

# Robot self-calibration using multiple kinematic chains – a simulation study on the iCub humanoid robot

Karla Stepanova<sup>1,2</sup>, Tomas Pajdla<sup>2</sup>, and Matej Hoffmann<sup>1</sup>

**Abstract**—Mechanism calibration is an important and non-trivial task in robotics. Advances in sensor technology make affordable but increasingly accurate devices such as cameras and tactile sensors available, making it possible to perform automated self-contained calibration relying on redundant information in these sensory streams. In this work, we use a simulated iCub humanoid robot with a stereo camera system and end-effector contact emulation to quantitatively compare the performance of kinematic calibration by employing different combinations of intersecting kinematic chains—either through self-observation or self-touch. The parameters varied were: (i) type and number of intersecting kinematic chains used for calibration, (ii) parameters and chains subject to optimization, (iii) amount of initial perturbation of kinematic parameters, (iv) number of poses/configurations used for optimization, (v) amount of measurement noise in end-effector positions / cameras. The main findings are: (1) calibrating parameters of a single chain (e.g. one arm) by employing multiple kinematic chains (“self-observation” and “self-touch”) is superior in terms of optimization results as well as observability; (2) when using multi-chain calibration, fewer poses suffice to get similar performance compared to when for example only observation from a single camera is used; (3) parameters of all chains (here 86 DH parameters) can be subject to calibration simultaneously and with 50 (100) poses, end-effector error of around 2 (1) mm can be achieved; (4) adding noise to a sensory modality degrades performance of all calibrations employing the chains relying on this information.

**Index Terms**—Humanoid robots, calibration and identification, force and tactile sensing, kinematics, optimization and optimal control.

## I. INTRODUCTION

ROBOTS performing manipulation tasks rely on models of their bodies and their success is largely determined by their accuracy. However, inaccuracies creep in many ways as for example in the assembly process, in mechanical elasticity, or simply because of cheap design of components. Therefore,

Manuscript received: September 10, 2018; Revised: December 11, 2018; Accepted: January 16, 2019.

This paper was recommended for publication by Editor Nikos Tsagarakis upon evaluation of the Associate Editor and Reviewers’ comments. K.S. and M.H. were supported by the Czech Science Foundation under Project GA17-15697Y; T.P. was supported by the European Regional Development Fund under the project Robotics for Industry 4.0 (reg. no. CZ.02.1.01/0.0/0.0/15/003/0000470).

<sup>1</sup>Karla Stepanova and Matej Hoffmann are with Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic. karla.stepanova@cvut.cz and matej.hoffmann@fel.cvut.cz.

<sup>2</sup>Karla Stepanova and Tomas Pajdla are with Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague, Czech Republic pajdla@cvut.cz.

Digital Object Identifier (DOI): see top of this page.

the actual model parameters of every robot exemplar have to be found by means of a calibration procedure, usually relying on external metrology systems. For kinematic calibration, such apparatuses can measure one or more of the components of the end-effector pose employing mechanical, visual, or laser systems (see [1] for a survey). Different arrangements have different accuracy, requirements on the environment, and cost. These conditions have to be present for recalibration to be performed.

Current trends in the robotics industry make classical calibration procedures less practical: with the advent of the so-called “collaborative robots”, for example, the machines are becoming cheaper, lightweight, compliant, and they are being deployed in more versatile ways according to the needs of customized production of smaller batches rather than being fixed in a single production line for their entire lifetime. All these factors increase the need for calibration to be performed more frequently. At the same time, the machines, including home and service robots, often come with richer sets of powerful sensory devices that are affordable and not difficult to operate. Both these trends speak for alternative solutions to the self-calibration problem that are more “self-contained” and can be performed autonomously by the robot.

Hollerbach et al. [1] classify different calibration methods into *open-loop*—where one or more of the components of the end-effector pose is measured employing mechanical, visual, or laser systems—and *closed-loop* where physical constraints on the end-effector position or orientation can substitute for measurements. Observing the end-effector—or in general any other points on the kinematic chain—using a camera falls into the open-loop calibration family, although components of the end-effector pose can be observed only indirectly through projection into the camera frame. Self-touch configurations employing two arms of the humanoid robot could be framed as a constraint if contact measurement only (e.g. from force/torque sensors) was available and hence treated as closed-loop. In this work, we follow up on [2] and emulate sensitive skin measurements, which provide the position of contact (and hence fit more naturally with open-loop calibration).

Our work is a simulation study that draws on calibration in the real world—like different approaches to kinematic calibration of the iCub humanoid robot relying on self-observation [3], [4] and self-touch [2]. Using the model of the robot with identical parameters, but exploiting the fact that we have complete knowledge of the system and capacity to emulate

different levels of model perturbation and measurement noise, our goal is to get insights into the pros and cons of different optimization problem formulations. In particular, we study how the calibration performance is dependent on the type and number of intersecting kinematic chains, the number of parameters calibrated, number of robot configurations, and the measurement noise. Accompanying video is available here <https://youtu.be/zP3c7Eq8yVk> and dataset at [5].

This article is structured as follows. Related work is reviewed in the next section, followed by Materials and Methods, Data Acquisition and Description, and Simulation Results. We close with a Discussion and Conclusion.

## II. RELATED WORK

We focus on humanoid robots or humanoid-like setups with many Degrees of Freedom (DoF) of two arms that can possibly self-touch, equipped with cameras and tactile or inertial sensors. These are challenging setups for calibration but they create new opportunities for automated self-contained calibration based on closing kinematic loops by touch (self-contact) and vision.

Most often, the loops are closed through self-observation of the end-effector using cameras located in the robot head (*open-loop calibration* method per [1]). Hersch et al. [6] and Martinez-Cantin et al. [7] present online methods to calibrate humanoid torso kinematics relying on gradient descent and recursive least squares estimation, respectively. The iCub humanoid was employed in [3], [4]. Vicente et al. [4] used a model of the hand’s appearance to estimate its 6D pose and used that information to calibrate the joint offsets. Fanello et al. [3] had the robot observe its fingertip and learned essentially a single transformation only to account for the discrepancy between forward kinematics of the arm and the projection of the finger into the cameras.

Next to cameras, inertial sensors also contain information that can be exploited for calibration. Kinematic calibration was shown exploiting 3-axis accelerometers embedded in the artificial skin modules distributed on robot body [8], [9] or in the control boards on the iCub [10] or CMU/Sarcos [11].

