0001 (0.0001)	0 (0)
EE1	0 (0)	0.354 (0)	0 (0)	0 (0)

Table 4.3: The optimized values of DH parameters of both arms after Leica calibration, the differences from nominal ones are written in brackets, gray parameters were not calibrated.

Previous calibration - Manipulator 1 (right arm)

i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
S1	0.149 (-0.001)	1.416 (0)	-1.571 (0.0001)	0 (0)
L1	0.6148 (0.0008)	0.0037 (0.0037)	3.141 (-0.0003)	-1.569 (0.0021)
U1	0.2009 (0.0009)	0.0042 (0.0042)	-1.569 (0.0018)	0.007 (0.007)
R1	-0.0007 (-0.0007)	-0.6397 (0.0003)	1.571 (0.0001)	-0.02 (-0.02)
B1	0.0308 (0.0008)	0.0007 (0.0007)	1.572 (0.0015)	-1.574 (-0.003)
T1	0 (0)	0.2 (0)	0 (0)	0 (0)

Previous calibration - Manipulator 2 (left arm)

i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
S2	0.149 (-0.001)	1.416 (0)	-1.571 (0.0001)	0 (0)
L2	0.6149 (0.0009)	0.0083 (0.0083)	3.142 (0.0005)	-1.57 (0.0009)
U2	0.2008 (0.0008)	0.0091 (0.0091)	-1.571 (0.0003)	-0.053 (-0.053)
R2	-0.0005 (-0.0005)	-0.64 (0)	1.571 (0.0005)	-0.007 (-0.007)
B2	0.03 (0)	0.001 (0.001)	1.573 (0.0027)	-1.57 (0.0006)
T2	0 (0)	0.2 (0)	0 (0)	0 (0)

Table 4.4: The values of DH parameters from [1], the differences from nominal ones are written in brackets.

■ 4.2 Calibration camera kinematic chains

In the case of a sequential calibration, the camera DH parameters are calibrated first. We fixed all DH parameters except the cameras ones. As a dataset, we used a combination of a self-touch dataset and all datasets with touches to a plane. The calibration on this huge dataset was repeated ten times, and the parameters with the lowest RMS error were used for calibration of other chains. The corrections and errors can be seen in Figure 4.4 and the chosen values of DH parameters in Table 4.5. The corrections are significant and show that the initial DH parameters from [17] were not made perfectly.

Camera 1

i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
RC1	0.0733 (-0.0767)	1.8035 (-0.1565)	-2.5087 (-0.1525)	-2.7764 (0.1907)
RC2	0 (0)	-0.5677 (-0.1227)	0 (0)	0.2853 (0.2853)

Camera 2

i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
LC1	0.2324 (0.0824)	1.8592 (-0.1008)	2.5484 (0.1922)	0.0850 (0.2595)
LC2	0 (0)	-0.4998 (-0.0548)	0 (0)	3.2205 (0.0789)

Table 4.5: The optimized values of DH parameters of both camera chains after calibration, the differences from nominal ones are written in brackets, gray parameters were not calibrated.
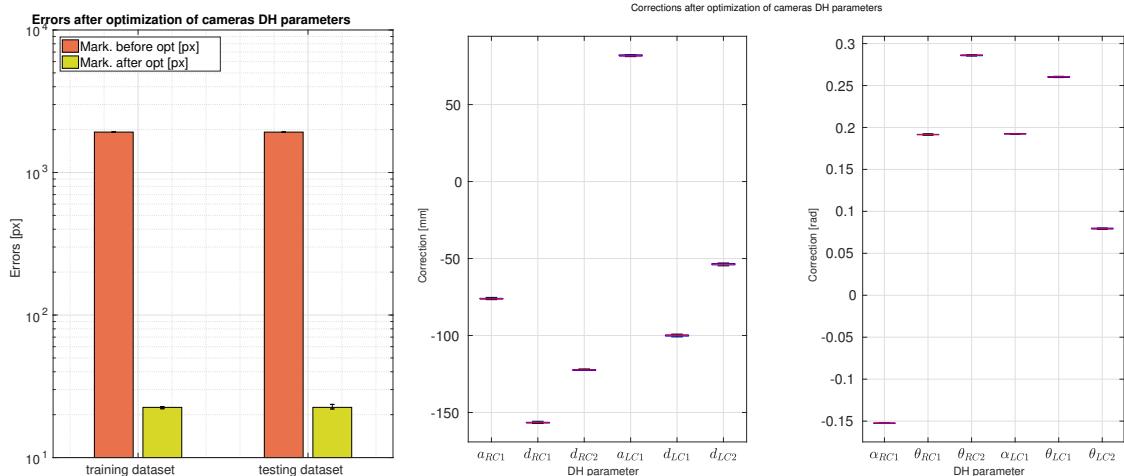
Figure 4.4: Errors (left) and corrections (right) after calibration of the cameras parameters. We show the absolute errors of marker distances in camera frame (in pixels).

4.3 End effector calibration

At first, we carried out sets of calibrations with only an end effectors (tools) to be calibrated. This is a very typical problem in industry, where different tools are attached to the robotic arm, and their length and orientation is to be calibrated. In order to calibrate all DH parameters of the robot, it was also a natural first step in our calibration since we installed the new end-effector with unknown parameters (see Section 2.1.1). As can be seen in the same section, tip link and end effector lie on the same axis, so the DH parameters a and α cannot differ from zero. d and θ, only parameters which can be calibrated, are shown in Figure 4.5. If the calibration is provided using only touching constraints, not cameras, we are not able to identify the θ parameter (see Figure 4.6 because we gained only 3D information – a position without an orientation – from touching constraints. We take the second arm as ground truth for the self-touch constraint.
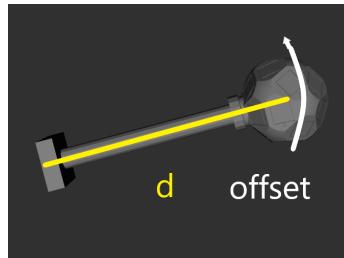
Figure 4.5: End effector with marked d and θ.

We compare the calibration ways based on the errors after optimization and the parameters correction. The reason is that we want to have a corresponding robot model after optimization, it means the DH parameters are suitable, and for example, the robot is not in collision with himself.

4.3.1 Calibration without perturbation

We started from an approximately measured end effector length (parameter d) and rotation (offset), and we observed how it was changing for different calibration setup. The first calibration was done without any information from cameras, so the offset was not calibrated. The following Figure 4.7 shows the correction of the end effector length which means, how long the end effector should be to minimize the root-mean-square error of the chosen calibration set and the touch distance and plane distance errors of individual setups after optimization.

As can be seen, the most significant change in the parameter value was when we used calibration by a vertical plane. However, the other planes and their combinations even with the vertical plane have a similar change in the end effector length. For self-touch and all constraints together, the length did not change at all. From Figure 4.7, we can observe that if we combine all constraints, we get similar touch distance and plane distance errors as when optimized separately by planar constraints or by self-touch, but the length correction differs. In the first case, the correction is almost zero. In the second case, the correction is almost zero for the self-touch as well, but for planes, the mean change is about 1 cm.
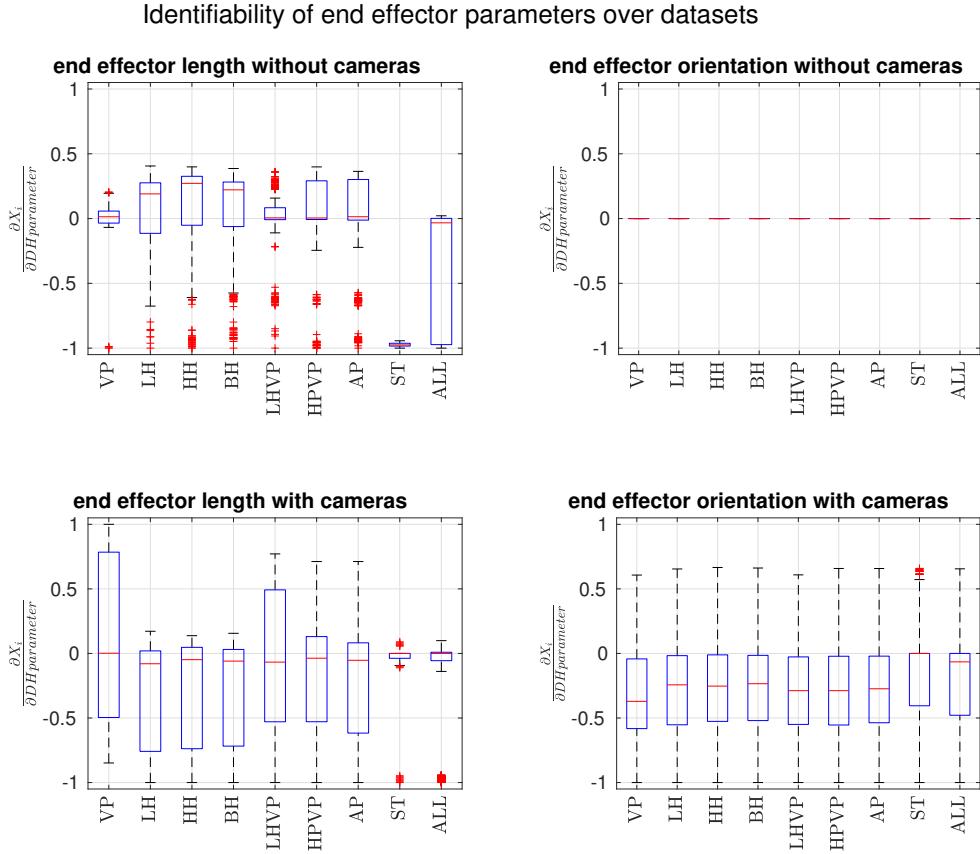
Figure 4.6: Identifiability of end effector d and θ parameters for individual calibration set. First row shows the identifiability for calibration without the self-observation, second row shows it with the self-observation.

Then the information from cameras was added to the calibration. We compared two ways of calibration. In the first case, the calibration was made sequentially; it means that the cameras were calibrated at first (see Section 4.2), and then the optimized camera DH parameters were used for the calibration of end effector length and offset. In the second case, we made the joint calibration, so the end effector parameters were calibrated together with cameras. The errors after optimization are compared in Figure 4.8; we can see that the error of marker distances in camera frame is better for the sequential calibration, on the other hand, the other errors are slightly better in the joint calibration. The comparison of correction in Figure 4.9 shows that the precalibrated cameras got us rid of huge corrections of the end effector length in the case of the calibration by a vertical plane. Unfortunately, the joint calibration compensated this massive correction of almost 25 cm by changing the camera DH parameters. The corrections of cameras DH parameters can be found in Appendix (see Section A.3).

without cameras

	with end effector orientation				without end effector orientation			
setup	O_1	O_2	O_3	O_4	O_1 $[10^9]$	O_2 $[10^0]$	O_3 $[10^{11}]$	O_4 $[10^{11}]$
VP	0	0	0	0	0.0000187	1	0.00000849	0.00000849
LH	0	0	0	0	0.0024	1	0.0009352	0.0009352
HH	0	0	0	0	0.002509	1	0.001098	0.001098
BH	0	0	0	0	0.003544	1	0.002095	0.002095
LHVP	0	0	0	0	0.001515	1	0.0009163	0.0009163
HPVP	0	0	0	0	0.00189	1	0.001191	0.001191
AP	0	0	0	0	0.002822	1	0.002087	0.002087
ST	0	0	0	0	6.724	1	4.839	4.839
ALL	0	0	0	0	4.696	1	4.846	4.846

with cameras

	precalibrated cameras				non-precalibrated cameras			
setup	O_1 $[10^8]$	O_2 $[10^{-3}]$	O_3 $[10^8]$	O_4 $[10^5]$	O_1 $[10^8]$	O_2 $[10^{-4}]$	O_3 $[10^8]$	O_4 $[10^4]$
VP	0.28	2.599	0.6523	1.695	0.1095	0.0533	0.003545	0.000189
LH	0.2227	1.995	0.3881	0.7741	0.1225	2.619	0.1212	0.3173
HH	0.2765	1.872	0.5306	0.9933	0.1737	2.82	0.1819	0.513
BH	0.3524	1.896	0.9012	1.709	0.2333	2.968	0.3264	0.9688
LHVP	0.3567	2.31	1.026	2.37	0.3285	4.61	0.495	2.282
HPVP	0.3917	2.202	1.164	2.564	0.3427	4.815	0.5968	2.874
AP	0.4444	2.163	1.524	3.296	0.4156	4.55	0.7555	3.437
ST	1.393	0.3317	1.824	0.605	0.5123	1.271	0.7205	0.9159
ALL	1.398	0.5422	3.357	1.82	0.7726	2.309	1.559	3.599

