Automatic kinematic chain calibration using artificial skin: self-touch in the iCub humanoid robot

Alessandro Roncone, Matej Hoffmann, Ugo Pattacini, and Giorgio Metta

***Abstract*— Calibration continues to receive significant attention in robotics because of its key impact on performance and cost associated with the operation of complex robots. Calibration of kinematic parameters is typically the first mandatory step. To this end, a variety of metrology systems and corresponding algorithms have been described in the literature relying on measurements of the pose of the end-effector using a camera or laser tracking system, or, exploiting constraints arising from contacts of the end-effector with the environment. In this work, we take inspiration from the behavior of infants and certain animals, who are believed to use self-stimulation or self-touch to "calibrate" their body representations, and present a new solution to this problem by letting the robot close the kinematic chain by touching its own body. The robot considered in this paper is sensorized with tactile arrays for a total of about 4200 sensing points. The correspondence between the predicted contact point from existing forward kinematics and the actual position on the robot's 'skin' provides sample data that allows refining the kinematic representation (DH parameters). The data collection procedure is automated—self-touch is autonomously executed by the robot—and can be repeated at any time, providing a compact self-calibration system that does not require an external measurement apparatus.**

I. INTRODUCTION

Practically all robots performing manipulation tasks rely on models of their kinematics and dynamics. Their success is largely determined by the accuracy of such models. This is even more so if they operate with limited feedback, as it is often the case when we consider humanoid robots in real-time interaction with the environment. The models are typically based on mechanical design specifications (such as CAD drawings) of the robot. However, inaccuracies creep in in many ways as for example in the assembly process, in mechanical elasticity, or simply because of cheap design or components. Therefore, the actual model parameters of every robot exemplar have to be found by means of a calibration procedure.

In this work, we will be concerned with calibration of the standard Denavit-Hartenberg (DH) parameters that fully describe the robot's kinematics through a series of rotations and translations from the base of the robot up to the end-effector. If the configuration of every joint is known, the full pose (3D position and 3D orientation) of the end-effector can be obtained in the base reference frame. However, in order to calibrate the robot's parameters additional information is required. This can be obtained by observing the end-effector configuration (or several of its components) w.r.t. the base. The literature provides various examples of apparatuses that can measure one or more of the components of the end-effector pose employing mechanical, visual, or laser systems. An overview of these—so-called *open-loop calibration methods*—is presented in [1], [2]. Alternatively, physical constraints on the end-effector position or orientation can substitute for measurements (cf. [1], [2] again for an overview). As the robot is in contact for example with the ground, these methods are called *closed-loop*. In fact, the problem can be framed in such a way that the open- and closed-loop methods are mathematically equivalent—the external measurement systems can be modeled as additional joints and links that close a virtual loop; in both cases the excess of sensed over actual degrees of freedom is needed, as expressed by the calibration index [1], which at the same time defines the number of equations per pose that are available for calibration. Recently, apparatuses extending the kinematic chain using a laser pointer have become popular (e.g., [3], [4]). Different arrangements have different calibration indices, accuracy, requirements on the environment, and cost. Nevertheless, all of them invariably require to know beforehand a number of quantities from the robot's environment (such as a measurement system with a known pose w.r.t. the robot base, a fixed contact point in the environment where the robot can be attached, a surface that is known to be planar on which the robot can slide, etc.). These conditions have to be present for recalibration to be performed.

This has motivated alternative solutions to the self-calibration problem that are also "self-contained" and can be performed autonomously by the robot. One option is self-observation using a stereo camera mounted on the robot. This has been shown in a humanoid robot in [5] and in [6] in a humanoid torso setup. The limit of these approaches is usually to be found in the accuracy of the pose observation from visual input. Special markers need to be used and attached to known positions on the robot, such as on the end-effector. Alternatively, inertial sensors can be used. Xsens for example developed a wearable setup for humans composed of several inertial measurement units (IMUs) [7], which, however, requires a specific *a priori* body model. Mittendorfer and Cheng [8] presented a method that uses data from accelerometers distributed on the surface of a robot (this matches the artificial skin they developed [9]) to calibrate the DH parameters. Other approaches that do not rely on an

This work was supported by the project Xperience (FP7-ICT-270273) and EFAA (FP7-ICT-270490). M.H. was also supported by the Swiss National Science Foundation Fellowship PBZHP2-147259.

A. Roncone, M. Hoffmann, U. Pattacini, and G. Metta are with iCub Facility, Istituto Italiano di Tecnologia, Via Morego 30, 16123 Genova, Italy. {alessandro.roncone, matej.hoffmann, ugo.pattacini, giorgio.metta}@iit.it

explicit given representation—like the DH parameters—, but that learn more implicit relationships between proprioceptive and visual variables, for example, have been also developed (see [10] for a review).

In this article, we present an approach that closes the kinematic loop in a completely new way: i.e. by self-touch. Our inspiration is in biology: infants do not have access to calibration chambers or ground truth measurements. In addition, in early infancy, the visual system is still immature, thus an unlikely source of accurate calibration information. A possible "self-calibration" strategy entails therefore self-stimulation: touching their own body gives rise to unique stimulation patterns—tactile stimulation on the touching and the touched part, together with corresponding proprioceptive feedback [11], [12]. From a robotics point of view, this constitutes a compact calibration procedure that can be repeated at any given time and that does not incur any additional cost. Furthermore, not only the kinematic model, but also the geometry of the robot (its volume in space) can be learned. The new requirements induced by this approach are: (i) the availability of tactile arrays on the robot's surface; (ii) a sufficient agility to permit self-touch (or self-collision) configurations; (iii) the availability of a controller—such as hybrid position-force control—that allows safe execution of the movements which generate self collisions. These three conditions are satisfied by the iCub humanoid robot (see [13] for the platform in general, [14] for the whole-body skin, and [15] for the mechanical design)—the experimental platform employed in this work. The method proposed is however applicable to a much wider set of platforms. First and foremost, solutions to robot tactile sensing are now numerous (see [16] for a survey)—many of them with the ambition of providing a portable solution that can be simply attached to robots of any shape. Second, any robot that possesses multiple limbs—especially humanoid robots—will be able to self-touch (apart from pathological cases). Third, many robots have force/torque sensing and corresponding controllers that can actively control impedance, thus allowing for safe interactions with the environment and with themselves (e.g., [17]).