The advent of robotic skin technologies [12], [13] opens up the possibility of a new family of approaches, whereby the chain is closed through contact like in closed-loop calibration, but the contact position can be extracted from the tactile array. Roncone et al. [2] showed this on the iCub robot that performs autonomous self-touch using a finger with sensitive fingertip to touch the skin-equipped forearm of the contralateral arm; Li et al. [14] 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 robot kinematic calibration. In this sense, the skin provides a pose measurement rather than constraint and as such, 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. [15] exploit this principle using an industrial manipulator to calibrate the legs of a humanoid robot. Another variant is exploiting the sensitive fingertips to touch a known external surface [16].

Birbach et al. [17] were to our knowledge the only ones to employ truly “multisensorial” or “multimodal” calibration. Using the humanoid robot Justin observing its wrist, the error functions comparing the wrist’s position from forward kinematics with its projection into the left and right camera images, Kinect image, and Kinect disparity, together with an inertial term, were aggregated into a single cost function to be minimized. It is claimed that while pair-wise calibration can lead to inconsistencies, calibrating everything together in a “mutually supportive way” is most efficient.

In this work, we compare calibration through self-observation (with projection into cameras) and calibration through self-touch and the effect of their synergy. Our work makes a unique contribution, also compared to [17] who, first, employ essentially only “hand-eye” kinematic chains terminating in different vision-like sensors in the robot head, and, second, consider only the case where all chains are combined together using a single cost function.

## III. MATERIALS AND METHODS

### A. iCub robot kinematic model and camera parameters

In this work, we use the upper body of the iCub humanoid robot (see Fig. 1) and its kinematic model expressed in the Denavit-Hartenberg convention, where every link  $i$  is described by 4 parameters:  $\{a_i, d_i, \alpha_i, o_i\}$ . In this platform, all joints are revolute. We will consider several kinematic chains: all start in a single inertial or base frame—denoted iCub *Root* Reference Frame here. For every chain, the DH parameters uniquely define a chain of transformation matrices from the inertial frame to the end-effector. The position and orientation of the end-effector in the *Root* frame is thus given by  $\mathbf{T}_n^{Root} = A_1(q_1) \dots A_n(q_n)$  where the homogeneous transformation matrices  $A_i$  can be constructed from the DH representation and  $q_i$  are current joint angles of the robot actuators.

The links are schematically illustrated in Fig. 1. iCub kinematics version 1 was used [18] with the following modification: the *Root* was moved from the waist area to the third torso joint, which is the new inertial frame for our purposes.

The four chains under consideration are:

- 1) Left Arm (LA). DH parameters in Table I. Short names to denote the links/joints: Root-to-LAShoulder, LA Shoulder Pitch, LA Shoulder Roll, LA Shoulder Yaw, LA Elbow, LA Wrist Prosup (for pronosupination), LA Wrist Pitch, LA Wrist Yaw.
- 2) Right Arm (RA). DH parameters analogous to LA (see [18]). Link/joint names: Root-to-RAshoulder, RA Shoulder Pitch, RA Shoulder Roll, RA Shoulder Yaw, RA Elbow, RA Wrist Prosup, RA Wrist Pitch, RA Wrist Yaw.
- 3) Left Eye (LEye). DH parameters in Table II. Link/joint names: Root-to-neck, Neck Pitch, Neck Roll, Neck Yaw, Eyes Tilt, Left Eye Pan.
- 4) Right Eye (REye). DH parameters different than LEye in Table III. Link/joint names: Root-to-neck, Neck Pitch, Neck Roll, Neck Yaw, Eyes Tilt, Right Eye Pan.

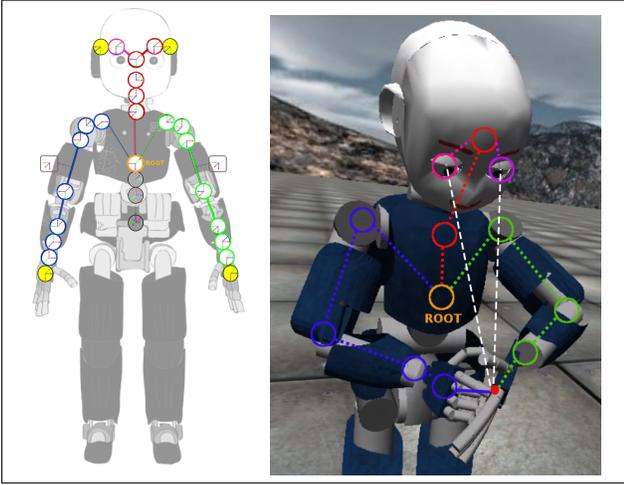


Fig. 1. iCub upper body and schematic illustration of kinematic chains considered. All chains originate in a common *Root* which is located at the third torso joint. The left and right arm chains are drawn in green and blue respectively. The eye chains have a common Root-to-head chain part marked in red. The right panel illustrates the self-calibration by connecting different chains—self-touch and self-observation. White lines denote projection into the eyes/cameras.

Links or parameters not subject to calibration are shown shaded in grey in the corresponding tables. The first link always originates in the *Root* frame and is fixed in all chains (the torso joint is not moving) and is also excluded from calibration. The alpha parameter of the last link in the arm chains is also not being calibrated as it is not observable because we observe only position and not the orientation of the end-effectors. The right arm chain is further extended with a fixed transform from the end-effector in the palm to the tip of the index finger—not subject to calibration. The eye chains differ in the last link only.

Link(i)	a(i) [mm]	d(i) [mm]	$\alpha$ [rad]	$o$ [rad]
1	23.36	143.3	$\pi/2$	$105 * \pi/180$
2	0	107.74	$-\pi/2$	$\pi/2$
3	0	0	$\pi/2$	$-\pi/2$
4	15	152.28	$-\pi/2$	$75 * \pi/180$
5	-15	0	$\pi/2$	0
6	0	137.3	$\pi/2$	$-\pi/2$
7	0	0	$\pi/2$	$\pi/2$
8	62.5	-16	0	0

TABLE I

DH PARAMETERS ( $a$ ,  $d$ ,  $\alpha$  AND OFFSETS  $o$ ) DESCRIBING ALL LINKS IN LEFT ARM KINEMATIC CHAIN.