Table 4.6: Values of different observation indices for individual calibration setups for end effector calibration.
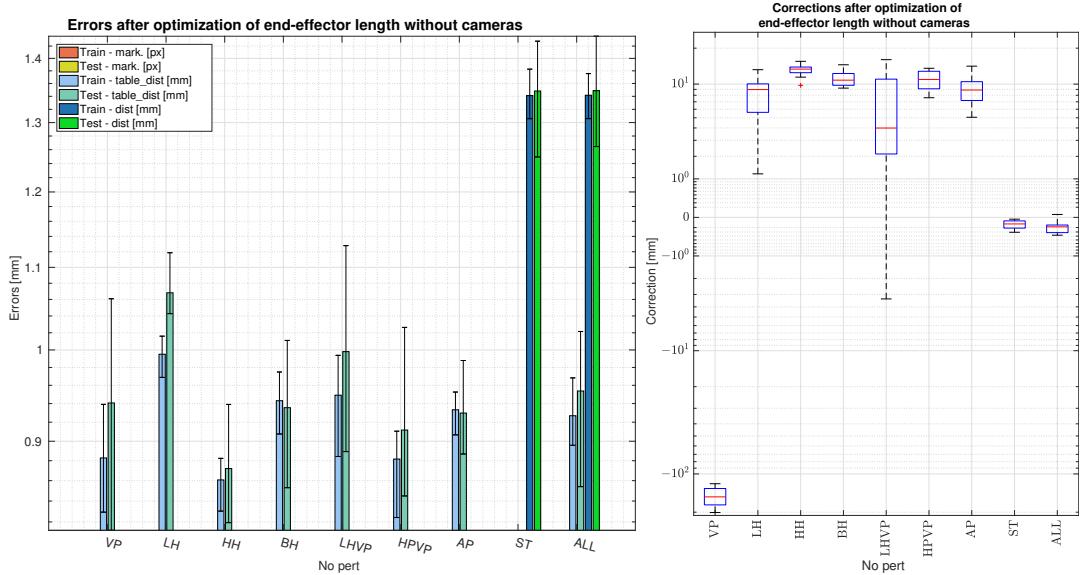
Figure 4.7: Errors (left) and corrections (right) after calibration of the end effector length of individual calibration set without perturbation and without cameras. We show absolute errors of icosahedrons distances (in mm) and distances between touches and the computed planes over a train and a test dataset.
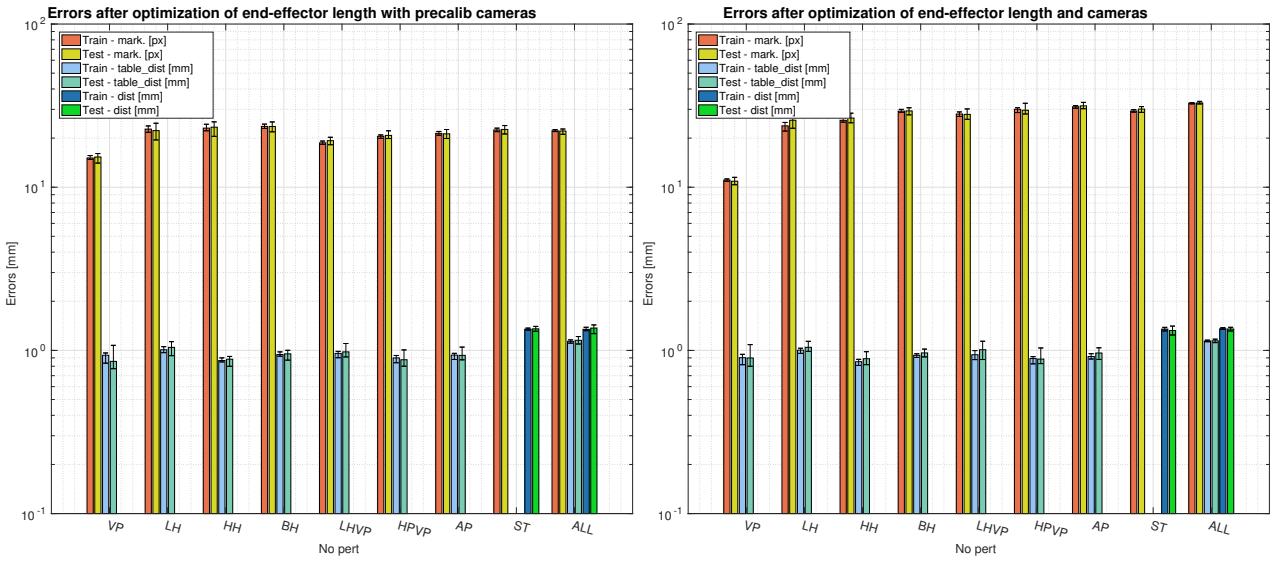
Figure 4.8: Errors after calibration of the end effector length and rotation of individual calibration set without perturbation with precalibrated cameras (left) or without (right). We show absolute errors of markers distances in camera frame (in pixels), icosahedrons distances (in mm) and distances between touches and the computed planes over a train and a test dataset.
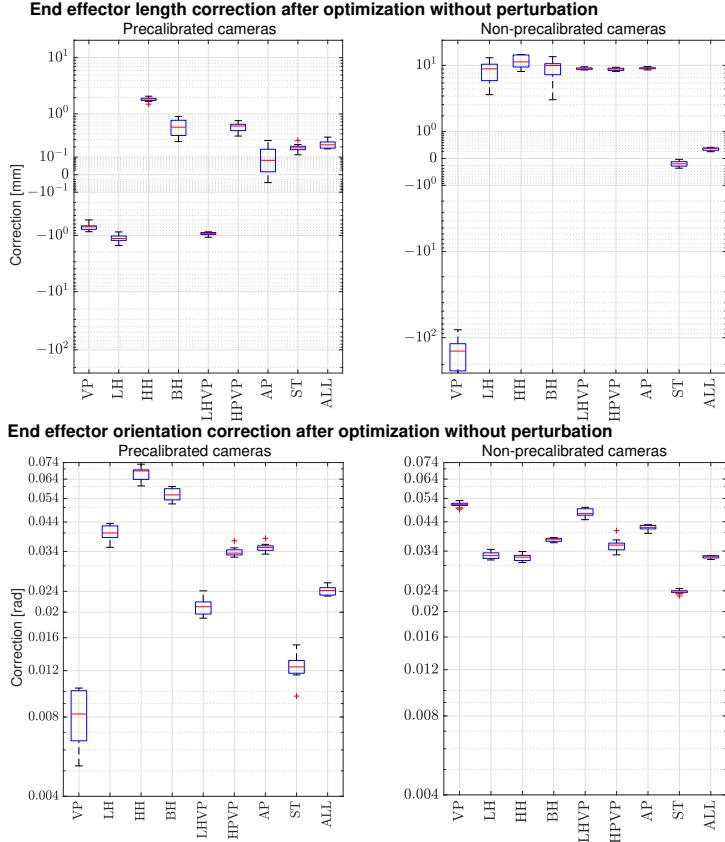
Figure 4.9: Corrections of end effector parameters after calibration of individual calibration set without perturbation with precalibrated cameras (left) or with non-precalibrated cameras (right).

4.3.2 Calibration with perturbation

The parameters were perturbed before the calibration. The perturbations were taken from a uniform distribution. Similar to [17], we chose the same three distinct levels of perturbation symmetric ranges (0.01 m, 0.03 m and 0.1 m). As before, we started with the calibration without the self-observation, so we calibrated only the end effector length. From Figure 4.10 showing the correction of the end effector length for different levels of perturbation, it can be concluded that the calibration using only the vertical plane does not work at all. The touch distance and plane distance errors of individual sets after optimization for different levels of perturbation can be found in Figure 4.11.

We can observe that the calibration by self-touch can deal with the slight perturbation and the parameter correction is the same as without perturbation then. The touch distance errors after slightly perturbed optimization are close to errors after optimization without perturbation and even to errors after moderate perturbed optimization. However, the parameter correction after the moderate perturbation has a median of 10 cm for self-touch, which is too far from the initial value, and the parameter value would not be suitable. The intensive perturbation has the same parameter correction as the median for moderate perturbation in each repetition while reducing the touch distance error mean by

The most remarkable thing comes up when the plane distance errors after optimization for different perturbation levels are compared. It can be seen that the errors are very similar, even compared to the calibration without perturbation. If we look at the parameter correction, it is almost the same for all perturbation ranges and without them as well. It seems that the calibration by a planar constraint is immune to any perturbation of the initial parameter.

Corrections after optimization of end-effector length without cameras
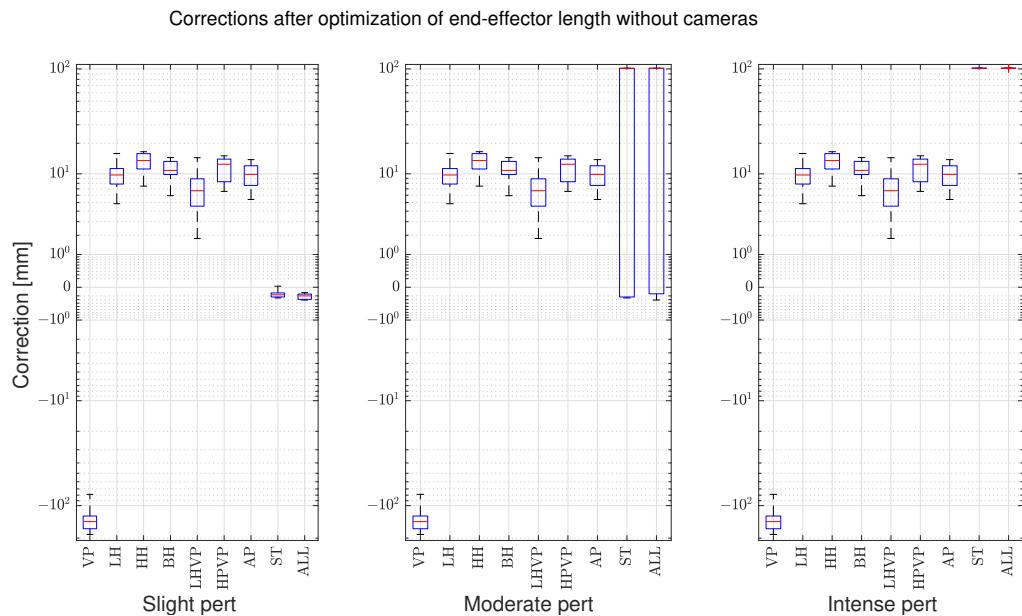
Figure 4.10: Corrections after calibration of the end effector length of individual calibration set without the self-observation. Results for 3 different values of initial perturbation of nominal value (0.354 m) are compared.
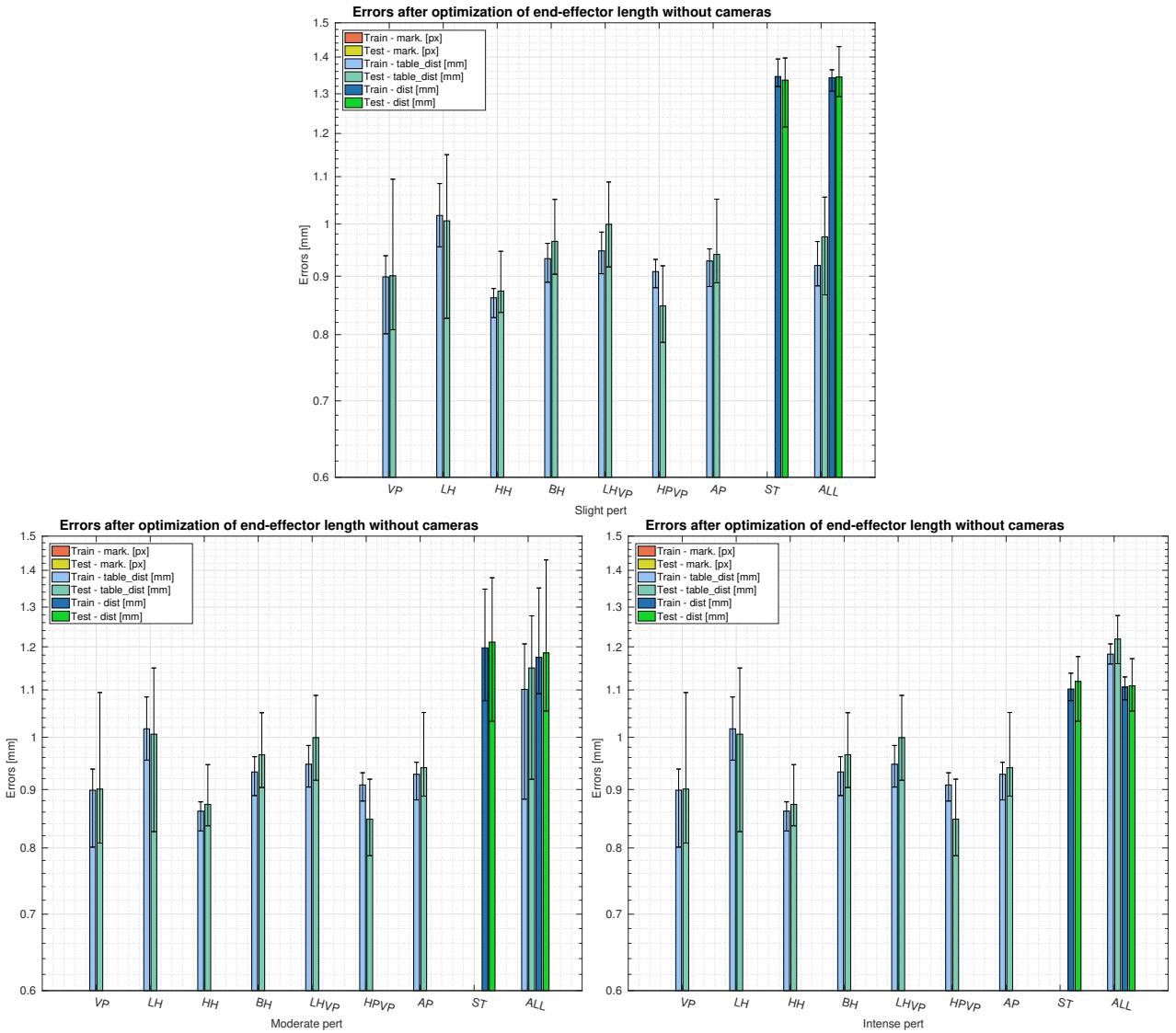
Figure 4.11: Errors after calibration of the end effector length of individual calibration set. Results for 3 different values of initial perturbation of nominal value (0.354 m) are compared. We show the absolute errors of icosahedrons distances (in mm) and distances between touches and the computed planes over train and test dataset.

Then we added the information from the self-observation and perturbed the camera DH parameters as well, but as we can see in Figure 4.2 the corrections are high enough, so the initial DH parameters worked as a perturbation too. The joint calibration is compared to the sequential one. From Figure 4.13, where errors after calibration are shown, it can be concluded that the errors after the sequential calibration stayed the same for different levels of perturbation except for the self-touch with intense perturbation. In that case, they are about ten times higher for markers and twice higher for the position. Errors after joint calibration are the same over different perturbations for all setups except the combination of self-touch and planar constraints because these errors are higher already with moderate perturbation. As can be seen, the sequential calibration has better results of projections

errors on most of the setups, only the errors for the vertical plane are worse than the same errors after the joint calibration. On the other hand, the distance errors are almost the same for both types except a few cases influenced by a significant perturbation.