We will present a method for calibration using self-touch and describe its experimental validation. The theoretical contribution consists in an advantageous transformation of the problem of controlling two serial chains (like two arms of a humanoid robot) to self-collide at a certain point in space into a single floating-base serial chain that originates at the contact point and ends in the "end-effector" of the other arm—the point that is "touching". Interestingly, this method encompasses both open- and closed-loop calibration. If different points on the robot's skin are touched, an initial calibration of the skin serves as a "metrology" to observe the 3D position of the end-effector. This can be used for calibrating the DH parameters only, or, simultaneously, to improve the calibration of the tactile array. Alternatively, keeping the same contact point and varying the position of the joints constitutes a closed-loop calibration setup, where the end-effector is constrained in a known position. The advantage of our approach is that a large number of such points are automatically available corresponding to the skin sensors' positions. The experimental validation in this work uses the first method: contact points are varied and corresponding joint configurations are recorded. Optimization of the DH parameters is then performed starting from different initial configurations.

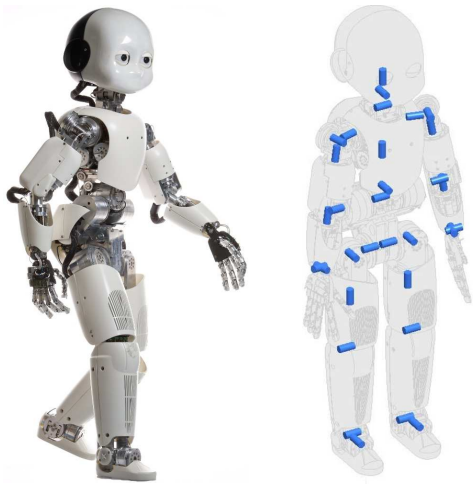


Fig. 1: **The iCub robot (left) and its kinematic structure (right).** (from [15])

This article is structured as follows. In Section II, the robot and the scenario are presented. A solution to the self-touch problem is developed, which is part of our contributions. The experimental protocol and the optimization problem are defined. The optimization experiments are presented in Section III, followed by Conclusions and Future work.

II. METHODS AND EXPERIMENTAL SETUP

In this section, we present the experimental platform and the components most relevant for this work: that is, the iCub overall kinematics and the artificial skin. Then we introduce the implementation of movements that generate controlled self-collisions which in turn involves a reformulation of the kinematic representation of the task, an inverse kinematics solver and a corresponding controller. Finally, we present the data collection and optimization procedures for the update of the DH parameters.

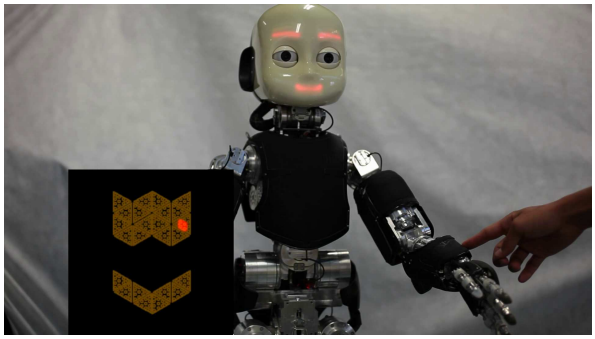
A. The iCub humanoid robot

The iCub (Fig. 1, left) is an open-source platform for research in cognitive robotics [13]. Its mechanical design is detailed in [15].

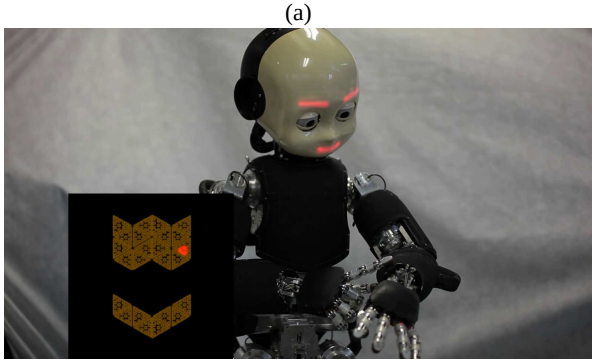
1) Kinematic representation: In this work, we will be concerned with the upper body of the robot only. Its kinematic structure can be seen in Fig. 1, right. It consists of revolute (1 DoF) joints only.

2) Artificial skin: The iCub has recently been equipped with an artificial pressure-sensitive skin covering most body parts [14]. The latest iCub version contains approximately 4200 tactile elements (taxels) [15]—in the fingers, palms, forearms and upper arms, torso, and lately also in the legs

and feet. In the experiments performed in this work, we restrict ourselves to the forearm, which is covered by two skin patches of 84 and 192 taxels. Their shape and activation when the robot is touched can be seen in Fig. 2. A calibration of the skin has been performed in [18]—the pose of each taxel in the reference frame of the corresponding link is available.