Link(i)	a(i) [mm]	d(i) [mm]	$\alpha$ [rad]	$o$ [rad]
1	2.31	-193.3	$-\pi/2$	$\pi/4$
2	33	0	$\pi/2$	$\pi/4$
3	0	1	$-\pi/2$	$\pi/4$
4	-54	82.5	$-\pi/2$	$\pi/4$
5	0	-34	$-\pi/2$	0
6	0	0	$\pi/2$	$-\pi/4$

TABLE II

DH PARAMETERS – LEFT EYE KINEMATIC CHAIN.

The camera intrinsic parameters were taken from the real robot cameras and were not subject to calibration: resolution 320 x 240, focal length  $f_x = 257.34$ ,  $f_y = 257.34$ ,  $c_y = 120$ .

Link(i)	a(i) [mm]	d(i) [mm]	$\alpha$ [rad]	$o$ [rad]
5	0	34	$\pi/2$	$-\pi/4$
6	0	0	$-\pi/2$	0

TABLE III

DH PARAMETERS – RIGHT EYE KINEMATIC CHAIN. LINKS 1-4 SHARED WITH LEFT EYE KINEMATIC CHAIN.

### B. Optimization problem formulation

By calibration we mean estimation of the parameter vector  $\phi = \{[a_1, \dots, a_n], [d_1, \dots, d_n], [\alpha_1, \dots, \alpha_n], [o_1, \dots, o_n]\}$  with  $i \in N$ , where  $N = \{1, \dots, n\}$  is a set of indices identifying individual links;  $a$ ,  $d$  and  $\alpha$  are the first three parameters of the DH formulation and  $o$  the offset that specifies the positioning of the encoders on the joints with respect to the DH representation. We often estimate a subset of these parameters only, assuming that the others are known. This subset can for example consist of a subset of links  $N' \subset N$  (e.g., only parameters of one arm are to be calibrated) or a subset of the parameters (e.g., only offsets  $o$  are to be calibrated—sometimes dubbed “daily calibration” [19]).

The estimation of the parameter vector  $\phi$  is done by optimizing a given objective function:

$$\phi^* = \underset{\phi}{\operatorname{argmin}} \sum_{m=1}^M \|\mathbf{p}_m^r - \mathbf{p}_m^e(\phi, \Theta_m)\|^2, \quad (1)$$

where  $M$  is the number of robot configurations and corresponding end-effector positions used for calibration (hereafter, often referred to as “poses” for short),  $\mathbf{p}_m^r$  is a real (observed) end-effector position,  $\mathbf{p}_m^e$  is an estimated end-effector position computed using forward kinematic function for a given parameter estimate  $\phi$  and joint angles from joint encoders  $\Theta_m$ . For chains involving cameras, the reprojection error is used instead, as described in the next section.

### C. Kinematic chain calibration

We study different combinations of intersecting chains and their performance in calibrating one another.

1) *Two arms chain (LA-RA)*: This corresponds to the self-touch scenario, with touch occurring directly at the end-effectors (the right arm end-effector being shifted from palm to tip of index finger using a fixed transform). The newly established kinematic chain for upper body includes both arms while head and eyes are excluded. To optimize parameters describing this chain, we minimize the distance between estimated positions in 3D space of left and right arm end-effectors. In this case, the parameter vector  $\phi$  consists of the following parameters:  $\phi = \{\phi^r, \phi^l\}$ , where  $\phi^r$  and  $\phi^l$  are parameters corresponding to the robot right and left arm, respectively. The objective function to be optimized is:

$$\phi^* = \underset{\phi}{\operatorname{argmin}} \sum_{m=1}^M \|\mathbf{X}_m^{r,R}(\phi^r, \Theta_m^r) - \mathbf{X}_m^{l,R}(\phi^l, \Theta_m^l)\|^2 \quad (2)$$

where  $M$  is the number of poses used for calibration,  $\mathbf{X}_m^{r,R}$  and  $\mathbf{X}_m^{l,R}$  are the  $m^{\text{th}}$  estimated end-effector positions in the *Root* frame for the right and left arm respectively, computed using a given parameter estimate  $\phi$  and joint angles from joint encoders  $\Theta_m$ .

2) *Hand to eye chains (LA-LEye, LA-REye, RA-LEye, RA-REye)*: To predict position of the end-effector in each of the robot cameras (similar to [17]), the estimated end-effector position,  $\mathbf{X}^{Root}$ , is given by a current hypothetical robot calibration of the parameter vector  $\phi$  and is computed via forward kinematics.  $\mathbf{X}^{Root}$  is then mapped to left camera coordinates ( $\mathbf{X}^{LEye}$ ) using a transformation matrix  $\mathbf{T}_{Root}^{LEye}$ . Then we use a pinhole camera model to transform the 3D point ( $\mathbf{X}^{LEye}$ ) into image coordinates ( $\mathbf{X}^{img}$ ):

$$\begin{pmatrix} X_x^{img} \\ X_y^{img} \end{pmatrix} = \begin{pmatrix} f_x X_x^{LEye} / X_z^{LEye} \\ f_y X_y^{LEye} / X_z^{LEye} \end{pmatrix}, \quad (3)$$

where  $f_x, f_y$  are focal lengths of the camera. Radial distortion of cameras was not considered.

This approach doesn't require information from both eyes and enables us to estimate only one side of the robot body (e.g. parameters of the left arm and left eye). For example, the estimated parameter vector  $\phi$  in the case of the kinematic chain connecting left arm and left eye consists of the following parameters:  $\phi = \{\phi^l, \phi^{le}\}$ , where  $\phi^l$  and  $\phi^{le}$  are parameters corresponding to the robot left arm and to the left eye, respectively. The objective function is then defined as:

$$\phi^* = \underset{\phi}{\operatorname{argmin}} \sum_{m=1}^M \|\mathbf{X}_m^{l,img}(\phi^l, \phi^{le}) - \mathbf{u}_m^L\|^2, \quad (4)$$

where  $\mathbf{X}_m^{l,img}$  is the  $m^{\text{th}}$  2D position of the estimated left arm end-effector projected to left eye image coordinates and  $\mathbf{u}_m^L$  is the  $m^{\text{th}}$  2D position of the observed left arm end-effector in the left camera. For two arms and two eyes we get four possible combined chains: left/right arm to right/left eye. Since the results are similar due to symmetry, we present in the experimental section results only for the Left arm - Left eye (LA-LEye) chain.