Figures 4.12 and 4.14 show the corrections of the length and orientation of end effector. The corrections for the case with the precalibrated cameras correspond with the errors described above. The corrections are the same for all setups and all perturbation levels except the self-touch with intense perturbation, which has a length correction with a median of 9 cm and a rotation with a median of almost -3 rad. On the other hand, the corrections of the end effector for joint calibration are compensated by the correction of camera DH parameters as it was already mentioned in previous section. It can be seen, that the corrections of the end effector length are similar to the corrections of the end effector length after the calibration without self-observation information.
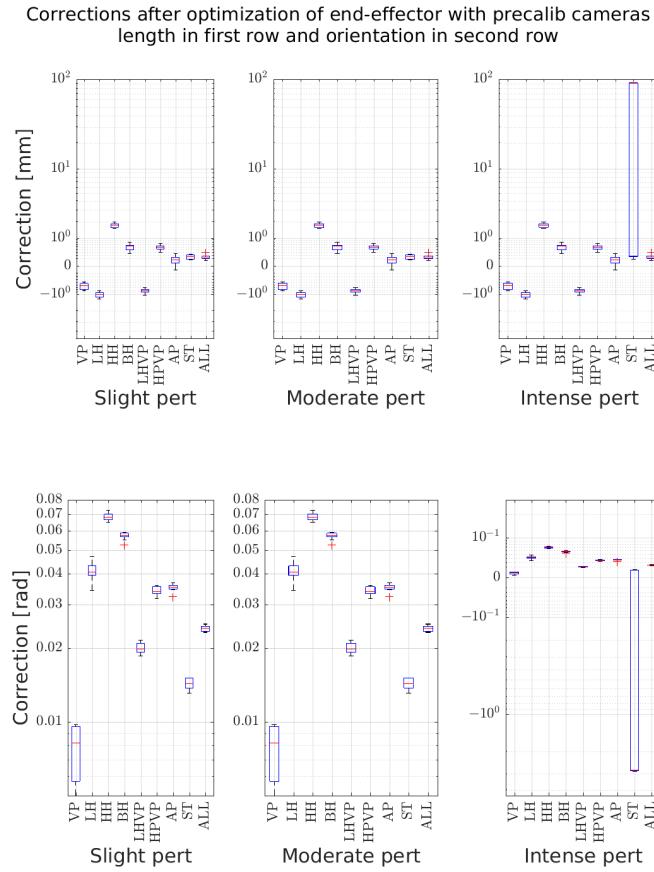
Figure 4.12: Corrections after calibration of the end effector length with non-precalibrated cameras. Results for 3 different values of initial perturbation of nominal value (0.354 m) are compared.
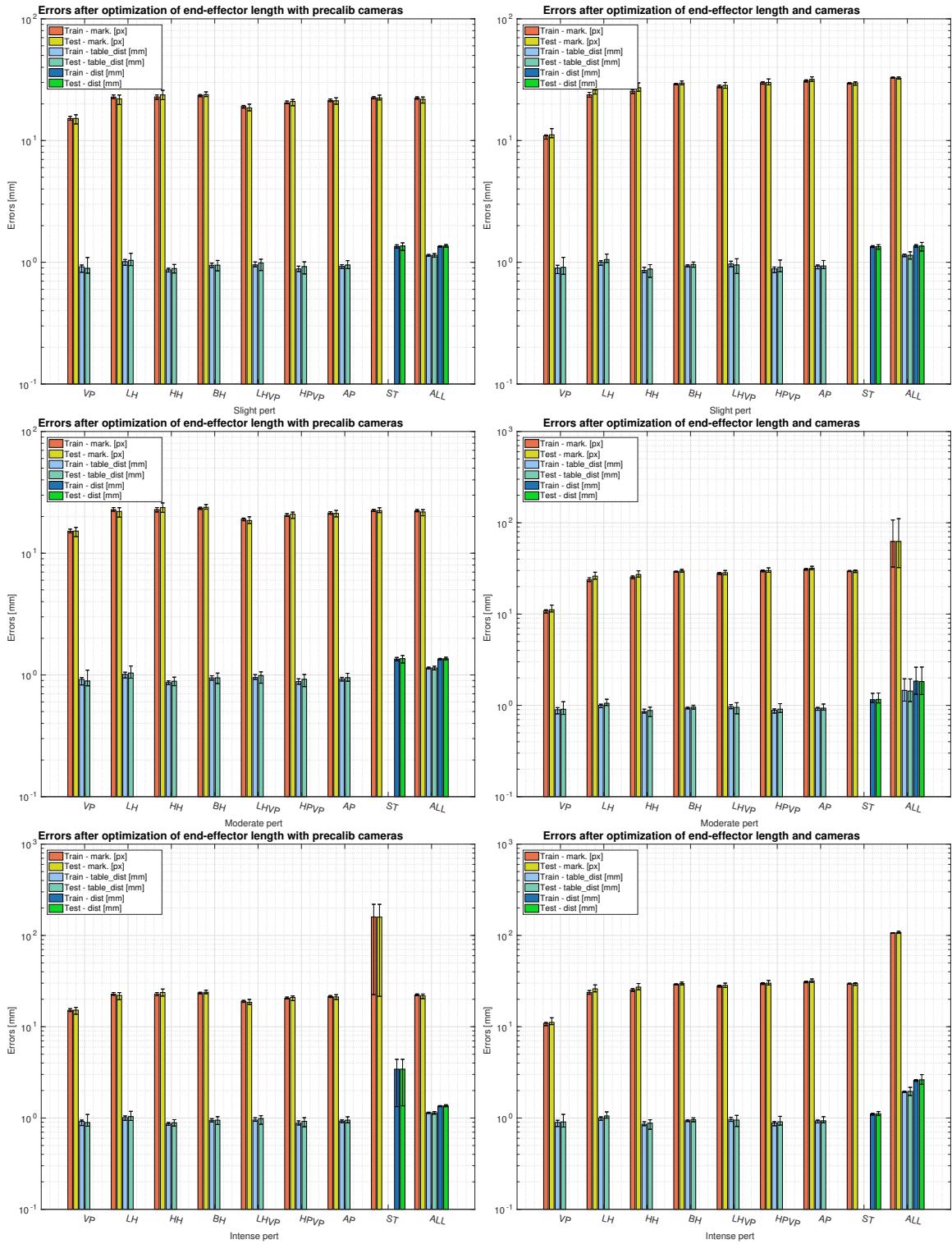
Figure 4.13: Errors after calibration of the end effector length of individual calibration set. Results for 3 different values of initial perturbation of nominal value (0.354 m) are compared. We show absolute errors of markers distances in camera frame (in pixels), icosahedrons distances (in mm) and distances between touches and the computed planes over a train and a test dataset. The sequential calibration in the first column and the joint calibration in the second column.
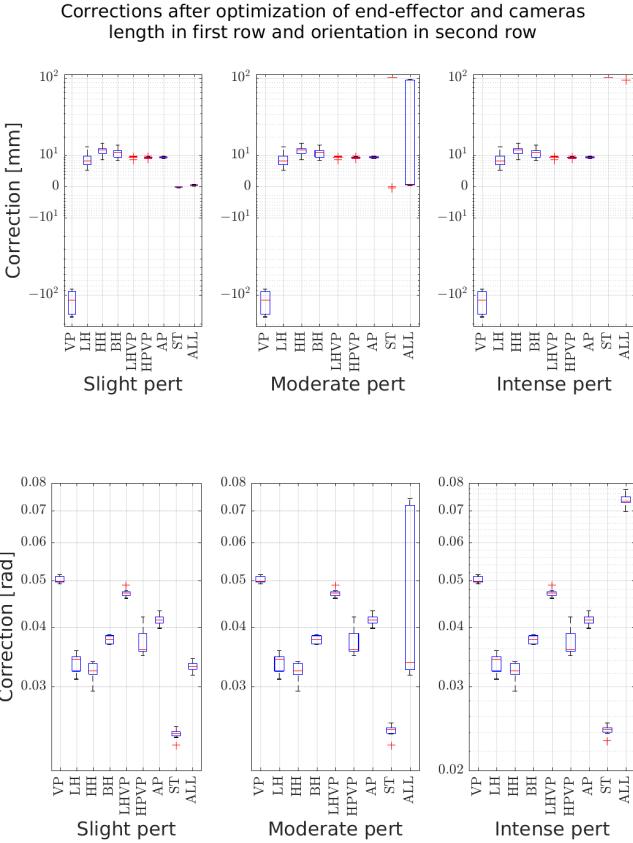
Figure 4.14: Corrections after calibration of the end effector length with non-precalibrated cameras. Results for 3 different values of initial perturbation of nominal value (0.354 m) are compared.

4.3.3 Self-touch approaches

In the self-touch experiment where the two end-effectors come into the contact, we supposed, that one of the arms has known parameters and we estimated parameters only of one of the arms (in our case right arm). The problem is that in the case of unknown end-effectors attached to both arms, one end-effector can compensate the other one. To deal with this problem, both end-effectors have to be calibrated in parallel. We tested calibration under assumption that both end-effectors have the same length compared to the case, where the length of end-effector on both arms can differ. In Figure 4.15 can be seen, that without the information from cameras, the orientation cannot be calibrated, as we already know from previous sections.

The calibration of both end effectors minimized the touch distance error more than the original approach (where one end-effector was set to the nominal value), but the correction is almost 5 cm, and for the left arm even higher for unperturbed parameters. This problem can be eliminated by incorporating information from cameras (self-observation) as it can be seen in Figure 4.15, the corrections are less than 1 mm; on the other hand, the errors

are similar to other approaches. The approach of same end effectors can deal with the slight and moderate perturbation having no correction (without cameras) or less than 1 mm correction (with cameras). However, the intense disturbance makes the corrections oscillate between no correction and 25 cm correction, and the errors have a corresponding range as well. If we compare the error graphs, we can see that the errors except intense perturbation are similar for both methods (using and not using cameras). The correction of the orientation makes smaller corrections of the length.
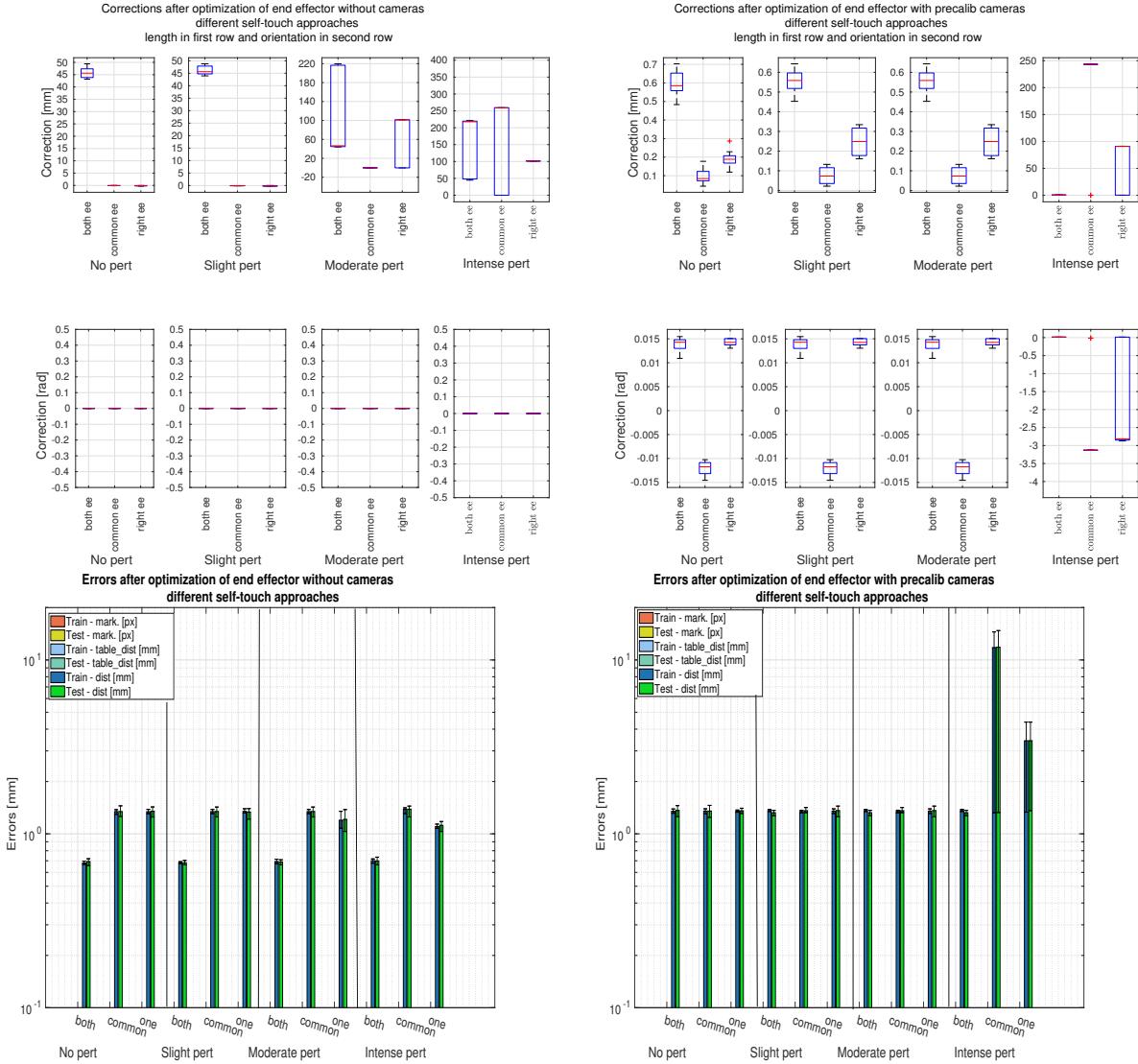
Figure 4.15: Errors (second row) and corrections (first row) after self-touch calibration of the end effector length and orientation using different methods (left without cameras and right with precalibrated cameras). Results for 3 different values of initial perturbation of nominal value (0.354 m) are compared with the unperturbed calibration. We show absolute errors of icosahedrons distances (in mm) over train and test dataset.