(a)



(b)

Fig. 2: **Self-touch experiment**. (a) The iCub touched on its left forearm by the experimenter; corresponding skin activation shown on the left. (b) The iCub touching the previously stimulated point using the index finger of the contralateral arm.

B. Self-touch implementation

From a robotic perspective, self-touch is the closure of an open kinematic chain. Clearly, robotic manipulators are designed with a different goal in mind—reaching in the operational space—and self-collisions are typically not desirable in the first place. In what follows, we will be specifically concerned with the case of self-collision, or self-touch, of two manipulators whose operational spaces are overlapping—like two arms of a humanoid robot. In order for the two end-effectors to collide, standard techniques could be used: a dual inverse kinematics task for each of the arms with the target set to any point of the shared operational space. However, a very accurate model is necessary to achieve contact. In addition, we are interested in calibrating the whole 3D geometry of the robot, not only the mapping from the base to the end-effector. Hence, other parts of the body (in this paper the forearm) need to be touched by the contralateral arm.

1) Difficulties of classical approach to self-collision of two robotic manipulators: A schematic illustration of this situation is depicted in Fig. 3a: without loss of generality, we can assign the point to be touched (blue cross) to the left forearm; thus, the goal is to reach this point with the end-effector $\mathbf{EE_1}$ of the contralateral arm (the DoF in the schematics do not match those of the real robot). In this case, we are facing the following difficulties:

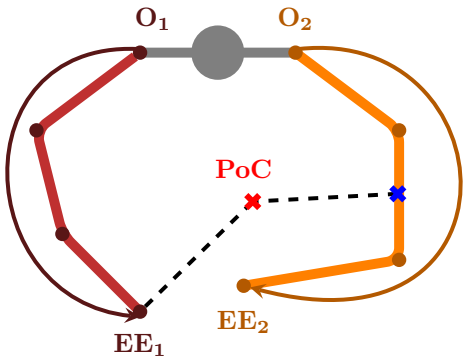
i *Limited number of Degrees of Freedom (DoF) for the task and kinematic constraints*. The closer the desired point of contact to the base of its kinematic chain (i.e. $\mathbf{O_2}$ in Fig. 3a), the smaller number of DoF are usable to position the manipulator in a suitable configuration for being reached by the contralateral one. In addition, the operational space of a manipulator is generally bigger at a certain distance from the base and shrinks as one moves toward the origin of the kinematic chain. However, since many self-touch configurations are located closer to the origin of the chain, this results in poor reachability/manipulability measure.

ii *Defining the point-of-contact (PoC) in operational space*. In view of point i), there is only a limited number of configurations that realize self-touch and sometimes no solution exists. Moreover, the coordinates of the solution are unknown. Therefore, a suitable heuristics is needed in order to find the common solution of two inverse kinematics problems.

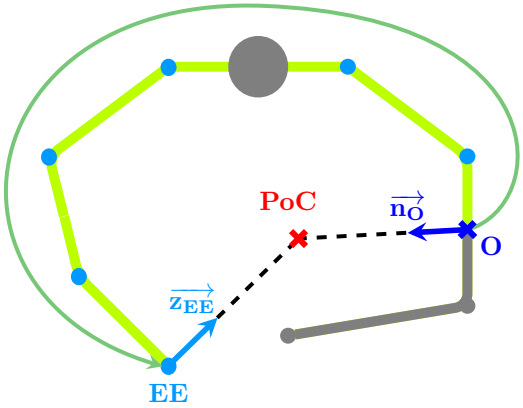
iii *Undesired self-collisions*. Apart from the specifically requested contact point, collisions between other parts can occur. Some body parts are not covered by skin (e.g., joints) and some parts, like fingers, are very fragile.

2) Reformulating the kinematic chain: From two fixed-base serial chains to a single floating-base serial chain: The above mentioned difficulties can be significantly mitigated if the problem is reformulated: instead of parallel control of two kinematic chains, the task can be transformed into the control of a single chain that spans from the point to be touched to the contralateral end-effector. The new situation is schematically depicted in Fig. 3b. Under this reformulation, the task is to move the end-effector $\mathbf{EE}$ to the origin $\mathbf{O}$ of the kinematic chain. Compared to the previous situation, this brings about one key advantage, i.e. the final Cartesian PoC of the two arms is defined *implicitly* since the base of the kinematic chain is floating. The inverse kinematic solution will move both the base and the end-effector in order to make them converge at a specific point of the 3D operational space. To this end, first, one part of the kinematic chain is "reversed"—because it has to be traversed "upside down", from the point to be touched up to the shoulder (O_2 in Figure 3a). Second, an inverse kinematic solver has to be employed to get the solution to this task. The next paragraphs describe the reversal of the kinematic chain, whereas the design of the solver will be detailed in the next section.

The description of the kinematics is typically based on the Denavit Hartenberg (DH) convention [19] with four

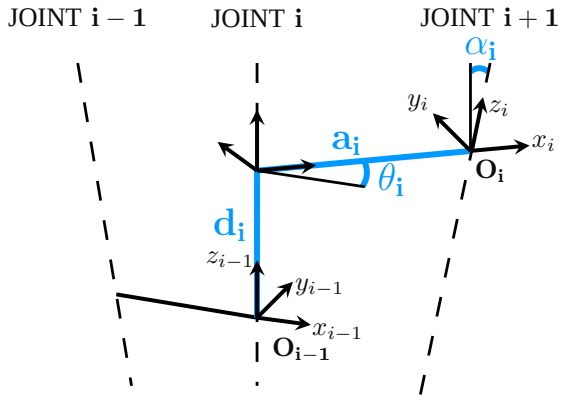


(a) Classical Approach: the two arms, with origins $\mathbf{O_1}$ and $\mathbf{O_2}$, corresponding to the shoulders of the iCub, and end-effectors $\mathbf{EE_1}$ and $\mathbf{EE_2}$ located in the palms of its hands, are controlled in parallel.