3) *Combining multiple chains (LA-RA-LEye, LA-RA-LEye-REye)*: In order to estimate all kinematic parameters of the robot, we can take advantage of combining some or all of the above mentioned kinematic chains. For example, in the case that we combine LA-RA, LA-LEye and LA-REye chains together into LA-RA-LReye, the estimated parameter vector  $\phi$  consists of the following parameters:  $\phi = \{\phi^r, \phi^l, \phi^{re}, \phi^{le}\}$ , where  $\phi^l, \phi^r, \phi^{re}$ , and  $\phi^{le}$  are parameters corresponding to the left arm, right arm, right eye, and left eye, respectively. The objective function is in this case defined as:

$$\begin{aligned} \phi^* = \underset{\phi}{\operatorname{argmin}} \sum_{m=1}^M \{ & \mu \cdot \|\mathbf{X}_m^{r,R}(\phi^r, \Theta_m^r) - \mathbf{X}_m^{l,R}(\phi^l, \Theta_m^l)\| + \\ & \|\mathbf{X}_m^{l,I}(\phi^l, \phi^{le}) - \mathbf{u}_m^{l,I}\| + \|\mathbf{X}_m^{r,I}(\phi^r, \phi^{le}) - \mathbf{u}_m^{r,I}\| + \\ & \|\mathbf{X}_m^{l,I}(\phi^l, \phi^{re}) - \mathbf{u}_m^{l,I}\| + \|\mathbf{X}_m^{r,I}(\phi^r, \phi^{re}) - \mathbf{u}_m^{r,I}\| \}^2, \end{aligned} \quad (5)$$

where  $M$  is the number of poses (configurations) used for calibration,  $\mathbf{X}_m^{r,R}$  and  $\mathbf{X}_m^{l,R}$  are the  $m^{\text{th}}$  estimated end-effector positions in the Root frame for the right and left arm, respectively. These are computed using a given parameter estimate  $\phi$  and joint angles from joint encoders  $\Theta_m$ . Values  $\mathbf{X}_m^{l,I}$  and  $\mathbf{X}_m^{r,I}$  are the  $m^{\text{th}}$  positions of the estimated left arm end-effector projected to left eye and right eye image coordinates,

respectively, and  $\mathbf{u}_m^L$  and  $\mathbf{u}_m^{r,I}$  are the  $m^{\text{th}}$  2D position (pixel coordinates) of the left arm end-effector observed in the left and right eye/camera, respectively (variables  $\mathbf{X}_m^{l,I}, \mathbf{X}_m^{r,I}, \mathbf{u}_m^L$  and  $\mathbf{u}_m^{r,I}$  correspond to the right arm). Since the cost function contains both 3D and reprojection errors, the distances in space were multiplied by a coefficient  $\mu$  determined from the intrinsic parameters of cameras and distance  $d$  of the end-effector from the eye:  $\mu = 320px / (d * (\pi/3))$ .

#### D. Non-linear least squares optimization

The objective functions (Eqs. [1]- [5]) defined for the optimization problem described in Section III-B are of the least-squares form and therefore can be minimized by Levenberg-Marquardt algorithm for non-linear least squares optimization (we used MATLAB implementation of the algorithm, same as in [17]). This iterative local algorithm performs minimization of a non-linear objective function by linearizing it at the current estimate every iteration. It interpolates between the Gauss-Newton and gradient descent method, combining advantages of both.

#### E. Error metrics

For comparing the results achieved for individual settings, we make use of the following error metrics:

1) *Cartesian error between poses (position)*: Cartesian position error  $E_c$  between two generic poses, A and B, where  $\mathbf{P}_A = [x_A, y_A, z_A]$  and  $\mathbf{P}_B = [x_B, y_B, z_B]$  are 3D Cartesian positions of the end-effector, is defined as:

$$E_c = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}. \quad (6)$$

We evaluate the Cartesian error over the set of  $N$  testing poses, which are selected as described in the section IV-B.

2) *Quality of estimated parameters*: For each estimated parameter  $\phi_i$  we compute the mean difference ( $e_i$ ) of the estimated parameter  $\phi_i^e$  from the target parameter value  $\phi_i^t$  (averaged over  $R$  repetitions of the experiment):

$$e_i = \frac{\sum_{r=1}^R |\phi_{i,r}^e - \phi_i^t|}{R}, \quad (7)$$

as well as standard deviation of the parameter.

## IV. DATA ACQUISITION AND DESCRIPTION

### A. Pose set generation

With the goal of comparing different calibration methods on a humanoid robot, we chose a dataset where the two arms of the robot are in contact—thereby physically closing the kinematic chain through self-touch. At the same time, the robot gazes at the contact point (self-observation). The points were chosen from a cubic volume in front of the robot. For each target, using the Cartesian solver and controller [20], the iCub moves the left hand with end-effector in the palm to the specified point. Then it moves the right hand, with end-effector in the tip of the index finger, to the same point, with the additional constraint that the finger can be at most  $50^\circ$  away from the perpendicular direction of the palm. 5055 points and corresponding joint configurations were thus generated,

with a difference on left and right effector position in every configuration of maximum  $0.01\text{mm}$ —see Fig. 2, right. The gaze controller [21] was used to command the neck and eyes of the robot to gaze at the same target (code and video can be accessed at [22]). The full dataset thus consists of 5055 data vectors  $\mathbf{X}_i = [\mathbf{X}_i^{\text{target}}, \mathbf{X}_i^{\text{RA}}, \mathbf{X}_i^{\text{LA}}, \Theta_i]$  composed of target point coordinates ( $\mathbf{X}_i^{\text{target}} \in \mathbb{R}^3$ ), corresponding right arm and left arm end-effector positions ( $\mathbf{X}_i^{\text{RA}} \in \mathbb{R}^3$ ,  $\mathbf{X}_i^{\text{LA}} \in \mathbb{R}^3$ ), and joint angles  $\Theta_i$  for every joint of the torso, arms, neck, and eyes ( $\Theta_i \in \mathbb{R}^{20}$ ). Note that the solvers work with a given tolerance and hence  $\mathbf{X}_i^{\text{target}} \neq \mathbf{X}_i^{\text{RA}} \neq \mathbf{X}_i^{\text{LA}}$ .