4.4 All DH calibration

The main task was the calibration of all possible robot DH parameters. As in the case end-effector calibration, we compare the individual sets of calibration based on the errors after optimization and the parameters correction. We take the second arm as ground truth for the self-touch constraint. At first, we checked the identifiability of parameters, Figure 4.16 shows the identifiability of robot DH parameters. It can be seen, that the end effector orientation is unidentifiable in the case of calibration without cameras, for the same reason as in the end effector calibration (see Section 4.3), so we omitted the orientation from the calibration. The problem now has comparable observability indices over setups, and the calibration is faster. The identifiability of the end effector orientation is a problem even when the self-observation information is included; it can be seen in Figure 4.16. The observation indices are highest for the combination of all three planes and self-touch. However, the self-touch only has some of the worst indices (see Table 4.7). As it was expected, merging planes increases the observability of the problem.

without cameras

setup	with end effector orientation				without end effector orientation			
	O_1	O_2	O_3	O_4	O_1 $[10^7]$	O_2 $[10^{-8}]$	O_3 $[10^5]$	O_4 $[10^{-3}]$
VP	0	0	0	0	0.0004422	0.000401	0.0000051	0.000000002
LH	0	0	0	0	0.2109	0.7416	0.04801	0.0356
HH	0	0	0	0	0.05546	0.04798	0.004114	0.0001974
BH	0	0	0	0	0.1067	0.05147	0.007944	0.0004089
LHVP	0	0	0	0	0.3666	2.676	0.23	0.6157
HPVP	0	0	0	0	0.4956	6.635	0.5951	3.949
AP	0	0	0	0	0.7593	5.847	0.8132	4.755
ST	0	0	0	0	0.5325	0.9613	0.9884	0.9502
ALL	0	0	0	0	8.129	1.148	1.161	1.333

with cameras

setup	precalibrated cameras				non-precalibrated cameras			
	O_1 $[10^8]$	O_2 $[10^{-5}]$	O_3 $[10^7]$	O_4 $[10^2]$	O_1 $[10^8]$	O_2 $[10^{-5}]$	O_3 $[10^7]$	O_4 $[10^2]$
VP	0.1685	0.1884	0.08758	0.0165	0.04536	0.03307	0.015	0.000496
LH	0.6189	1.241	0.9427	1.169	0.1959	0.2081	0.1508	0.03139
HH	0.6982	1.111	1.081	1.2	0.2274	0.2066	0.2022	0.04177
BH	1.018	1.523	2.526	3.847	0.3629	0.3768	0.643	0.2423
LHVP	1.109	2.111	1.936	4.087	0.4624	1.439	1.263	1.818
HPVP	1.151	1.864	2.091	3.897	0.4648	1.704	1.891	3.222
AP	1.516	1.882	3.456	6.505	0.6166	1.644	3.022	4.968
ST	0.2814	0.001749	0.0169	0.00002957	0.2498	0.002154	0.01958	0.00004219
ALL	3.029	0.524	5.824	3.052	1.4	0.4261	4.756	2.027

Table 4.7: Values of different observation indices for individual calibration setups for all DH calibration.
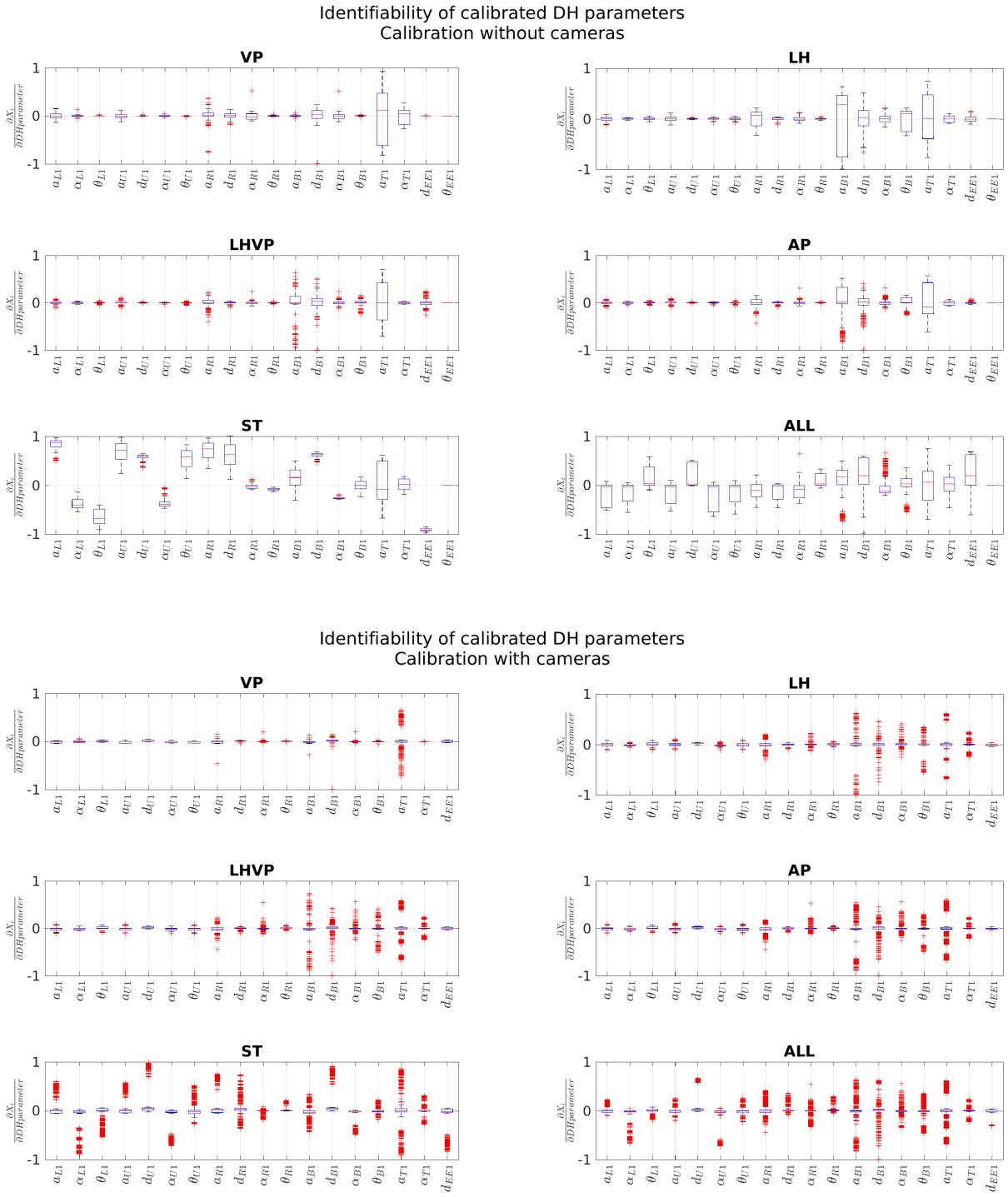
Figure 4.16: Identifiability of DH parameters for calibration without cameras (up) and with cameras (down) in selected calibration set.

49

■ **4.4.1** **Calibration without perturbation**

We started from the nominal DH parameters, and the calibration was made on all nine setups (see Table 4.1). We compared three approaches; at first, we calibrated the robot parameters using only touching constraints without the information from cameras. The second approach was the calibration with precalibrated cameras (for precalibration see Section 4.2), and the last one was the calibration robot and cameras DH parameters together.

Figure 4.19 shows the errors after the calibration, and we can observe that the calibration with the self-observation information has almost the same errors for the joint and the sequential calibration. The calibration without cameras has lower distance errors than other approaches, especially after the calibration using a vertical plane. However, as it can be seen in Figure 4.17, the final DH parameters after calibration without cameras are non-sense, because the corrections are too high (in decimetres) for a few parameters (see Figure 4.18). From the same Figure, it can be concluded that only the sequential calibration found corresponding DH parameters. Figure 4.18 shows the corrections of DH parameters for selected setups after calibration with precalibrated cameras and the corrections after Leica tracker calibration as a comparison. We chose this calibration way based on Figure 4.17; corrections for other ways can be found in Appendix (see Section A.1). It can be seen that the corrections after calibration using all three planes and self-touch has the lowest variability, but the values do not correspond with the corrections from Leica tracker. On the other hand, the corrections after self-touch are widely spread and significant, thus not suitable.
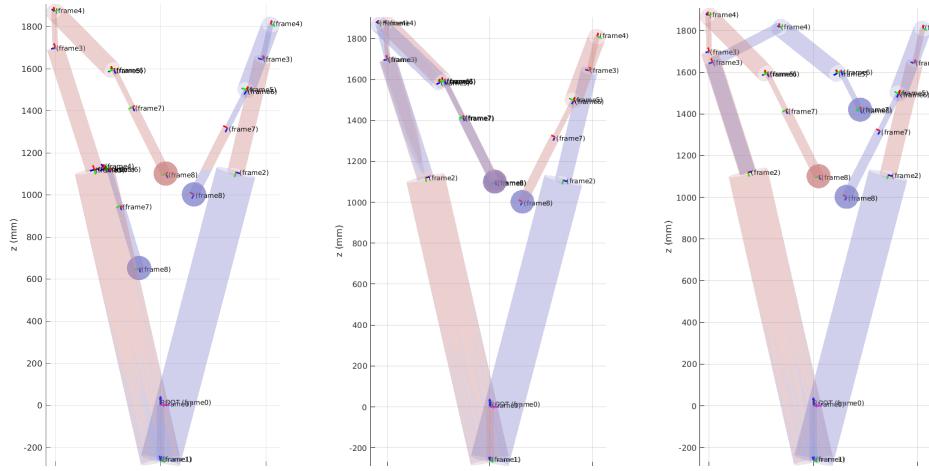
Figure 4.17: Robot visualisation after the calibration using vertical plane. The selected DH parameters had the median error on test dataset. Only right arm was calibrated. Comparison of nominal (red visualisation) and calibrated (blue visualisation) parameters. Calibration without cameras (left), sequential calibration (center) and joint calibration (right).
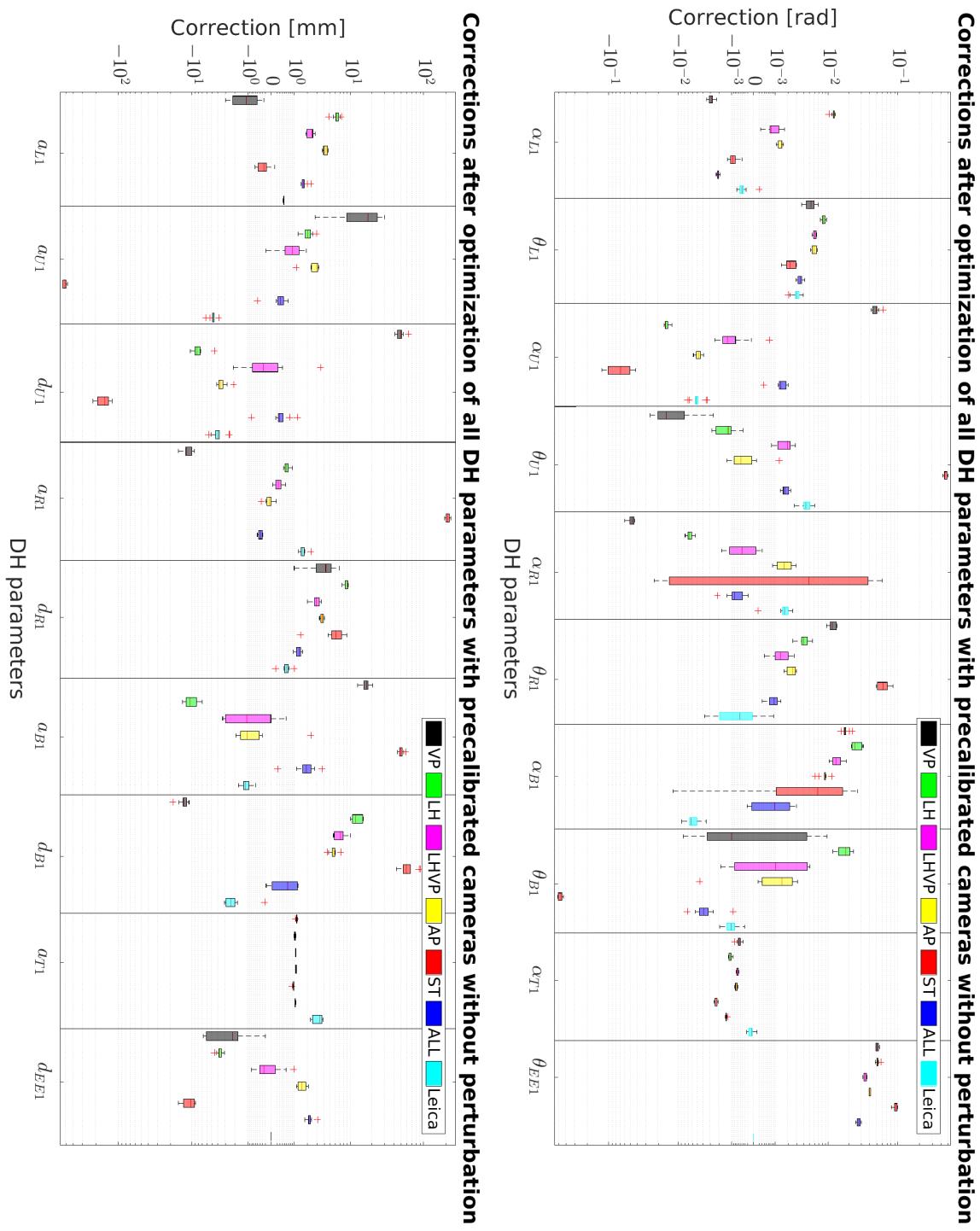
Figure 4.18: Corrections of DH parameters after calibration in selected setups and its comparison with the corrections after Leica calibration.