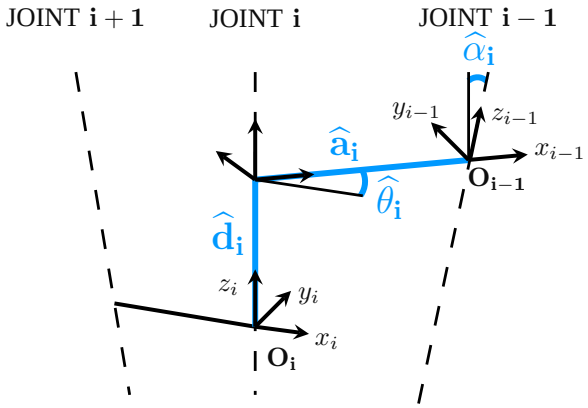


(b) Proposed Approach: the problem has been reformulated into a single floating-base serial chain, with origin $\mathbf{O}$ in the point to be touched and end-effector $\mathbf{EE}$ in the contralateral arm.

Fig. 3: **Comparison of approaches to self-collision.** The blue cross is the point to be touched (left forearm), whereas the red one is the final Cartesian point in which the two arms get in contact (PoC); the grey links are either not controlled or fixed.



(a)



(b)

Fig. 4: **Comparison between the original DH convention (a) and the proposed solution for the reversed chain (b).** In order to "reverse" the original kinematic chain, the reference frames for the links have been kept coincident with the original ones.

parameters for each joint. With an initial choice of axes, it is subsequently possible to compute a homogeneous transform matrix to describe each relative roto-translation from one joint to the next one. However, although these matrices have an inverse, they cannot be transformed into a set of valid DH parameters suitable for the reversed kinematic chain. In the following, we propose a method to compute a DH-compatible set of parameters.

Figure 4 depicts a comparison between the original DH convention for the forward chain and the proposed approach for its reversed version. Figure 4a illustrates the reference frames' attachment for a generic joint i belonging to a chain pointing from left to right, whereas Figure 4b describes the proposed solution for the reversed chain (traversed from right to left). In order to reverse the chain, we changed the order of the joints such that the end-effector of the original direct chain becomes the origin of the new one. However, the placement of the reference frames has been kept the same. This requires a corresponding change of the parameters of every joint. The new parameters, $\widehat{\mathbf{\Phi}}_i = \{\widehat{a}_i,\ \widehat{d}_i,\ \widehat{\alpha}_i,\ \widehat{\theta}_i\}$, are calculated as:

$$\begin{aligned} \langle \widehat{\mathbf{a_0}}, \widehat{\mathbf{a_1}}, \ldots, \widehat{\mathbf{a_n}} \rangle &= \langle -\mathbf{a_n}, -\mathbf{a_{n-1}}, \ldots, -\mathbf{a_0} \rangle \\ \langle \widehat{\mathbf{d_0}}, \widehat{\mathbf{d_1}}, \ldots, \widehat{\mathbf{d_n}} \rangle &= \langle -\mathbf{d_n}, -\mathbf{d_{n-1}}, \ldots, -\mathbf{d_0} \rangle \\ \langle \widehat{\alpha_0}, \widehat{\alpha_1}, \ldots, \widehat{\alpha_n} \rangle &= \langle -\alpha_\mathbf{n}, -\alpha_\mathbf{n-1}, \ldots, -\alpha_\mathbf{0} \rangle \\ \langle \widehat{\theta_0}, \widehat{\theta_1}, \ldots, \widehat{\theta_n} \rangle &= \langle -\theta_\mathbf{n}, -\theta_\mathbf{n-1}, \ldots, -\theta_\mathbf{0} \rangle \end{aligned} \quad (1)$$

where $\mathbf{\Phi}_i = \{a_i,\ d_i,\ \alpha_i,\ \theta_i\}$ is the original set of DH parameters for the i-th link.

The standard transform matrix for the direct chain dDH is shown below:

$$dDH = \begin{bmatrix} c_\theta & -s_\theta c_\alpha & s_\theta s_\alpha & a c_\theta \\ s_\theta & c_\theta c_\alpha & -c_\theta s_\alpha & a s_\theta \\ 0 & s_\alpha & c_\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Under the convention defined above, the transform matrix for the reversed chain, rDH corresponds to the inverse of the standard transform matrix (i.e. dDH^{-1}), but with the new

set of parameters $\widehat{\mathbf{\Phi}}_i$. Substituting $\mathbf{\Phi}_i$ with $\widehat{\mathbf{\Phi}}_i$ in dDH^{-1} thus gives the following form for rDH:

$$rDH = \begin{bmatrix} c_{\widehat{\theta}} & -s_{\widehat{\theta}} & 0 & \widehat{a} \\ s_{\widehat{\theta}}c_{\widehat{\alpha}} & c_{\widehat{\theta}}c_{\widehat{\alpha}} & -s_{\widehat{\alpha}} & -\widehat{d}s_{\widehat{\alpha}} \\ s_{\widehat{\theta}}s_{\widehat{\alpha}} & c_{\widehat{\theta}}s_{\widehat{\alpha}} & c_{\widehat{\alpha}} & \widehat{d}c_{\widehat{\alpha}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Under this approach, the parameters lose their physical meaning (those related to lengths, such as $\widehat{a}_i$ and $\widehat{d}_i$, become negative), but the geometrical representation is correctly preserved. In addition, the existing machinery for the kinematic computation (e.g. the differential kinematics) can be maintained, because the Jacobians are still dependent on the same parameter (i.e. $\widehat{\theta}_i$ for revolute joints).