This way of dataset generation draws on previous work [2] and is hence feasible on the real robot provided sufficient quality of the initial model. Li et al. [14] provide an alternative control method: “tactile servoing”. The robot could be also manipulated into the desired configurations while in gravity compensation mode.

### B. Training and testing dataset

We had 5055 configurations with  $|\mathbf{X}_i^{\text{RA}} - \mathbf{X}_i^{\text{LA}}| < 0.01\text{mm}$ . The  $0.01\text{mm}$  error will at the same time constitute the lower bound on the maximum achievable calibration accuracy using the closure of the kinematic chain through self-touch. For the case of loop closure through the cameras, we employ the neck and eye joint values obtained from the solver in the simulator but reproject the end-effector positions directly and accurately into the cameras simulated in Matlab. The 5055 data points were further divided into training and testing datasets in the following way:  $N$  out of 4755 poses are used as a training set on which the optimization process is performed (with a subset of 10, 20, 50, or 1000 poses chosen at random in different experiments) and 300 poses are used for testing purposes. Fig. 2, left, shows the distribution of joint values for individual joints in the dataset—this may impact the identifiability of individual parameters.

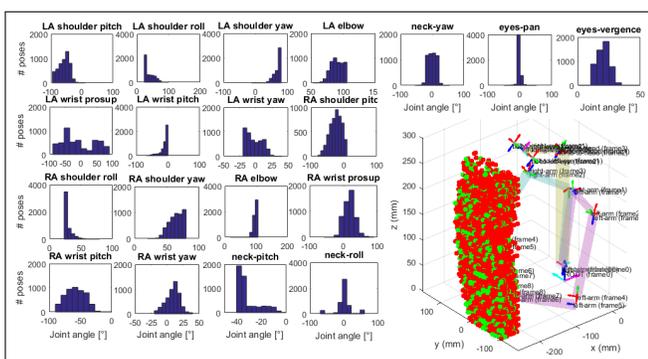


Fig. 2. Dataset visualization – 5055 configurations. (left) Distribution of joint values. (right) End-effector positions. Red – left arm; Green – right arm.

### C. Measurement error

Measurement noise with a Gaussian distribution was added motivated by the sensory accuracy in the real robot. Since distance between individual taxels on the real iCub sensitive skin is around  $5\text{mm}$ , we decided to use Gaussian noise with zero mean and  $\sigma^2 = 5$  for touch as a baseline. For cameras,

we introduce a  $5\text{px}$  error (Gaussian noise with zero mean and  $\sigma^2 = 5\text{px}$ ), inspired by the setup in [3] where the iCub is detecting its fingertip in the camera frame. These errors are used in all experiments in the Simulation results section if not stated otherwise. In Section V-C we evaluate how changing the size of these measurement errors affects the resulting accuracy of end-effector position detection for individual chains.

### D. Perturbation of the initial parameters estimate

To evaluate the dependence of the optimization performance on the quality of the initial estimates of the parameters, we perturbed all estimated parameters by a *perturbation factor*  $p = \{2, 5, 10, 20\}$ . We perturbed all initial offset values  $o_i$  as follows:

$$o_i^{\text{new}} = 1/100 * p * \text{uniform}[-1; 1] + o_i [\text{rad}], \quad (8)$$

It is reasonable to expect that the remaining DH parameters ( $\alpha$ ,  $a$ , and  $d$ ) will be in general more accurate as they can be extracted from CAD models and there is no moving part and no encoder involved. Therefore, their perturbation was chosen as follows:

$$\alpha : \alpha_i^{\text{new}} = 1/1000 * p * \text{uniform}[-1; 1] + \alpha_i [\text{rad}], \quad (9)$$

$$a, d : \Phi_i^{\text{new}} = 0.1 * p * \text{uniform}[-1; 1] + \Phi_i [\text{mm}].$$

## V. SIMULATION RESULTS

In this section we show the calibration results. We evaluated our approach using both error of the end-effector position—the cost function optimized (or distance in camera frame for projections into eyes)—as well as error in individual parameters (vs. their correct values). We compared kinematic chains used for calibration, number of free parameters which were estimated by the optimization process, different perturbation factor on individual parameters, number of training poses (data points), as well as measurement noise levels. Performance is always evaluated on the testing dataset.

### A. Results for different chain combinations and number of training poses

Fig. 3 (top) shows the performance in terms of end-effector position estimation when DH parameters of the left arm (LA) chain are calibrated, utilizing different kinematic chain combinations: The “self-observation” from a single camera (LALEye) and “self-touch” only (LARA) are outperformed by “stereo self-observation” (LALREye) and all the chains together provide the best results (LARALREye). Clearly, more training poses (50 vs. 20) improve calibration results; 1000 poses should be sufficient to reach an optimal value and serve as a lower bound on the error. The effect of initial parameter perturbation factor is also shown; for all perturbation levels, the performance is stable (low error variance).

In Fig. 3 (bottom) only the largest “multi-chain” LARALREye is employed for training but the chains whose parameters are subject to calibration are varied. The error of end-effector position estimation is increasing with higher number of parameters estimated; however, even if parameters of all chains (86 DH parameters) are perturbed and subject to calibration

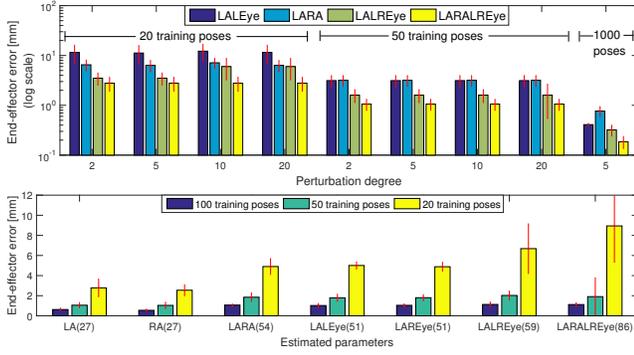


Fig. 3. End-effector position error after optimization—averaged over 10 repetitions. (Top) Left Arm chain calibration (full DH) using different chain combinations, different initial perturbation factors (2, 5, 10, 20) and training on 20 (left), 50 (middle), and 1000 poses (right – pert. factor 5 only). (Bottom) Performance of different parameter sets subject to calibration – LARALREye chain was used for calibration of parameters. Free parameters (being calibrated) in a given chain are denoted. E.g., LALEye denotes that all 51 DH parameters of left arm and left eye (including head) are calibrated, and the rest of the DH parameters (e.g. right arm) is considered to be known.

simultaneously, end-effector error of around 2 (1)  $mm$  can be achieved with 50 (100) poses.