51
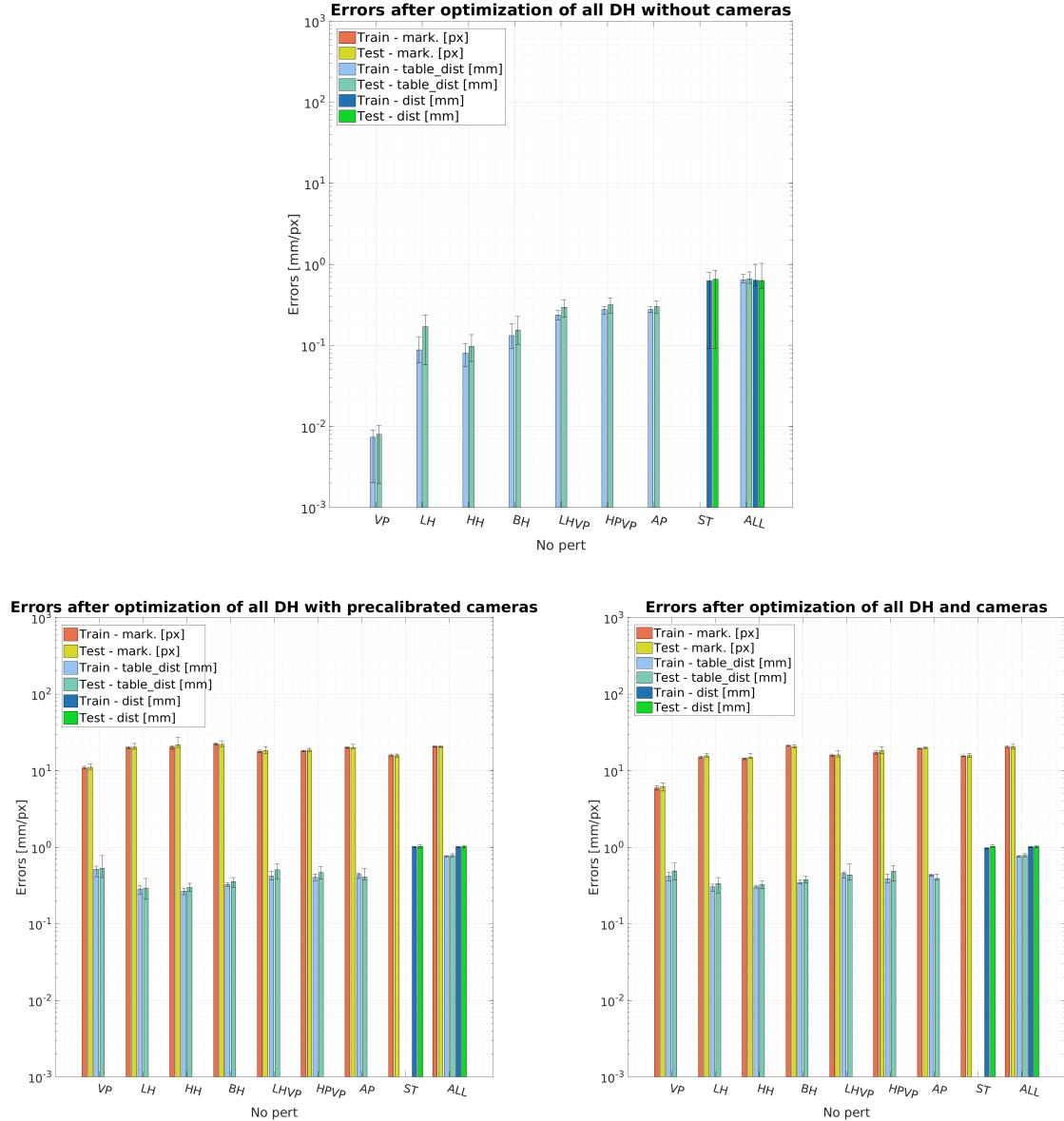
Figure 4.19: Errors after calibration of all DH of individual calibration set. Different types of calibration are shown - without cameras (left), with precalibrated cameras (center) and together with cameras (right). We show absolute errors of markers distances in camera frame (in pixels), icosahedrons distances (in mm) and distances between touches and the computed planes over a train and a test dataset.

4.4.2 Calibration with perturbation

The parameters were perturbed before the calibration. The perturbation ranges were the same as before. Since the calibration without the self-observation information has unsatisfactory DH parameters corrections (see Section 4.4.1 and Figure 4.17), we omitted this calibration type from further exploration. We compared only the sequential and

the joint calibrations with cameras then. Figure 4.20 shows the errors after calibrations. The joint calibration has smaller projections and distance errors in most cases for slight and moderate perturbation. The intense perturbation is influenced by a few very high projections errors for both ways of calibration ($> 10^4$). If we look at the figure in more detail, we can see that the difference between joint and sequential calibration is the maximal error values. The minimal error values are the same, but the maximal ones are various, and they influence the displayed mean. For that reason, we can say that both ways of calibration are able to find the same minimum during the optimization.

The corrections for slight and moderate perturbations can be found in Appendix (see Section A.2). The corrections after both perturbations levels are very similar for sequential calibration; only the moderate one has extra outliers. However, they differ from the corrections after calibration without perturbation. The corrections for joint calibration are influenced by the corrections of cameras, so the corrections after moderate perturbation have a significant spread compared to the ones after slight perturbation.
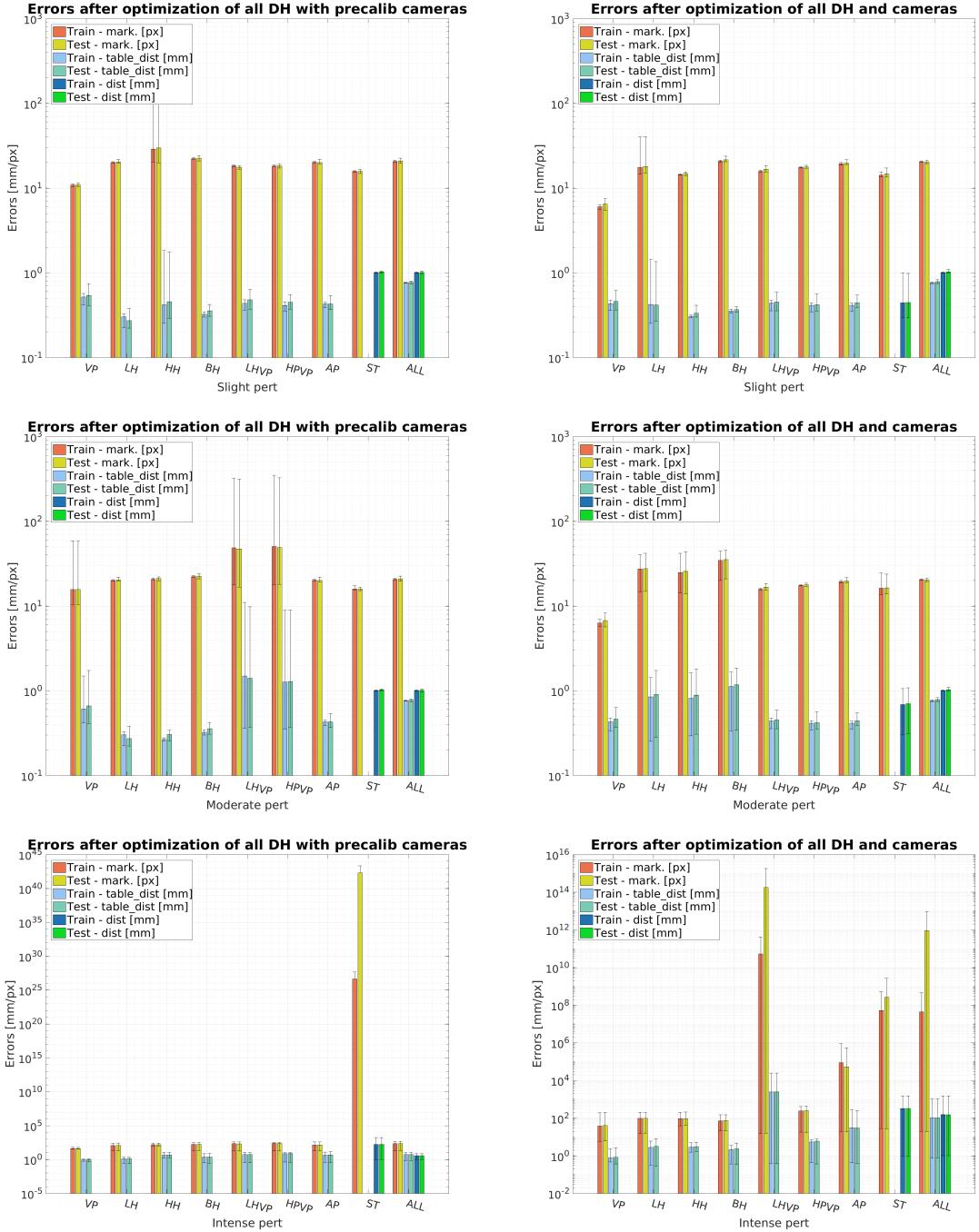
Figure 4.20: Errors after calibration of the of all DH of individual calibration set. Results for 3 different values of initial perturbation of nominal values are compared. We show absolute errors of markers distances in camera frame (in pixels), icosahedrons distances (in mm) and distances between touches and the computed planes over a train and a test dataset. The sequential calibration in the first column and the joint calibration in the second column.

■ 4.4.3 Comparison on Leica dataset

As a final comparison of all setups and calibration by Leica tracker, we decided to choose errors on Leica tracker dataset. We took parameters with the smallest errors on testing datasets for each setup from Section 4.4.1 and computed the error of distances as it was defined in Section 2.4.1. Since this comparison approach uses the Leica ball reflector as an end effector, the results can be distorted. This method does not compare, what we optimized (the error on the end effector), but the error on the link ante the end effector.

Figure 4.21 shows the RMS errors for each setup. The nominal DH parameters have better results than our calibration approaches. The smallest error is after the calibration by all planes combined with the self-touch and the non-precalibrated cameras, followed by the same method but with the precalibrated cameras. These errors are almost twice as big as the nominal ones, but they are less than one-third of the error computed from the parameters gained from the previous calibration [16]. Except for these two setups, four more setups are having RMS error lower than 1 mm. We can see, the self-observation information dramatically reduces the error; on the other hand, it does not matter whether the cameras are precalibrated or not. The DH parameters of a few selected combinations of approaches and setups can be found in Table 4.8 (for the others see Appendix B). We can notice that the corrections for best setups (based on Figure 4.21) are less than 1 cm and 0.05 rad. On the other hand, the self-touch setup has corrections even about 0.5 m and 0.5 rad.
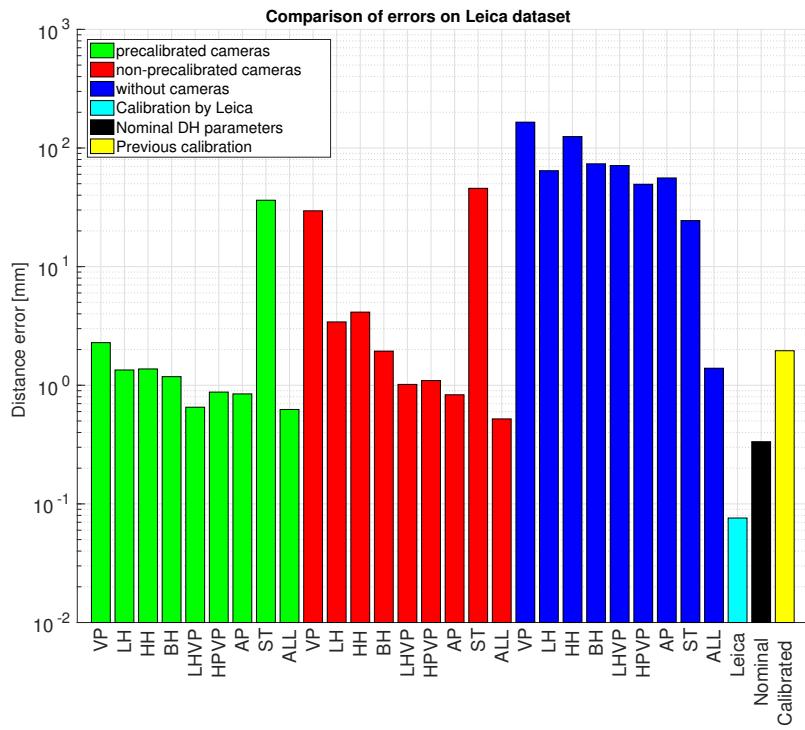
Figure 4.21: Distance error on the independent (Leica) dataset after calibration of all DH. Our setups are compared with the error for DH parameters from Leica, the nominal DH parameters and the values from the previous calibration [16].

Manipulator 1 (right arm)

setup	i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
ALL Precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.61571 (0.00171)	0 (0)	3.1396 (-0.00203)	-1.5672 (0.00362)
	U1	0.20046 (0.00046)	0.00042 (0.00042)	-1.569 (0.00176)	0.00185 (0.00185)
	R1	-0.00032 (-0.00032)	-0.6389 (0.0011)	1.5687 (-0.00213)	0.00118 (0.00118)
	B1	0.03114 (0.00114)	0.00123 (0.00123)	1.5738 (0.00296)	-1.5734 (-0.00261)
	T1	0.00106 (0.00106)	0.2 (0)	-0.00143 (-0.00143)	0 (0)
	EE1	0 (0)	0.35609 (0.00209)	0 (0)	0.02504 (0.02504)
ALL Non-precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.61594 (0.00194)	0 (0)	3.14 (-0.0016)	-1.5702 (0.00056)
	U1	0.19988 (-0.00012)	0.00242 (0.00242)	-1.5684 (0.00239)	0.00086 (0.00086)
	R1	-0.00023 (-0.00023)	-0.63936 (0.00064)	1.5688 (-0.00196)	0.0008 (0.0008)
	B1	0.03368 (0.00368)	0.00222 (0.00222)	1.5757 (0.00494)	-1.578 (-0.00722)
	T1	0.00108 (0.00108)	0.2 (0)	-0.00144 (-0.00144)	0 (0)
	EE1	0 (0)	0.35813 (0.00413)	0 (0)	0.02865 (0.02865)
ALL Without cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.6143 (0.0003)	0 (0)	3.1406 (-0.00102)	-1.5707 (6e-05)
	U1	0.19689 (-0.00311)	0.01193 (0.01193)	-1.5573 (0.01352)	0.00668 (0.00668)
	R1	-0.0013 (-0.0013)	-0.63743 (0.00257)	1.5678 (-0.00297)	-0.00673 (-0.00673)
	B1	0.02315 (-0.00685)	-0.0078 (-0.0078)	1.5584 (-0.01236)	-1.5584 (0.01235)
	T1	0.0011 (0.0011)	0.2 (0)	-0.00153 (-0.00153)	0 (0)
	EE1	0 (0)	0.3493 (-0.0047)	0 (0)	0 (0)
ST Precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.61376 (-0.00024)	0 (0)	3.1406 (-0.00097)	-1.568 (0.00283)
	U1	-0.34574 (-0.54574)	-0.18967 (-0.18967)	-1.6747 (-0.10393)	0.49334 (0.49334)
	R1	0.21862 (0.21862)	-0.63109 (0.00891)	1.6085 (0.03775)	0.04922 (0.04922)
	B1	0.08164 (0.05164)	0.06625 (0.06625)	1.5588 (-0.01204)	-2.088 (-0.51718)
	T1	0.00094 (0.00094)	0.2 (0)	-0.00249 (-0.00249)	0 (0)
	EE1	0 (0)	0.34074 (-0.01326)	0 (0)	0.09622 (0.09622)
LHVP Precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.61595 (0.00195)	0 (0)	3.1423 (0.00074)	-1.5643 (0.00649)
	U1	0.20134 (0.00134)	0.00042 (0.00042)	-1.5709 (-8e-05)	0.00211 (0.00211)
	R1	0.00015 (0.00015)	-0.63694 (0.00306)	1.569 (-0.00178)	0.00236 (0.00236)
	B1	0.02957 (-0.00043)	0.0078 (0.0078)	1.586 (0.01521)	-1.571 (-0.00016)
	T1	0.00111 (0.00111)	0.2 (0)	-0.00059 (-0.00059)	0 (0)
	EE1	0 (0)	0.35369 (-0.00031)	0 (0)	0.03481 (0.03481)

Table 4.8: DH parameters of selected setup chosen for final comparison, the differences from nominal ones are written in brackets, gray parameters were not calibrated.