3) Solver: The task—in the formulation proposed above—is an inverse kinematic task of a serial chain with a floating base. Hence, we consider the problem of computing the values of joint angles $\mathbf{q}^* \in \mathbb{R}^n$ such that the end-effector reaches a given position $\mathbf{x}_d \in \mathbb{R}^6$ (we are considering the desired position and orientation as a single $6D$ vector, rotations are described by the axis-angle representation). In addition, $\mathbf{q}^*$ has to satisfy a number of constraints, expressed as a set of inequalities. Formally, the problem can be stated as follows:

$$\begin{aligned} \mathbf{q}^* &= arg\min_{\mathbf{q}\in\mathbb{R}^n} \langle \mathbf{n_O}, \mathbf{z_{EE}} \rangle = \\ &= arg\min_{\mathbf{q}\in\mathbb{R}^n} \Big\{ \|\mathbf{n_O}\| \cdot \|\mathbf{z_{EE}}\| \cdot cos(\alpha) \Big\} \\ &\qquad s.t. \begin{cases} \|K_x(\mathbf{q}) - \mathbf{O}\|^2 < \epsilon \\ \mathbf{q}_l < \mathbf{q} < \mathbf{q}_u \end{cases} \end{aligned} \tag{2}$$

where $\mathbf{O}$ is the origin of the chain (i.e. $[0\ 0\ 0]$); $\mathbf{n_O}$ is a unit vector perpendicular to the first link of the kinematic chain and originating in $\mathbf{O}$; $\mathbf{z_{EE}}$ is the $z-$axis of the end-effector, as specified in Figure 3b; α is the angle between $\mathbf{n_O}$ and $\mathbf{z_{EE}}$; K_x is the forward kinematic function that represents the position of the end-effector; $\mathbf{q}_l$ and $\mathbf{q}_u$ are vectors describing the joints' lower and upper limits. The optimization criterion is thus the minimization of the scalar product between the $z-$axis of the end-effector and a vector $\mathbf{n_O}$ normal to the surface to be touched. Since both vectors are of unit length, the optimization boils down to the minimization of $cos(\alpha)$, that is the normal of the target and the $z-$axis of the end-effector pointing in exactly opposite directions (where $\alpha = \pi$ and $cos(\alpha) = -1$).

Further, the solution to problem (1) has to satisfy a set of additional constraints: in particular, we require that the end-effector's position is coincident with the origin of the kinematic chain (up to a certain tolerance ϵ), and that the solution lies between a set of lower and upper bounds ($\mathbf{q}_l, \mathbf{q}_u \in \mathbb{R}^n$) of physically admissible values for the joints. This description of the problem entails that the final position and joint limits are always satisfied (being a constrained optimization problem) whereas the orientation of the end-effector may have a residual error (the minimum cost may not be zero).

Kinematically, the task can by explained as a reaching problem (the end-effector reaches the origin) with orientation normal to the surface of the touched point. The particular formulation of the problem (namely, a serial chain with a floating base) implies that both arms are automatically controlled in order to solve the task. Since the iCub arms are redundant, the solver has a certain freedom to impose the final configuration while satisfying the joint limit or collision constraints. Nonetheless, the solution we proposed has dramatically increased the redundancy of the task: the adoption of a single serial chain that spans from the point to be touched up to the contralateral end-effector has further increased the number of degrees of freedom available[1].

In order to solve the problem described by Equation 2, an interior point optimization technique is used, in particular we employ IpOpt [20], a public domain software package designed for large-scale nonlinear optimization problems.

4) Controller: The motor control has been achieved by means of a simple position control in most of the joints. Two of them (namely, the shoulder yaw and the elbow of the left arm) have been controlled in impedance mode, in order to ensure more compliance and thus intrinsic safety during contact.

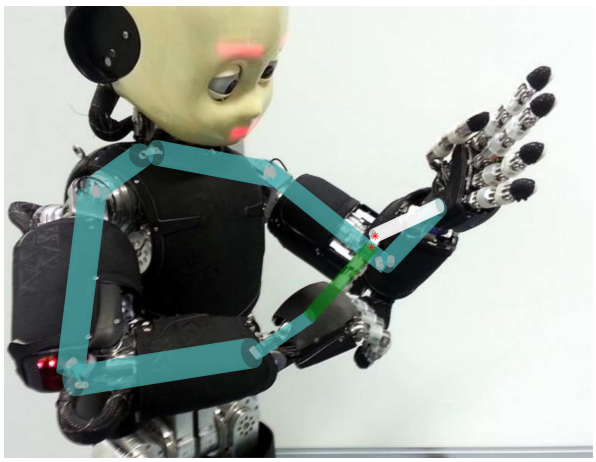
C. Data collection and optimization

1) Experimental protocol: The experimental protocol is schematically illustrated in Fig. 2. The experimenter touches the robot on the left forearm. This is detected by the robot and the position of the active taxel (in fact the average position of all the taxels that are stimulated) is recorded to be used later by the solver. The arms start moving toward the self-touch configuration. If this is successfully achieved, the contact is detected by the robot (on the left forearm and on the tip of the right index finger). The coordinates of this point (relying on the skin calibration) are recorded together with the current joint configuration, and constitute one data point that is later used for subsequent optimization of the kinematic model. Due to inaccuracies in the initial model, not all attempts result in a final self-touch configuration—these trials are ignored. In total, 100 successful data points were collected in this way. Then, in order to speed up data collection (only the final configuration is important for the calibration), further 200 points were collected by setting the joints to idle and generating the self-touch configurations by the experimenters.