To investigate the distribution of errors for individual chains, we examined error residuals for every testing pose. For a higher number of training poses, error residuals have a zero mean and Gaussian distribution. For lower number of poses (especially for higher perturbation), the residuals are bigger and skewed and the resulting calibration also strongly depends on initialization. In Fig. 4, the end-effector error residuals for perturbation factor  $p = 10$  are shown for their  $x$  and  $z$  coordinates (other 2D projections were qualitatively similar)—for different chains and different number of training poses.

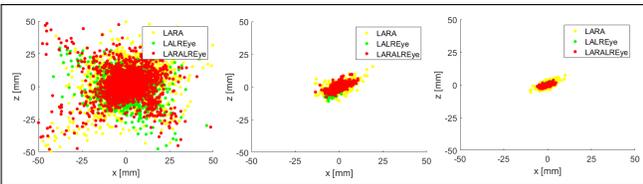


Fig. 4. Error residuals – Left Arm (LA) chain calibration using LARA, LALREye and LARALREye chains. Results visualized on 300 testing poses for each of 10 repetitions of the optimization, with random parameter initialization (3000 points in total per chain shown). (Left) 10 training poses; (Middle) 20 training poses; (Right) 50 training poses. Perturbation factor 10 and measurement errors 5 mm for skin and 5 px for cameras were considered.

### B. Observability analysis of individual chains

We conducted an observability analysis using Singular Value Decomposition (SVD) of the identification Jacobian matrix  $J = [J_1, \dots, J_n]$ , where  $n$  is the number of configurations in the training pose set and  $J_n(i, j) = \left[ \frac{\partial(X_i^r - X_i^e)}{\partial \phi_j} \right]$ ,  $\phi_j$  is the parameter  $j$  to be estimated,  $(X_i^r - X_i^e)$  denotes the error between the real/observed ( $X^r$ ) and estimated ( $X^e$ ) value of the  $i^{\text{th}}$  coordinate in the given chain.<sup>1</sup> The Jacobian matrix

<sup>1</sup>E.g., for LALEye,  $X$  corresponds to 2 errors: error on the coordinate  $u$  and  $v$  as a reprojection of the end-effector position into the cameras; for LARA chain,  $X$  will correspond to 3 numbers: distance in  $x$ ,  $y$  and  $z$  coordinate between right ( $X^{r,R}$ ) and left arm ( $X^{l,R}$ ) end-effector 3D positions.

represents the sensitivity of end-effector positions or their camera reprojections to the change of individual parameters. Using SVD, we can obtain a vector of singular numbers  $\sigma_i$ . Comparison of the obtained singular numbers for individual chains for the task of estimating all DH parameters of the left arm (using same training pose set) can be seen in Fig. 5. We also evaluated observability indices  $O_1$  [23] and  $O_4$  [24] (performance of observability indices for industrial robot calibration was evaluated by Joubair [25]).  $O_1$  index is defined as:  $O_1 = \frac{(\sigma_1 \sigma_2 \dots \sigma_m)^{1/m}}{\sqrt{n}}$ , where  $m$  is the number of independent parameters to be identified,  $\sigma_i$  is the  $i^{\text{th}}$  singular number, and  $n$  is the number of calibration configurations. Index  $O_4$  is defined as:  $\frac{\sigma_m^2}{\sigma_1}$ . See Fig. 5 (bottom panels). The chain LALEye for 10 poses has very low observability caused by not full rank Jacobian (we have 24 parameters to estimate but only 20 equations). The highest observability is achieved in all cases for the largest chain LARALREye, where the information from touch and both cameras was used.

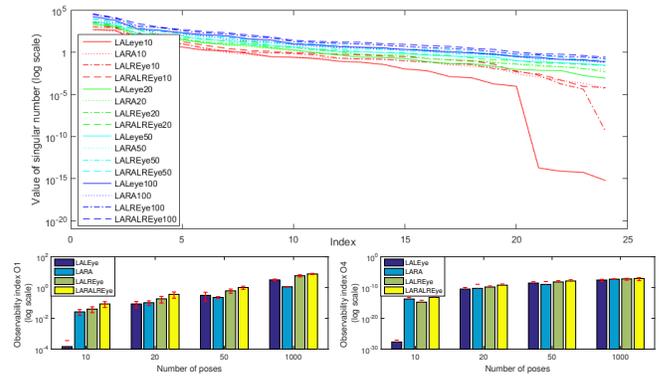


Fig. 5. Observability – Left Arm (LA) chain calibration (full DH) using different chain combinations. (Top) singular numbers of identification Jacobian for different chains used for calibration. Evaluation is performed over the same pose set for every chain. Red, green, turquoise, and blue color of the lines denote 10, 20, 50, and 1000 poses in the training set respectively. (Bottom left) Observability index  $O_1$  [23]. (Bottom right) Observability index  $O_4$  [24].

### C. Evaluation of error based on measurement noise

We evaluated the effect of measurement noise in individual sensors (touch, cameras) on the accuracy of end-effector position error on the testing data set—see Fig. 6. With same error in pixels on cameras and in  $mm$  on “touch sensors” (first two columns –  $2px/2mm$ ,  $5px/5mm$ ), LALREye chain (both eyes, no touch) and LARALREye (both eyes and touch) have smallest final end-effector errors, for the “multi-chain” even smaller. When error on cameras increases ( $5E2T$ ,  $10E2T$ ,  $10E5T$ ,  $10E10T$ ), the camera chains (LALREye, LARALREye) are affected whereas the performance of the chain with touch (LARALREye) is not degraded. Conversely, more error on “touch” ( $2E5T$ ,  $2E10T$ ,  $5E10T$ ) impacts the “touch only” chain (LARA), but the LARALREye remains robust.