Chapter 5

Conclusion and Discussion

5.1 Conclusion

This work followed up on [17], and as a first step, the robot platform was upgraded: we installed a new end effector beam from lead (see Figure 2.2) and new cameras with higher resolution but without the ability to keep the settings of focus after a restart. After that, the cameras were calibrated to gain their intrinsic parameters. With the new experimental setup, we collected a comprehensive dataset (available in our Google drive [18]). First, we modified the data collecting program to collect records for different experiments. These experiments were: the self-touch experiment, the horizontal plane touch experiment, the vertical plane touch experiment. A part of all of these experiments was captured by Leica tracker. Above that, we collected a dataset for the calibration by Leica tracker used for a final comparison as well. Once the dataset was transformed into a new structure, we were prepared to carry out the calibration of DH parameters. The dataset made us define new objective functions for optimization. We formulated optimization using planar constraints as minimizing distances between points and the fitted plane. Since we chose the data from Leica tracker as our ground truth, we prepared a way how to compare two different sets of 3D points based on [27], and it was used as an objective function in calibration by Leica tracker.

The calibration by Leica tracker was our first calibration task. We carried out this calibration for both arms in 100 repetitions. As final parameters, we chose the median values, because they had almost the same RMS error as the parameters with the smallest one. Following calibration task was the precalibration of cameras DH parameters. We performed the hand-eye calibration with fixed robot DH parameters to estimate the cameras ones. In the case of sequential calibration, we used the camera DH parameters with the lowest RMS error.

After that, we proceeded to the calibration of the robot. At first, we carried out sets of calibrations with only an end effector (tool) to be calibrated. This is a typical problem in industry, where different tools are attached to the robotic arm, and their length and orientation is to be calibrated. It was also a natural first step since we installed the new end-effector with unknown parameters. We noticed that without the information from cameras, the end effector orientation is not identifiable and cannot be calibrated then. We compared the corrections of the parameters without and with perturbation of initial parameters. We found out that the corrections of parameters after calibration by a plane or planes are almost the same for all perturbation ranges and without them as well. This calibration task by the planar constraint is immune to any perturbation of the initial parameters. We investigated the possibility of end effectors parameters compensation in

the self-touch experiment. The errors were almost the same for all three approaches, but the one that supposed that the end effectors have the same parameters on both arms had the smallest corrections.

The final task was the calibration of the robot all possible DH parameters. The observation indices are highest for the combination of all three planes and self-touch. On the other hand, the self-touch only has some of the worst indices. As it was expected, merging planes increases the observation indices. It can be seen that the end effector orientation is unidentifiable in case of calibration without cameras, for the same reason as before. We found out that the corrections of DH parameters after calibration without cameras were too high (in decimetres) for a few parameters in almost all setups. The calibration with the self-observation information has almost the same errors for joint and sequential calibration. The most suitable results had sequential calibration since the cameras could not compensate for the corrections of robot parameters.

As a final comparison of all setups and calibration by Leica tracker, we chose to look at the errors on Leica tracker dataset. The best calibration setups based on these errors were the calibration by all planes combined with the self-touch and the non-precalibrated cameras, followed by the same method but with the precalibrated cameras. These errors are twice as big as the nominal ones. On the other hand, they are less than one-third of the error computed from the parameters gained from the previous calibration [16].

5.2 Discussion and future work

In this section, we provide an interpretation of the results including their limitations and possibilities for future work.

First, we studied how only the tool of the robot can be calibrated using different approaches – motivated by industrial applications where tools are changed often while the robot structure is constant. From our results, it follows that tool calibration by a vertical plane is insufficient (errors of the real tool dimensions are in decimetres), except for the sequential calibration approach – this should be further explored in the future. On the other hand, the rest of planar constraints and their combinations restrict the calibration problem enough, so that the optimization finds the desired minimum even after intense perturbation of nominal parameters. This implies that tools can be calibrated even without approximate estimation of the end effector parameters.

Second, the possibilities of calibrating all DH parameters of the robot (19 parameters per manipulator) were investigated. None of the approaches (self-touch, self-observation, planar constraints) or their combinations achieved better performance in terms of an error on a testing set as compared to the nominal parameters (shown in Figure 4.21). This may be due to the fact that all the approaches (except for Leica tracker calibration) were trained on a different dataset that sampled from different areas of the robot workspace and that the end-effector was different: our calibration approaches employed the icosahedron as the tool while the testing comparison employed the Leica ball reflector pose – this link was not part of the calibration problem formulation). A possible remedy would be to calibrate all the DH parameters except for the last link to the tool. This remains as future work.

Third, calibration without self-observation information has worse results than when using cameras (see Figure 4.17). The reason is that contact constraints do not provide 6D information about poses. We can see that the cameras increased observability (see Tables 4.6, 4.7) as well. It can be concluded that the constraint in the form of the end effector orientation helps us to gain meaningful parameters. However, our camera poses (camera extrinsics – represented as additional kinematic chains with DH parameters, see Table 4.5) had to be precalibrated first using the robot nominal DH parameters and hand-eye calibration. Without precalibration, the camera DH parameters compensated the robot DH parameters, as we can see in Figure 4.9. For that reason, the corrections after sequential calibration are more suitable than after joint calibration (simultaneous calibration of all chains). This could be also solved by more accurate initial estimation of the camera poses. Also, a better value of a constant k (see Section 2.4.4) that relates projection errors (pixels) and distance errors in the robot workspace could lead to better results after joint calibration. In future work, we will also look at hand-eye calibration, and we will compare it with the other approaches.

The self-touch calibration performed relatively well in comparison to planar constraints in the absence of self-observation (Figure 4.21); however, its performance was not satisfactory when contact constraints were combined with self-observation. This may be explained by the joint angle distribution of the self-touch experiment. As can be seen in Figure 3.6, the stimulation of joints R1 and B1 may be insufficient compared to the other experiments. This was due to a very restricted workspace, where the end effectors were able to touch themselves. The inadequate stimulation of joints was to some degree present also in other experiments than self-touch.

Finally, the calibration by Leica tracker has different corrections than the calibration mentioned in [1]. As can be seen in Figure 4.3, the correction values of the previous calibration are often outside the box or even further from the median than outliers. Our results can be skewed by the inaccurate Leica reflector placement. The differences could be influenced by the distinct approach of the calibration as well. Beneš et al. [16] used two trackers, and they did not omit parameters from calibration.

In future work, the robot DH parameters could be bounded, for example by providing ranges of possible corrections, since we know that the robot links cannot prolong by decimetres. Similarly, we could limit the location of plane grids for the calibration by planar constraints because we know the approximate coordinates and size of grids. Eventually, we can set ranges for touch positions.

Bibliography

[1] V. Petrík and V. Smutný, "Comparison of calibrations for the CloPeMa robot," 2014.

[2] J. Hollerbach, W. Khalil, and M. Gautier, "Model identification," in *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.), pp. 113–138, Springer, 2nd ed., 2016.

[3] D. J. Bennett and J. M. Hollerbach, "Self-calibration of single-loop, closed kinematic chains formed by dual or redundant manipulators," in *Decision and Control, 1988., Proceedings of the 27th IEEE Conference on*, pp. 627–629, IEEE, 1988.

[4] A. Joubair and I. A. Bonev, "Kinematic calibration of a six-axis serial robot using distance and sphere constraints," *The International Journal of Advanced Manufacturing Technology*, vol. 77, no. 1-4, pp. 515–523, 2015.

[5] M. C. Koval, M. Klingensmith, S. S. Srinivasa, N. S. Pollard, and M. Kaess, "The manifold particle filter for state estimation on high-dimensional implicit manifolds," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 4673–4680, IEEE, 2017.

[6] A. Roncone, M. Hoffmann, U. Pattacini, and G. Metta, "Automatic kinematic chain calibration using artificial skin: self-touch in the iCub humanoid robot," in *Robotics and Automation (ICRA), 2014 IEEE International Conference*, pp. 2305–2312, 2014.

[7] Q. Li, R. Haschke, and H. Ritter, "Towards body schema learning using training data acquired by continuous self-touch," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pp. 1109–1114, IEEE, 2015.

[8] R. Khusainov, A. Klimchik, and E. Magid, "Humanoid robot kinematic calibration using industrial manipulator," in *2017 International Conference on Mechanical, System and Control Engineering (ICMSC)*, pp. 184–189, May 2017.

[9] K. Stepanova, T. Pajdla, and M. Hoffmann, "Robot self-calibration using multiple kinematic chains-a simulation study on the icub humanoid robot," *IEEE Robotics and Automation Letters*, 2019.

[10] O. Birbach, U. Frese, and B. Bäuml, "Rapid calibration of a multi-sensorial humanoid's upper body: An automatic and self-contained approach," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 420–436, 2015.

[11] M. Ikits and J. M. Hollerbach, "Kinematic calibration using a plane constraint," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, pp. 3191–3196, IEEE, 1997.

[12] H. Zhuang, S. H. Motaghedi, and Z. S. Roth, "Robot calibration with planar constraints," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 1, pp. 805–810, IEEE, 1999.

[13] A. Joubair and I. A. Bonev, "Non-kinematic calibration of a six-axis serial robot using planar constraints," *Precision Engineering*, vol. 40, pp. 325–333, 2015.

[14] R. Zenha, P. Vicente, L. Jamone, and A. Bernardino, "Incremental adaptation of a robot body schema based on touch events," in *IEEE International Conference on Developmental and Learning and on Epigenetic Robotics (ICDL-EpiRob)*, 2018.

[15] P. Beneš, M. Valášek, Z. Šika, V. Bauma, and V. Hamrle, "Experiments with redundant parallel calibration and measuring machine redcam," 2007.

[16] Beneš, Petr, and Petrík, Vladimír and Valášek, Michael and Šika, Zbyněk, "Calibration using two laser trackers leica at901 mr." `http://clopema.felk.cvut.cz/files/reports/Benes_CalibrationTR2014.pdf`. Accessed: May 22, 2019.

[17] F. Puciow, "Automatic self-calibration from self-observation and self-touch on a dual arm industrial manipulator," Master's thesis, FEE CTU, 2018.

[18] J. Rozlivek, "The collected datasets." `https://drive.google.com/drive/folders/1E6oWW9o8nojSVxqxj-P-wBLxkXOaGgzN?usp=sharing`. Accessed: May 22, 2019.

[19] F. Puciow, "Our gitlab repository (FEE CTU account required)." `https://gitlab.fel.cvut.cz/body-schema/code-selfcalib-motoman`. Accessed: May 22, 2019.

[20] gPhoto, "Remote controlling cameras." `http://www.gphoto.org/doc/remote/`. Accessed: May 22, 2019.

[21] OpenCV team, "Detection of ArUco markers." `https://docs.opencv.org/3.3.0/d5/dae/tutorial_aruco_detection.html`. Accessed: May 22, 2019.

[22] YASKAWA India Pvt. Ltd., "Robot optimized for arc welding motoman-va and ma series." `http://yaskawaindia.in/wp-content/uploads/2012/10/ARC-Welding-MA-VA-Series.pdf`. Accessed: May 22, 2019.

[23] Leica geosystems, "Leica absolute tracker at402." `https://www.hexagonmi.com/products/laser-tracker-systems/leica-absolute-tracker-at403`. Accessed: March 20, 2019.

[24] ISO VIM, "International vocabulary of metrology - basic and general concepts and associated terms (vim)." `https://jcgm.bipm.org/vim/en/4.26.html`. Accessed: May 22, 2019.

[25] Bob Bridges Ph.D., "How laser trackers work." `https://www.qualitydigest.com/inside/twitter-ed/how-laser-trackers-work.html`. Accessed: May 22, 2019.

[26] OpenCV team, "Camera calibration and 3D reconstruction." `https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#findcirclesgrid`. Accessed: May 22, 2019.

[27] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, pp. 698–700, Sep. 1987.

[28] T. Werner, "Optimalizace - study material of the B33OPT course (in czech)." `https://cw.fel.cvut.cz/b181/_media/courses/b33opt/opt.pdf`. Accessed: February 18, 2019.

[29] Y. Sun and J. M. Hollerbach, "Observability index selection for robot calibration," in *2008 IEEE International Conference on Robotics and Automation*, pp. 831–836, May 2008.

[30] J. Borm and C. Menq, "Experimental study of observability of parameter errors in robot calibration," in *Proceedings, 1989 International Conference on Robotics and Automation*, pp. 587–592 vol.1, May 1989.

[31] M. R. Driels and U. S. Pathre, "Significance of observation strategy on the design of robot calibration experiments," *Journal of Robotic Systems*, vol. 7, pp. 197 – 223, 04 1990.

[32] A. Nahvi, J. M. Hollerbach, and V. Hayward, "Calibration of a parallel robot using multiple kinematic closed loops," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 407–412 vol.1, May 1994.

[33] A. Nahvi and J. M. Hollerbach, "The noise amplification index for optimal pose selection in robot calibration," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 647–654 vol.1, April 1996.