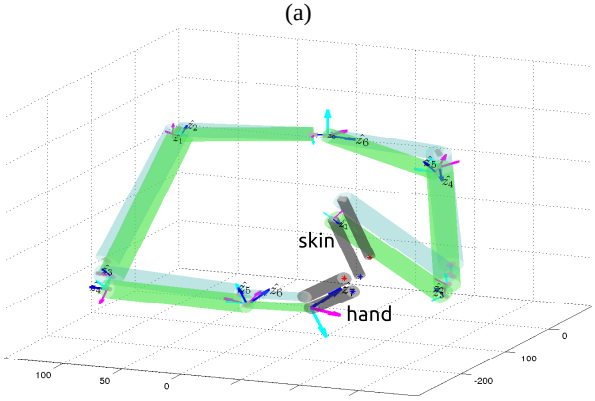
2) Problem formulation: Referring to [2], a kinematic calibration has to optimize the parameter vector $\mathbf{\Phi}_i = a_i, d_i, \alpha_i, o_i$ with $i \in [1, n]$, a, d, α the first three parameters of the DH formulation [19], and o representing the offset that specifies the positioning of the encoders on the joints with respect to the DH representation. Thus, o is part of the model that should be subject to calibration and is therefore also included in the optimization (like in [2]).

[1] In our concrete case, where the point to be touched is located on the robot's left forearm, 12 DoF are used—5 on the manipulator that is touched, 7 in the contralateral one.

In the experimental setup presented here, the kinematic chain consists of 12 DoF. With 4 parameters per joint, the number of parameters to be optimized is thus 48 (see Table I, first column). A schematic illustration of this is depicted in Fig. 5a. The kinematic chain is closed through two fixed transformations at its ends (skin – white in the figure; index finger – green in the figure). These transform matrices are kept constant and their parameters are not optimized.



(a)



(b)

Fig. 5: **Optimization of the kinematic chain**. The chain is closed on one end through the skin (a fixed transformation from the taxel to the wrist) and through the index finger on the other end (another fixed transformation from the contralateral hand to the fingertip). (a) Schematic illustration of the framework. (b) Initial vs. optimized kinematic chain. The initial parameters (CAD values) shown in blue, the optimized in green.

The parameter calibration is obtained by using the same non linear optimizer described earlier (IpOpt [20]). The cost is set in order to minimize the total position error:

$$\mathbf{\Phi}^* = arg\min_{\mathbf{\Phi}} \sum_{m=1}^{M} \|\mathbf{p}_s - \mathbf{p}_e(\mathbf{\Phi}, \theta_m)\| \qquad (3)$$

where θ_m are the joint angles of the m-th sample as read from the joint encoders (for a total of $M = 300$ samples), $\mathbf{p}_e$ is the estimated position (as function of the joint angles and the current parameter values) and $\mathbf{p}_s$ is the position of the end-effector as measured from the skin.

III. EXPERIMENTAL RESULTS

Using the data set collected according to the procedure described in the previous section, we conducted three optimization experiments.

A. Experiment 1 - Optimization of parameters from CAD model

In this experiment, the initial values of the DH parameters are those from the CAD model. Assuming that the skin calibration is correct, these parameters can be used by the optimizer with the aim of minimizing the error of the forward kinematic function w.r.t. the position of the taxels. The result is shown in Fig. 5b. The values after optimization can be seen in the "Optimized (Exp 1)" column of Table I. The measure of performance is the error on the end-effector position—Euclidean distance of the predicted position of the end-effector to the taxel position from the skin calibration—as shown in the first row of Table II. As expected, here the improvement is small owing to the accurate initial estimate of the DH parameters derived from the CAD models.

TABLE II: Error at the end-effector

	Initial (m)	**Optimized (m)**
Exp 1	0.0226	0.0208
Exp 2 (10% noise)	0.0819 ± 0.0299	0.0377 ± 0.0139
Exp 3 (30% noise)	0.1919 ± 0.0301	0.0664 ± 0.0175

B. Experiments 2 and 3 - Optimization with 10 and 30% noise on initial values

The initial parameter values in "Experiment 1" are naturally already reasonable guesses of the real parameters. In order to further test our proposed method, we have conducted additional experiments with more noisy estimates of the initial parameters. Therefore, we perturbed the CAD parameters as follows:

$$\mathbf{\Phi}_i^n = p * uniform[-1; 1] * \mathbf{\Phi}_i + \mathbf{\Phi}_i \qquad p = [0.1, 0.3] \qquad (4)$$

where $\mathbf{\Phi}_i^n$ is the new set of perturbed parameters, p is the amount of noise with a uniform distribution (effectively 10% or 30% of the parameter value), and $\mathbf{\Phi}_i$ is the original set of parameters. The noise is thus proportional to the initial value of every DH parameter (angle or length), with a special consequence that parameters with an initial value of zero are not perturbed. All parameters—including the ones with zero values—are then subject to optimization.