### D. Quality of DH parameter estimates

To get further insight and take advantage of the simulation study where we have access to ground truth values of all

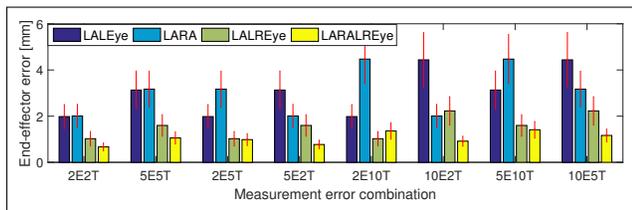


Fig. 6. End-effector position accuracy for different combinations of measurement noise on cameras and “touch sensor”. Different chains employed to estimate DH parameters of the left arm (50 training poses, error evaluated over 300 testing poses, averaged over 10 repetitions). X-axis labels read as follows: first number – error on cameras (“Eyes”) in pixels; second number – error on the touch sensor in *mm* (i.e. 5E2T denotes that we introduced zero-mean Gaussian error with 5px and 2mm variance to cameras and touch respectively).

parameters, we also studied whether the optimization based on end-effector error also leads to correct estimates of all DH parameters—focusing on the left arm (LA) chain.

Fig. 7 shows the results for all estimated parameters when the LA-RA (“self-touch”) chain was used for calibration, using different number of training poses. The errors on the length parameters (top panel) are on average distributed between approx. 1 and 10 *mm*. For the angular quantities, it is in the 0.1 to 1° range for the proximal joints.

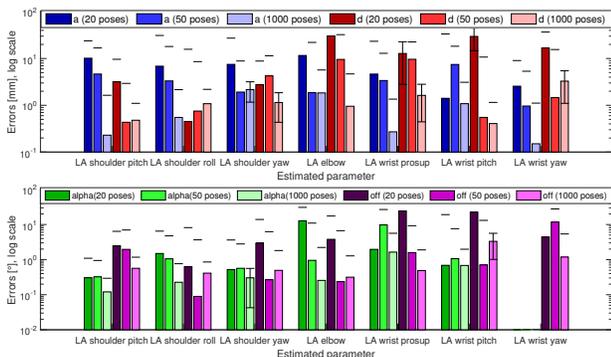


Fig. 7. Quality of DH parameter estimation for LA chain using LA-RA chain. Errors on individual parameters after optimization for different number of poses: (Top)  $a$  and  $d$  parameters; (Bottom)  $\alpha$  and *offsets*. Averaged over 10 repetitions, perturbation factor 5, measurement noise 5px on cameras and 5mm on touch.

Finally, having showed above that the “self-touch and self-observation” (LARALREye) chain slightly outperforms the “stereo self-observation” only chain (LALREye) (Fig. 3 top, Fig. 6), also in observability (Fig. 5), here in Fig. 8 we can observe a similar trend in the estimated parameters of the LA chain against their ground truth values. The parameter estimates obtained from LARALREye are significantly better for  $d$  for all joints except for wristPr and elbow and for  $a$  for all shoulder joints. The other parameters estimates are comparable. The wrist joint calibration seems to be sensitive on the selection of training poses and will need further study.

## VI. DISCUSSION AND CONCLUSION

We quantitatively and systematically investigated the potential of automatic self-contained kinematic calibration (DH

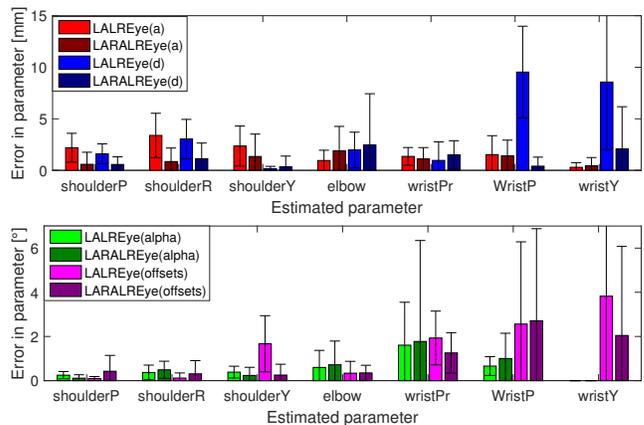


Fig. 8. Absolute error of estimated DH parameters of LA chain after optimization (50 training poses, perturbation factor 5, measurement noise 5 px on cameras and 5 mm on touch). (Top)  $a$  and  $d$  parameters. (Bottom)  $\alpha$  and *offsets*.

parameters including camera extrinsic parameters) of a humanoid robot employing different kinematic chains—in particular relying on self-observation and self-touch. The parameters varied were: (i) type and number of intersecting kinematic chains used for calibration, (ii) parameters and chains subject to optimization, (iii) amount of initial perturbation of kinematic parameters, (iv) number of poses/configurations used for optimization, (v) amount of measurement noise in end-effector positions / cameras. We also tracked the computation time and while the details differ depending on the settings (chain calibrated, number of poses, etc.), a typical optimization run would not take more than tens of seconds on an older laptop PC. Next to results w.r.t. the cost function itself (error on end-effector or camera reprojection) a number of additional analyses were performed including error residuals, errors on estimated parameters compared to ground truth, and observability analysis.

While some results were expected (such as improvement when more configurations are added or poor performance when using self-observation from a single camera), the most notable findings are: (1) calibrating parameters of a single chain (e.g. one arm) by employing multiple kinematic chains (“self-observation” and “self-touch”) is superior in terms of optimization results (Fig. 3 top) as well as observability (Fig. 5); (2) when using multi-chain calibration, fewer poses suffice to get similar performance compared to when e.g. only observation from a single camera is used (Fig. 3 top); (3) parameters of all chains (here 86 DH parameters) can be subject to calibration simultaneously and with 50 (100) poses, end-effector error of around 2 (1) mm can be achieved (Fig. 3 bottom); (4) adding noise to a sensory modality degrades performance of all calibrations employing the chains relying on this information (Fig. 6). The last point is interesting to discuss in relation to Birbach *et al.* [17] who put forth the hypothesis that calibrating multiple chains simultaneously is superior to pairwise sequential calibration. Our results support this provided that measurement noise is small. Instead, if a certain modality is noisy, it may be beneficial to preferentially employ chains that rely on more accurate measurements first

and then calibrate a “noisy chain” in a second step.

We have only reported results from simulation, however, we claim that this was the right tool for this type of investigation. At the same time, our setup and choice of parameters was drawing on experiments performed in the real robot—self-touch [2] and self-observation [3], [4] in particular—which makes the results grounded in a real setting and should inform future experimentation on the iCub. The method to combine chains and analyze the results presented here can be transferred to other platforms as well.