Appendix A

All DH calibration - additional results

A.1 Corrections after calibration without perturbation
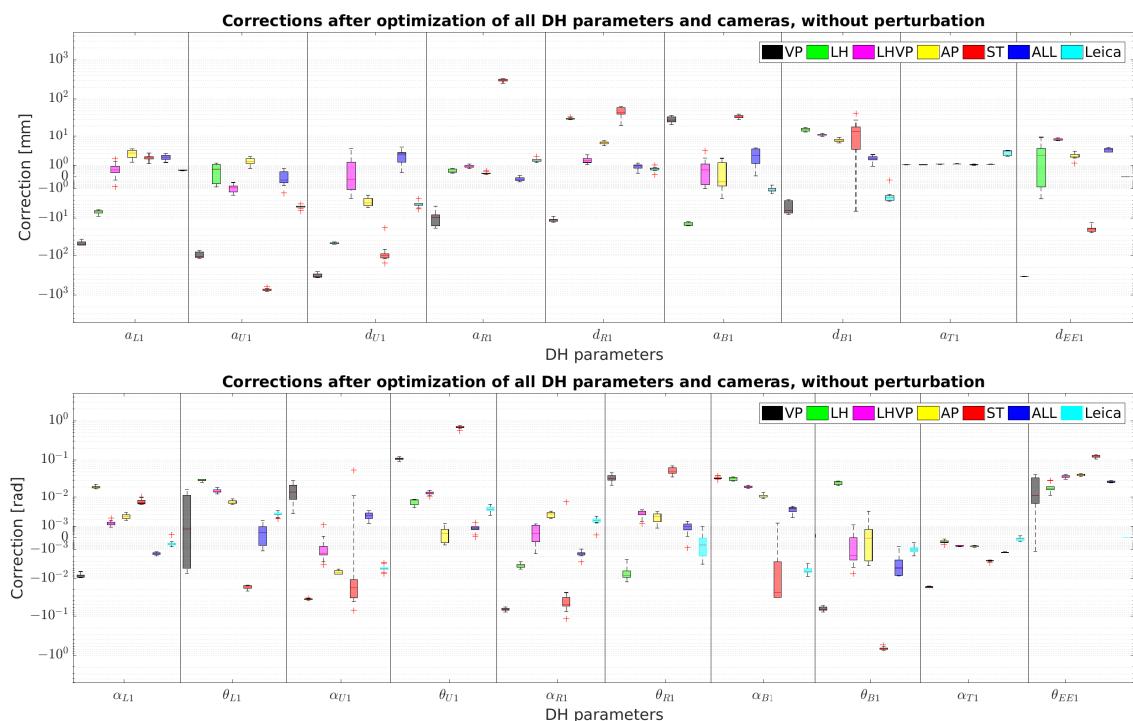
Figure A.1: Corrections of DH parameters after calibration without precalibrated cameras at all in selected setups and its comparison with the corrections after Leica calibration.
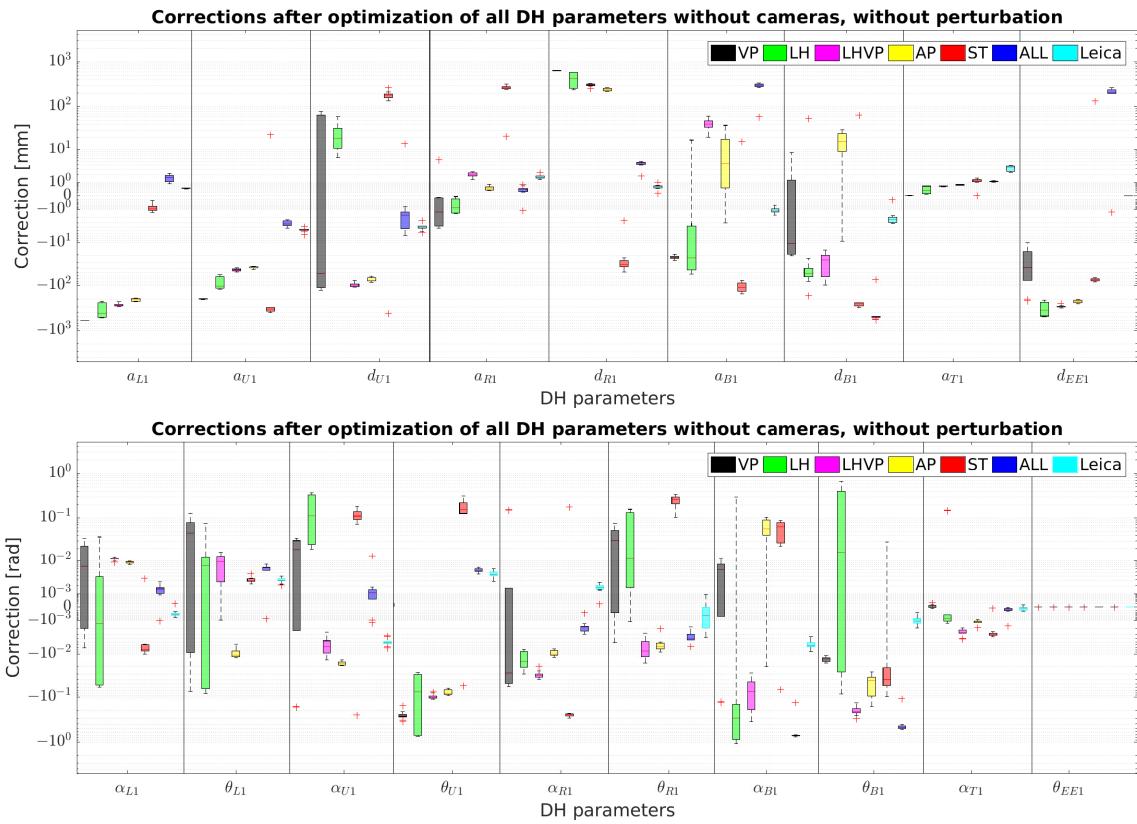
Figure A.2: Corrections of DH parameters after calibration without cameras at all in selected setups and its comparison with the corrections after Leica calibration.

A.2 Corrections after calibration with perturbation
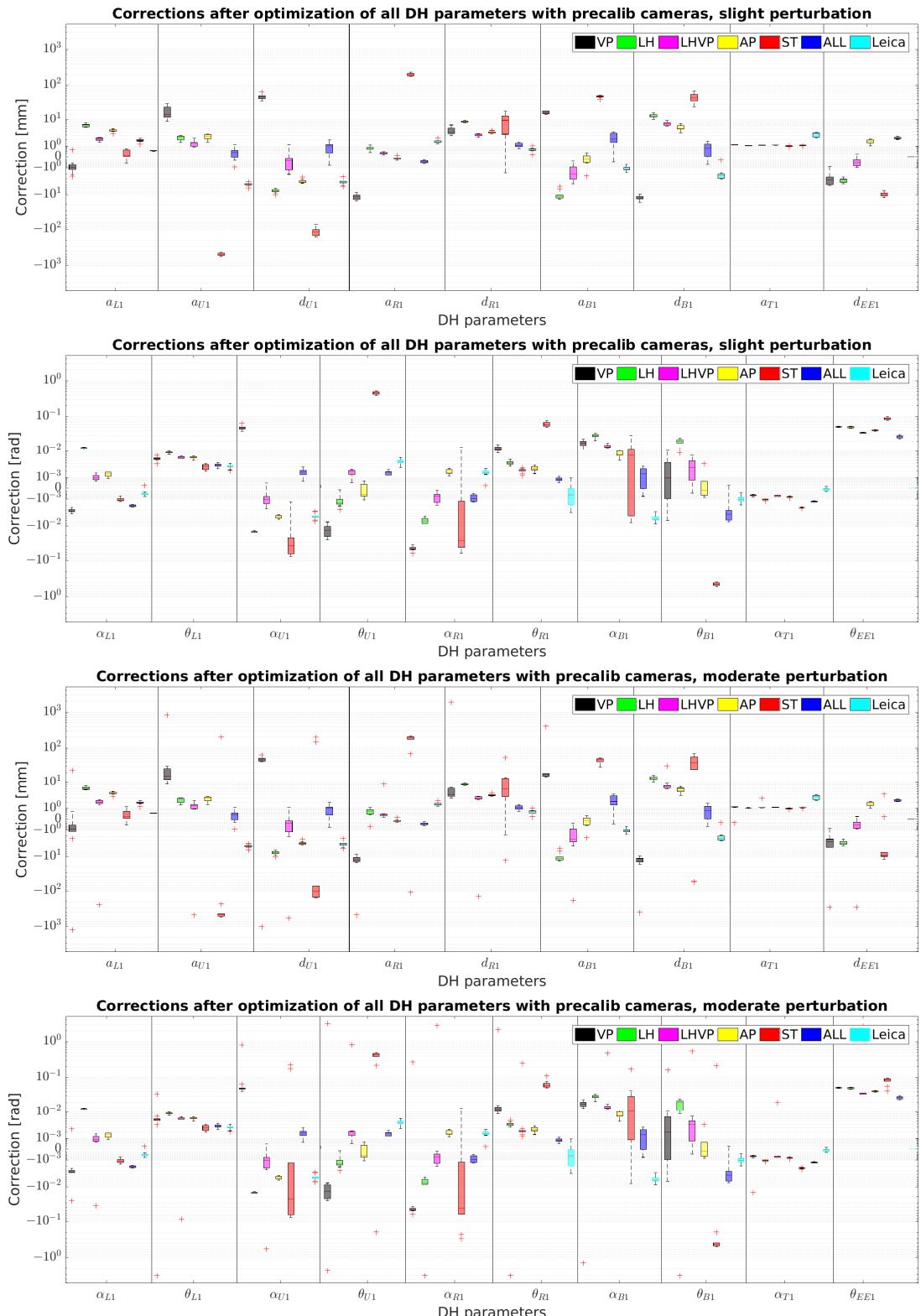
Figure A.3: Corrections of DH parameters after calibration of all DH with slight and moderate perturbation and precalibrated cameras in selected setups and its comparison with the corrections after Leica calibration (without perturbation).

Figure A.4: Corrections of DH parameters after calibration of all DH with slight and moderate perturbation and non-precalibrated cameras in selected setups and its comparison with the corrections after Leica calibration (without perturbation).

A.3 Corrections of camera DH parameters
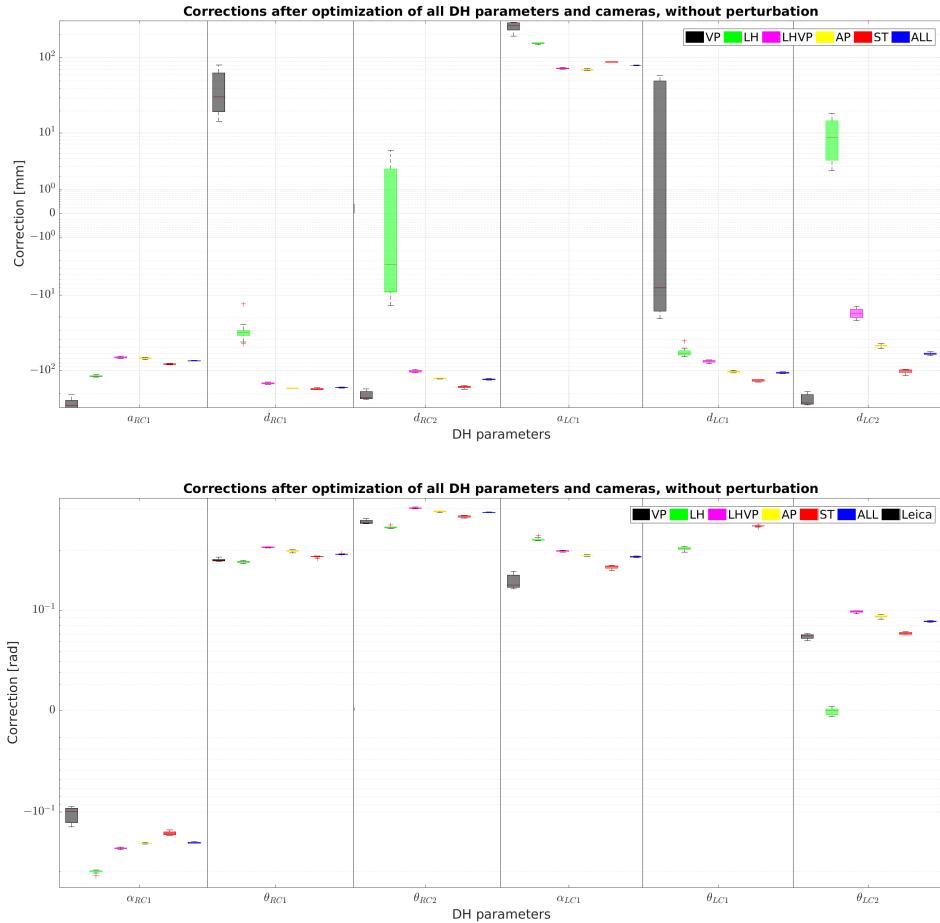
Figure A.5: Corrections of camera DH parameters after calibration of all DH without precalibrated cameras at all in selected setups.

Appendix B

Final comparision - additional results

B.1 DH parameters - calibration with precalibrated cameras

Manipulator 1 (right arm)

setup	i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
VP Precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.61288 (-0.00112)	0 (0)	3.1388 (-0.00276)	-1.5649 (0.00594)
	U1	0.21834 (0.01834)	0.04832 (0.04832)	-1.5237 (0.04713)	-0.01511 (-0.01511)
	R1	-0.01 (-0.01)	-0.63626 (0.00374)	1.5235 (-0.04726)	0.00972 (0.00972)
	B1	0.04544 (0.01544)	-0.01319 (-0.01319)	1.5882 (0.01742)	-1.5724 (-0.00158)
	T1	0.00117 (0.00117)	0.2 (0)	-0.00058 (-0.00058)	0 (0)
	EE1	0 (0)	0.35201 (-0.00199)	0 (0)	0.04855 (0.04855)
LH Precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.62024 (0.00624)	0 (0)	3.1528 (0.01123)	-1.5621 (0.0087)
	U1	0.20295 (0.00295)	-0.01061 (-0.01061)	-1.5873 (-0.01651)	-0.00229 (-0.00229)
	R1	0.00068 (0.00068)	-0.63063 (0.00937)	1.5637 (-0.00707)	0.00562 (0.00562)
	B1	0.01781 (-0.01219)	0.01376 (0.01376)	1.5996 (0.02879)	-1.5501 (0.02074)
	T1	0.00106 (0.00106)	0.2 (0)	-0.00118 (-0.00118)	0 (0)
	EE1	0 (0)	0.34945 (-0.00455)	0 (0)	0.05766 (0.05766)
HH Precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.62063 (0.00663)	0 (0)	3.142 (0.00043)	-1.5664 (0.00434)
	U1	0.20498 (0.00498)	-0.01605 (-0.01605)	-1.5877 (-0.01688)	-0.00667 (-0.00667)
	R1	-0.00091 (-0.00091)	-0.627 (0.013)	1.5682 (-0.00254)	0.00692 (0.00692)
	B1	0.02814 (-0.00186)	0.0013 (0.0013)	1.5734 (0.00264)	-1.5679 (0.00289)
	T1	0.00114 (0.00114)	0.2 (0)	-0.00127 (-0.00127)	0 (0)
	EE1	0 (0)	0.35726 (0.00326)	0 (0)	0.05569 (0.05569)

Table B.1: DH parameters of the three setups with precalibrated cameras chosen for final comparison, the differences from nominal ones are written in brackets, gray parameters were not calibrated.