Five different initial configurations were generated and the optimizer run using the same data set (300 data points). The results can be seen in Table I and Table II. In the 10% noise case, the reduction of error is substantial (54% on average - Table II) in all the tests that have been run. In Exp. 3, with 30% noise on the DH parameters, qualitatively similar

conclusions can be drawn: the error on the task is reduced 65% on average after optimization.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a new method for robotic self-calibration that does not rely on any external measurement apparatus or on constraints arising from specific contact with the environment. Furthermore, no sensing at a distance (vision, laser) is needed. Instead, taking inspiration from early infant development and exploiting the artificial skin on the iCub robot, we exploited the correspondences between the tactile and proprioceptive modality—in our case tactile inputs and joint angle values—to calibrate the parameters of a kinematic chain. The data sets were collected using a novel self-touch behavior that is generated autonomously by the robot: the inverse kinematic solver was relying on an advantageous reformulation of the reaching problem for the two arms of the iCub into a single floating-base kinematic chain. Then, optimization of the DH parameters was performed through minimization of the distance between positions predicted by forward kinematics and known positions of the taxels on the robotic skin. An improvement over the CAD values was achieved. Furthermore, configurations with 10 and 30% noise on the initial DH parameters were also subject to optimization and resulted in an average improvement of 54 and 65% respectively.

As part of our future work we will investigate a variation on the self-touch scenario where the robot will keep a contact configuration for an extended period of time while varying the joint configurations. The utility of data sets originating from this "closed-loop" strategy will be compared with the data set used in this initial work. Furthermore, several additional sources of inaccuracy were not considered here. These include the skin calibration, the kinematics of the hand (from wrist to fingertip), or the precision of the joint measurements. These could be subject to optimization in the future as well. At the same time, while self-touch theoretically provides the means to come close to perfect accuracy when reaching for one's own body, in the future, we want to investigate extrapolation of the calibration to the whole operational space. Finally, additional loops can be closed by adding the visual modality. Using stereo vision and adding the head and eye kinematics while keeping the same methodology—that is the point where "double touch" occurs can be observed—will allow for calibration of all the remaining components, including camera projective maps. A theoretical analysis of the observability and identifiability with respect to the contributions of different data collection methods and individual sensory modalities will be performed.

In our view, this work is not only relevant as a part of a new, compact self-calibration tool for robots with sensorized skin, but paves the way for robots to enter new application domains. Robots capable of reaching with any body part and also able to detect and locate contact on any body part can be said to posses whole-body awareness, making them intrinsically safe not only for themselves (as demonstrated here), but also for their environment, possibly a humanly populated one. This is certainly one of the overarching goals and future measure of success of humanoid robotics.

REFERENCES

[1] J. Hollerbach and C. Wampler, "The calibration index and taxonomy for robotic kinematic calibration methods," *International Journal of Robotics Research*, vol. 15(6), pp. 573–591, 1996.

[2] J. Hollerbach, W. Khalil, and M. Gautier, *Handbook of robotics*, ch. Model identification, pp. 321–343. Springer, 2008.

[3] C. Gatla, R. Lumia, J. Wood, and G. Starr, "An automated method to calibrate industrial robots using a virtual closed kinematic chain," *IEEE Transactions on Robotics*, vol. 23 (6), pp. 1105–1116, 2007.

[4] J.-S. Hu, J.-J. Wang, and Y.-J. Chang, "Kinematic calibration of manipulator using single laser pointer," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 426–430, IEEE, 2012.

[5] M. Hersch, E. Sauser, and A. Billard, "Online learning of the body schema," *International Journal of Humanoid Robotics*, vol. 5, pp. 161–181, 2008.

[6] R. Martinez-Cantin, M. Lopes, and L. Montesano, "Body schema acquisition through active learning," in *Proc. Int. Conf. on Robotics and Automation (ICRA)*, 2010.

[7] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens mvn: full 6dof human motion tracking using miniature inertial sensors," *Xsens Motion Technologies BV, Tech. Rep*, 2009.

[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*, pp. 4539–4545, IEEE, 2012.

[9] P. Mittendorfer and G. Cheng, "Humanoid multimodal tactile-sensing modules," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 401–410, 2011.

[10] M. Hoffmann, H. Marques, A. Hernandez Arieta, H. Sumioka, M. Lungarella, and R. Pfeifer, "Body schema in robotics: a review," *IEEE Trans. Auton. Mental Develop.*, vol. 2 (4), pp. 304–324, 2010.

[11] P. Rochat, "Self-perception and action in infancy," *Exp Brain Res*, vol. 123, pp. 102–109, 1998.

[12] S. Schutz-Bosbach, J. Musil, and P. Haggard, "Touchant-touché: The role of self-touch in the representation of body structure," *Consciousness and Cognition*, vol. 18, no. 1, pp. 2 – 11, 2009.

[13] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, "The iCub humanoid robot: An open-systems platform for research in cognitive development," *Neural Networks*, vol. 23, no. 8-9, pp. 1125–1134, 2010.

[14] A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, G. Cannata, and G. Metta, "Methods and technologies for the implementation of large-scale robot tactile sensors," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 389 –400, 2011.

[15] A. Parmiggiani, M. Maggiali, L. Natale, F. Nori, A. Schmitz, N. Tsagarakis, J. S. Victor, F. Becchi, G. Sandini, and G. Metta, "The design of the iCub humanoid robot," *International Journal of Humanoid Robotics*, vol. 9, no. 04, 2012.

[16] R. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing - from humans to humanoids," *Robotics, IEEE Transactions on*, vol. 26, no. 1, pp. 1–20, 2010.

[17] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight manipulator arm," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 1623–1630, IEEE, 2006.