There are several aspects that we want to further investigate in the future. First, we note that while we did control for the angle between the palm and the contralateral finger for self-touch in the dataset generation, we did not monitor whether the contact point would be also visible. Additional analyses revealed that the contact point would not be occluded and hence be visible by both cameras in 35% of the poses and by one of the cameras in 53%. We recomputed the observability with this subset of the dataset only and found no decrease. In the future, configurations with occlusions should be excluded from dataset generation. Second, we found that around 50 configurations (data points) suffice for reasonable calibration. Finding the optimal subset of not more than 10 configurations would be desirable, such that recalibration can be performed rapidly. Here, clever pose selection will be necessary to warrant adequate and stable performance. Third, the information from the two cameras can be used to reproject observed position of the end-effector in image coordinates of both eyes (pixel  $(u, v)$ ) to 3D space ( $X^{eye}$ ) (similar to [3], [26])—leading onto yet another formulation of the optimization problem. Fourth, our investigation can be extended considering also the contribution of inertial sensors—in the robot head [17] or distributed on the robot body [10], [8]. Fifth, the present method can be compared with filtering approaches [4], [16] or with methods that pose fewer assumptions on the initial model available (e.g., [27]). Finally, the self-touch scenario can be also turned around from using a tactile array to calibrate kinematics [2], [14] to calibrating the skin itself [28].

#### ACKNOWLEDGMENT

We thank Alessandro Roncone for assistance with the models of the iCub robot in MATLAB and the source files leading to Fig. 1 left and Ugo Pattacini for discussions, tips, and assistance with the use of Cartesian solvers leading to the generation of self-touch configurations.

#### REFERENCES

- [1] J. Hollerbach, W. Khalil, and M. Gautier, “Model identification,” in *Springer Handbook of Robotics*, 2nd ed., B. Siciliano and O. Khatib, Eds. Springer, 2016, pp. 113–138.
- [2] 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 on*, 2014, pp. 2305–2312.
- [3] S. R. Fanello, U. Pattacini, I. Gori, V. Tikhonoff, M. Randazzo, A. Roncone, F. Odone, and G. Metta, “3D stereo estimation and fully automated learning of eye-hand coordination in humanoid robots,” in *2014 IEEE-RAS Int. Conf. on Humanoid Robots - HUMANOIDS '14*, 2014.
- [4] P. Vicente, L. Jamone, and A. Bernardino, “Online body schema adaptation based on internal mental simulation and multisensory feedback,” *Frontiers in Robotics and AI*, vol. 3, p. 7, 2016.
- [5] 2019 (Project webpage). [Online]. Available: <https://sites.google.com/site/karlastepanova/robot-self-calibration-icub>
- [6] M. Hersch, E. Sauser, and A. Billard, “Online learning of the body schema,” *International Journal of Humanoid Robotics*, vol. 5, pp. 161–181, 2008.
- [7] R. Martinez-Cantin, M. Lopes, and L. Montesano, “Body schema acquisition through active learning,” in *Proc. Int. Conf. on Robotics and Automation (ICRA)*, 2010.
- [8] P. Mittendorfer and G. Cheng, “Open-loop self-calibration of articulated robots with artificial skins,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4539–4545.
- [9] E. Dean-Leon, K. Ramirez-Amaro, F. Bergner, I. Dianov, and G. Cheng, “Integration of robotic technologies for rapidly deployable robots,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1691–1700, 2018.
- [10] N. Guedelha, N. Kuppaswamy, S. Traversaro, and F. Nori, “Self-calibration of joint offsets for humanoid robots using accelerometer measurements,” in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 1233–1238.
- [11] K. Yamane, “Practical kinematic and dynamic calibration methods for force-controlled humanoid robots,” in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 269–275.
- [12] C. Bartolozzi, L. Natale, F. Nori, and G. Metta, “Robots with a sense of touch,” *Nature materials*, vol. 15, no. 9, pp. 921–925, 2016.
- [13] R. S. Dahiya and M. Valle, *Robotic Tactile Sensing*. Springer, 2013.
- [14] 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*. IEEE, 2015, pp. 1109–1114.
- [15] R. Khusainov, A. Klimchik, and E. Magid, “Humanoid robot kinematic calibration using industrial manipulator,” in *Mechanical, System and Control Engineering (ICMSC), 2017 International Conference on*. IEEE, 2017, pp. 184–189.
- [16] R. Zenha, P. Vicente, L. Jamone, and A. Bernardino, “Incremental adaptation of a robot body schema based on touch events,” in *Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2018.
- [17] O. Birbach, U. Frese, and B. Bäuml, “Rapid calibration of a multi-sensorial humanoids upper body: An automatic and self-contained approach,” *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 420–436, 2015.
- [18] 2017. [Online]. Available: <http://wiki.icub.org/wiki/ICubForwardKinematics>
- [19] K. Nickels, “Hand-Eye calibration for Robonaut,” NASA Summer Faculty Fellowship Program Final Report, Tech. Rep., 2003.
- [20] U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini, “An experimental evaluation of a novel minimum-jerk Cartesian controller for humanoid robots,” in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems (IROS)*, 2010.
- [21] A. Roncone, U. Pattacini, G. Metta, and L. Natale, “A Cartesian 6-DoF gaze controller for humanoid robots,” in *Robotics: Science and Systems*, vol. 2016, 2016.
- [22] 2017. [Online]. Available: <https://github.com/matejhof/icub-selftouch-with-gaze-generator>
- [23] J.-H. Borm and C.-H. Menq, “Experimental study of observability of parameter errors in robot calibration,” in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*. IEEE, 1989, pp. 587–592.
- [24] A. Nahvi and J. M. Hollerbach, “The noise amplification index for optimal pose selection in robot calibration,” in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 647–654.
- [25] A. Joubair, A. Tahan, and I. A. Bonev, “Performances of observability indices for industrial robot calibration,” in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2477–2484.
- [26] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2008.
- [27] P. Lanillos and G. Cheng, “Adaptive robot body learning and estimation through predictive coding,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 2018.
- [28] A. Albin, S. Denei, and G. Cannata, “Towards autonomous robotic skin spatial calibration: A framework based on vision and self-touch,” in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 153–159.