Manipulator 1 (right arm)

setup	i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
BH Precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.62046 (0.00646)	0 (0)	3.1465 (0.00492)	-1.5633 (0.00753)
	U1	0.20342 (0.00342)	-0.01122 (-0.01122)	-1.5852 (-0.01438)	-0.00275 (-0.00275)
	R1	-0.00032 (-0.00032)	-0.63105 (0.00895)	1.5677 (-0.00309)	0.00408 (0.00408)
	B1	0.02649 (-0.00351)	0.00371 (0.00371)	1.5783 (0.00754)	-1.5651 (0.00573)
	T1	0.00111 (0.00111)	0.2 (0)	-0.0011 (-0.0011)	0 (0)
	EE1	0 (0)	0.35426 (0.00026)	0 (0)	0.04988 (0.04988)
HPVP Precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.61746 (0.00346)	0 (0)	3.1409 (-0.00073)	-1.5664 (0.00435)
	U1	0.20508 (0.00508)	-0.00566 (-0.00566)	-1.5767 (-0.00585)	-0.0023 (-0.0023)
	R1	-0.00012 (-0.00012)	-0.6362 (0.0038)	1.5733 (0.00254)	0.00376 (0.00376)
	B1	0.02978 (-0.00022)	0.00534 (0.00534)	1.5784 (0.00765)	-1.5704 (0.00042)
	T1	0.00111 (0.00111)	0.2 (0)	-0.00069 (-0.00069)	0 (0)
	EE1	0 (0)	0.35741 (0.00341)	0 (0)	0.04257 (0.04257)
AP Precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.61788 (0.00388)	0 (0)	3.1427 (0.00113)	-1.565 (0.00582)
	U1	0.20306 (0.00306)	-0.00386 (-0.00386)	-1.5755 (-0.00471)	-0.00078 (-0.00078)
	R1	3e-05 (3e-05)	-0.63629 (0.00371)	1.572 (0.0012)	0.00258 (0.00258)
	B1	0.02846 (-0.00154)	0.00554 (0.00554)	1.5795 (0.0087)	-1.5685 (0.00233)
	T1	0.00113 (0.00113)	0.2 (0)	-0.0007 (-0.0007)	0 (0)
	EE1	0 (0)	0.35513 (0.00113)	0 (0)	0.0402 (0.0402)

Table B.2: DH parameters of the rest of setups with precalibrated cameras chosen for final comparison, the differences from nominal ones are written in brackets, gray parameters were not calibrated.

B.2 DH parameters - calibration with non-precalibrated cameras

Manipulator 1 (right arm)

setup	i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
VP Non-precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.56318 (-0.05082)	0 (0)	3.1336 (-0.00801)	-1.5779 (-0.00713)
	U1	0.09294 (-0.10706)	-0.32856 (-0.32856)	-1.561 (0.00983)	0.10711 (0.10711)
	R1	-0.00955 (-0.00955)	-0.65063 (-0.01063)	1.5063 (-0.06454)	0.03303 (0.03303)
	B1	0.0569 (0.0269)	-0.00591 (-0.00591)	1.6035 (0.03275)	-1.6418 (-0.07101)
	T1	0.00107 (0.00107)	0.2 (0)	-0.0181 (-0.0181)	0 (0)
	EE1	0 (0)	0.01007 (-0.34393)	0 (0)	0.01415 (0.01415)
LH Non-precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.60695 (-0.00705)	0 (0)	3.1647 (0.02315)	-1.5392 (0.03161)
	U1	0.20068 (0.00068)	-0.04465 (-0.04465)	-1.6067 (-0.03593)	0.00863 (0.00863)
	R1	0.00058 (0.00058)	-0.60958 (0.03042)	1.5655 (-0.00532)	-0.00863 (-0.00863)
	B1	0.01456 (-0.01544)	0.01712 (0.01712)	1.6057 (0.03494)	-1.5448 (0.02605)
	T1	0.00102 (0.00102)	0.2 (0)	-8e-05 (-8e-05)	0 (0)
	EE1	0 (0)	0.35783 (0.00383)	0 (0)	0.01828 (0.01828)
LHVP Non-precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.61378 (-0.00022)	0 (0)	3.1431 (0.00145)	-1.5564 (0.01443)
	U1	0.19915 (-0.00085)	0.00172 (0.00172)	-1.5715 (-0.00069)	0.01292 (0.01292)
	R1	0.00097 (0.00097)	-0.63783 (0.00217)	1.5711 (0.00028)	0.00389 (0.00389)
	B1	0.03203 (0.00203)	0.01059 (0.01059)	1.5897 (0.01888)	-1.5755 (-0.00469)
	T1	0.00113 (0.00113)	0.2 (0)	-0.00064 (-0.00064)	0 (0)
	EE1	0 (0)	0.36156 (0.00756)	0 (0)	0.04219 (0.04219)
AP Non-precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.61745 (0.00345)	0 (0)	3.1437 (0.00209)	-1.5638 (0.00697)
	U1	0.20195 (0.00195)	-0.00315 (-0.00315)	-1.5771 (-0.00634)	-0.00034 (-0.00034)
	R1	0.00014 (0.00014)	-0.63389 (0.00611)	1.5742 (0.00337)	0.00257 (0.00257)
	B1	0.03182 (0.00182)	0.00654 (0.00654)	1.5799 (0.00906)	-1.5747 (-0.00387)
	T1	0.00109 (0.00109)	0.2 (0)	-0.0006 (-0.0006)	0 (0)
	EE1	0 (0)	0.35617 (0.00217)	0 (0)	0.0432 (0.0432)
ST Non-precalib cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.6161 (0.0021)	0 (0)	3.1496 (0.00799)	-1.5896 (-0.01883)
	U1	-0.54352 (-0.74352)	-0.0905 (-0.0905)	-1.5815 (-0.01074)	0.66675 (0.66675)
	R1	0.30033 (0.30033)	-0.59632 (0.04368)	1.5174 (-0.05343)	0.0584 (0.0584)
	B1	0.06423 (0.03423)	0.01355 (0.01355)	1.5576 (-0.01317)	-2.2343 (-0.6635)
	T1	0.00106 (0.00106)	0.2 (0)	-0.00269 (-0.00269)	0 (0)
	EE1	0 (0)	0.33496 (-0.01904)	0 (0)	0.12144 (0.12144)

Table B.3: DH parameters of the rest of setups with precalibrated cameras chosen for final comparison, the differences from nominal ones are written in brackets, gray parameters were not calibrated.

■ B.3 DH parameters - calibration without cameras

Manipulator 1 (right arm)

setup	i	a_i [m]	d_i [m]	α_i [rad]	θ_i [rad]
VP Without cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	3e-05 (-0.61397)	0 (0)	3.1421 (0.00052)	-1.5139 (0.05689)
	U1	-0.0001 (-0.2001)	0.09241 (0.09241)	-1.5804 (-0.00956)	-0.29967 (-0.29967)
	R1	9e-05 (9e-05)	-6e-05 (0.63994)	1.5773 (0.00647)	0.00721 (0.00721)
	B1	0.00351 (-0.02649)	-0.00374 (-0.00374)	1.562 (-0.00879)	-1.5772 (-0.00644)
	T1	2e-05 (2e-05)	0.2 (0)	-8e-05 (-8e-05)	0 (0)
	EE1	0 (0)	0.31541 (-0.03859)	0 (0)	0 (0)
LH Without cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.38327 (-0.23073)	0 (0)	3.1451 (0.00354)	-1.5635 (0.00733)
	U1	0.14067 (-0.05933)	0.00458 (0.00458)	-1.5485 (0.02226)	-0.033 (-0.033)
	R1	-0.00035 (-0.00035)	-0.39562 (0.24438)	1.5639 (-0.00694)	-0.00045 (-0.00045)
	B1	0.02451 (-0.00549)	-0.04917 (-0.04917)	1.4291 (-0.14167)	-1.5892 (-0.01841)
	T1	0.00065 (0.00065)	0.2 (0)	-0.00098 (-0.00098)	0 (0)
	EE1	0 (0)	0.13092 (-0.22308)	0 (0)	0 (0)
LHVP Without cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.35812 (-0.25588)	0 (0)	3.1516 (0.00999)	-1.5605 (0.01028)
	U1	0.15404 (-0.04596)	-0.07782 (-0.07782)	-1.5752 (-0.00436)	-0.07198 (-0.07198)
	R1	0.00175 (0.00175)	-0.3589 (0.2811)	1.5445 (-0.02629)	-0.00414 (-0.00414)
	B1	0.04927 (0.01927)	-0.03047 (-0.03047)	1.4874 (-0.08344)	-1.6837 (-0.11291)
	T1	0.00065 (0.00065)	0.2 (0)	-0.00205 (-0.00205)	0 (0)
	EE1	0 (0)	0.08272 (-0.27128)	0 (0)	0 (0)
AP Without cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.41522 (-0.19878)	0 (0)	3.1498 (0.00816)	-1.5791 (-0.00828)
	U1	0.16106 (-0.03894)	-0.06225 (-0.06225)	-1.5842 (-0.01338)	-0.07024 (-0.07024)
	R1	0.00031 (0.00031)	-0.41259 (0.22741)	1.5623 (-0.00851)	-0.00554 (-0.00554)
	B1	0.04334 (0.01334)	0.00902 (0.00902)	1.6063 (0.03547)	-1.6406 (-0.06979)
	T1	0.00077 (0.00077)	0.2 (0)	-0.00115 (-0.00115)	0 (0)
	EE1	0 (0)	0.13545 (-0.21855)	0 (0)	0 (0)
ST Without cameras	TT1	0 (0)	-0.263 (0)	0.2618 (0)	-1.5708 (0)
	S1	0.15 (0)	1.4159 (0)	-1.5708 (0)	0 (0)
	L1	0.61307 (-0.00093)	0 (0)	3.1348 (-0.00682)	-1.5679 (0.00287)
	U1	-0.13266 (-0.33266)	0.19283 (0.19283)	-1.4511 (0.11971)	0.16781 (0.16781)
	R1	0.25722 (0.25722)	-0.67676 (-0.03676)	1.3382 (-0.23261)	0.2398 (0.2398)
	B1	-0.07259 (-0.10259)	-0.25876 (-0.25876)	1.6433 (0.07248)	-1.6195 (-0.04866)
	T1	0.00126 (0.00126)	0.2 (0)	-0.00271 (-0.00271)	0 (0)
	EE1	0 (0)	0.28305 (-0.07095)	0 (0)	0 (0)

Table B.4: DH parameters of the rest of setups with precalibrated cameras chosen for final comparison, the differences from nominal ones are written in brackets, gray parameters were not calibrated.

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Rozlivek Jakub**   Personal ID number: **466263**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Automatic Kinematic Calibration of Dual Arm Manipulator Using Self-contact and Plane Constraints

Bachelor's thesis title in Czech:

Automatická kinematická kalibrace dvojrukého manipulátoru za pomoci sebedoteku a rovinných omezení

Guidelines:

1. Familiarization with and preparation of robot setup - 2x Yaskawa Motoman DX100, inverse kinematics, force/torque sensors, cameras.
2. Experiment for automated data collection in self-contact and external contact configurations (horizontal and vertical plane).
3. Optimization problem formulation for kinematic calibration including plane constraint.
4. Evaluation of calibration using non-linear least squares methods with different problem formulations (self-contact, self-observation, planar constraint).
5. Parameter observability and identifiability in different setups.
6. Comparison with calibration using external metrology (Leica).

Bibliography / sources:

[1] Birbach, O.; Frese, U. & Bauml, B. - 'Rapid calibration of a multi-sensorial humanoid's upper body: An automatic and self-contained approach' .- The International Journal of Robotics Research 34(4-5), 420—436, 2015.
[2] Joubair, A., & Bonev, I. A. - Kinematic calibration of a six-axis serial robot using distance and sphere constraints. - The International Journal of Advanced Manufacturing Technology, 77(1-4), 515-523, 2015.
[3] Khusainov, R., Klimchik, A., & Magid, E. - Humanoid robot kinematic calibration using industrial manipulator. - In Mechanical, System and Control Engineering (ICMSC), 2017 International Conference on (pp. 184-189). IEEE, 2017.
[4] Roncone, A.; Hoffmann, M.; Pattacini, U. & Metta, G. - Automatic kinematic chain calibration using artificial skin: self-touch in the iCub humanoid robot - 'Robotics and Automation (ICRA), 2014 IEEE International Conference on', pp. 2305-2312, 2014.
[5] Zenha, R.; Vicente, P.; Jamone, L. & Bernardino, A.-Incremental adaptation of a robot body schema based on touch events, - 'Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)', 2018.

Name and workplace of bachelor's thesis supervisor:

Mgr. Karla Štěpánová, Ph.D.,   Robotic Perception,   CIIRC

Name and workplace of second bachelor's thesis supervisor or consultant:

Mgr. Matěj Hoffmann, Ph.D.,   Vision for Robotics and Autonomous Systems,   FEE

Date of bachelor's thesis assignment: **24.01.2019**   Deadline for bachelor thesis submission: **24.05.2019**

Assignment valid until: **20.09.2020**

_____   _____   _____
Mgr. Karla Štěpánová, Ph.D.   doc. Ing. Tomáš Svoboda, Ph.D.   prof. Ing. Pavel Ripka, CSc.
Supervisor's signature   Head of department's signature   Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

.

_____
Date of assignment receipt

_____
Student's signature