[18] A. Del Prete, S. Denei, L. Natale, F. M., F. Nori, G. Cannata, and G. Metta, "Skin spatial calibration using force/torque measurements," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pp. 3694 –3700, 2011.

[19] J. Denavit, "A kinematic notation for lower-pair mechanisms based on matrices.," *Trans. of the ASME. Journal of Applied Mechanics*, vol. 22, pp. 215–221, 1955.

[20] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

TABLE I: DH Parameters and optimization

Label	CAD model	Optimized (Exp 1)	Optimized $\pm\sigma$ (Exp 2 – 10% noise)	Optimized $\pm\sigma$ (Exp 2 – 30% noise)
a_1	0	0	0 ± 0	0 ± 0
d_1	-0.1373	-0.1373	-0.1343 ± 0.0097	-0.1477 ± 0.0206
α_1	-1.5708	-1.5968	-1.6292 ± 0.0787	-1.7148 ± 0.1119
o_1	1.5708	1.5887	1.5770 ± 0.1134	1.8133 ± 0.2217
a_2	0.0150	0.0126	0.0129 ± 0.0027	0.0051 ± 0.0073
d_2	0	-0.0026	-0.0020 ± 0.0028	-0.0074 ± 0.0083
α_2	-1.5708	-1.6014	-1.5165 ± 0.0703	-1.5658 ± 0.3268
o_2	0	-0.0040	-0.0013 ± 0.0572	0.0194 ± 0.1127
a_3	-0.0150	-0.0179	-0.0166 ± 0.0029	-0.0236 ± 0.0086
d_3	-0.1523	-0.1604	-0.1560 ± 0.0050	-0.1963 ± 0.0187
α_3	1.5708	1.5887	1.5848 ± 0.0526	1.6144 ± 0.2614
o_3	-1.3090	-1.3360	-1.3508 ± 0.0511	-1.3663 ± 0.4323
a_4	0	-0.0064	-0.0029 ± 0.0029	-0.0195 ± 0.0164
d_4	0	-0.0060	-0.0029 ± 0.0029	-0.0190 ± 0.0168
α_4	-1.5708	-1.5487	-1.5295 ± 0.0496	-1.4276 ± 0.3057
o_4	1.5708	1.5841	1.5163 ± 0.1037	1.5072 ± 0.2475
a_5	0	0.0216	-0.0029 ± 0.0029	-0.0162 ± 0.0232
d_5	-0.1077	-0.1153	-0.1099 ± 0.0042	-0.1175 ± 0.0187
α_5	1.5708	1.5841	1.5660 ± 0.1299	1.7101 ± 0.3909
o_5	-1.5708	-1.6014	-1.5712 ± 0.0439	-1.7181 ± 0.3643
a_6	0	-0.0060	-0.0029 ± 0.0029	-0.0217 ± 0.0194
d_6	-0.1077	-0.1119	-0.1123 ± 0.0072	-0.1222 ± 0.0288
α_6	1.5708	1.5839	1.5609 ± 0.0982	1.5766 ± 0.1594
o_6	-1.5708	-1.6024	-1.6297 ± 0.0964	-1.6054 ± 0.2078
a_7	0	-0.0069	-0.0041 ± 0.0052	0.0166 ± 0.0479
d_7	0	-0.0030	-0.0029 ± 0.0029	-0.0155 ± 0.0203
α_7	-1.5708	-1.5963	-1.5644 ± 0.0352	-1.5986 ± 0.2675
o_7	-1.5708	-1.5963	-1.5716 ± 0.0841	-1.6173 ± 0.3691
a_8	-0.0150	-0.0171	-0.0172 ± 0.0029	-0.0170 ± 0.0150
d_8	-0.1523	-0.1605	-0.1541 ± 0.0077	-0.1176 ± 0.0496
α_8	-1.5708	-1.5831	-1.5748 ± 0.1046	-1.6635 ± 0.2511
o_8	-1.8326	-1.8618	-1.8268 ± 0.1184	-1.8297 ± 0.2338
a_9	0.0150	0.0133	0.0129 ± 0.0030	0.0012 ± 0.0296
d_9	0	-0.0026	-0.0028 ± 0.0028	-0.0080 ± 0.0080
α_9	1.5708	1.5891	1.5990 ± 0.1019	1.5853 ± 0.2720
o_9	0	-0.0040	0.0011 ± 0.0565	0.0095 ± 0.0985
a_{10}	0	-0.0019	-0.0025 ± 0.0037	-0.0050 ± 0.0144
d_{10}	-0.1373	-0.1411	-0.1410 ± 0.0093	-0.1279 ± 0.0395
α_{10}	1.5708	1.5915	1.6521 ± 0.0582	1.6108 ± 0.2230
o_{10}	-1.5708	-1.5937	-1.6092 ± 0.1299	-1.6252 ± 0.3956
a_{11}	0	0.0008	-0.0028 ± 0.0028	-0.0045 ± 0.0148
d_{11}	0	-0.0011	-0.0029 ± 0.0029	-0.0046 ± 0.0147
α_{11}	1.5708	1.5918	1.6273 ± 0.1152	1.5263 ± 0.2460
o_{11}	1.5708	1.5915	1.6771 ± 0.0766	1.5118 ± 0.3519
a_{12}	0.0625	0.0623	0.0579 ± 0.0038	0.0528 ± 0.0099
d_{12}	0.0160	0.0151	0.0136 ± 0.0026	0.0102 ± 0.0142
α_{12}	0	-0.0012	-0.0102 ± 0.0838	-0.0244 ± 0.1054
o_{12}	3.1416	3.1842	3.1370 ± 0.1743	3.0448 ± 0.4766