# Incremental adaptation of a robot body schema based on touch events

Rodrigo Zenha[1], Alexandre Bernardino[1], José Santos Vitor[1]

***Abstract*— The term 'body schema' refers to a computational representation of a physical body; the neural representation of a human body, or the numerical representation of a robot body. In both humans and robots, such a representation is crucial to accurately control body movements. While humans learn and continuously adapt their body schema based on multimodal perception and neural plasticity, robots are typically assigned with a fixed analytical model (*e.g.*, the robot kinematics) which describes their bodies. However, there are always discrepancies between a model and the real robot, and they vary over time, thus affecting the accuracy of movement control. In this work, we equip a humanoid robot with the ability to incrementally estimate such model inaccuracies by touching known planar surfaces (*e.g.*, walls) in its vicinity through motor babbling exploration, effectively adapting its own body schema based on the contact information alone. The problem is formulated as an adaptive parameter estimation (Extended Kalman Filter) which makes use of planar constraints obtained at each contact detection. We compare different incremental update methods through an extensive set of experiments with a realistic simulation of the iCub humanoid robot, showing that the model inaccuracies can be reduced by more than 80%.**

## I. INTRODUCTION

Humans develop a neural representation of their body (*i.e.*, a body schema [1]) through an incremental learning process that starts in early infancy [2], and likely even prenatally [3], and goes through continuous adaptations over time, based on multimodal sensorimotor information acquired during motor experience [4]: visual, tactile, proprioceptive. This (physical) body schema is a crucial part of human self-awareness and it supports the precise control of body movements, coping with the morphological changes that occur in the body over time, *e.g.*: body growth, tool assimilation.

Clearly, endowing artificial agents with similar learning and adaptation capabilities is a major challenge for cognitive robotics and it paves the way for the next generation of robots able to act in complex environments. Indeed, robots are expected to play an increasingly active role in different sectors of the human society, from industrial manufacturing [5], [6] to healthcare [7] and education [8]. In all these applications, accurate models of the robot structure (*i.e.*, kinematics) are required for nearly all robotic tasks, to interact with objects and even more crucially to safely interact with people [9].

A variety of factors, such as difficult-to-model transmissions, friction, worn joints and bended rigid bodies, induce changes to the robot kinematic model over time. As a consequence, robots need to perform off-line calibration procedures from time to time in order to preserve their reliableness. However, such procedures are time-consuming, they require to stop the robot normal operations, and they need to be triggered by an explicit detection of a miscalibration issue. Online incremental procedures that can be performed by the robot during its normal operations are therefore more desirable, in any robotic application.

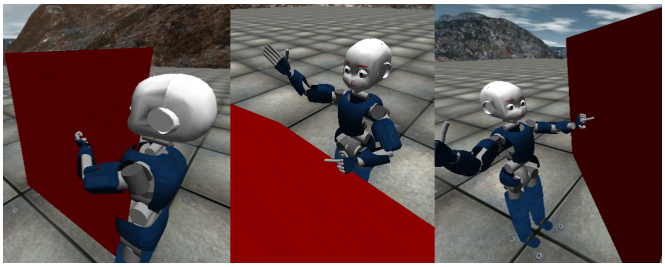


Fig. 1. Simulation of an iCub robot reaching for three different walls in its workspace.

Humanoid robots, such as the iCub [10] (shown in Fig. 1), typically rely on on-board sensors (more than on external ones) to control their movements, and therefore it would be best to perform such continuous calibration based on those sensors alone, without having to employ external devices. Ideally, information from different on-board sensors should be combined to increase the robustness and reliability of the calibration (in other terms, of the body schema adaptation).

In this work we develop an incremental calibration strategy that is performed automatically by the robot during the execution of any arm movement that involves contacts on known planar surfaces, using the Extended Kalman Filter for adaptive parameter estimation. To do so, we make use of contact (pressure sensitive fingertips) and proprioception (joint encoders) sensors, commonly present in many humanoid robots, such as the iCub. Notably, we compare six different approaches for the update of the body schema from new data obtained after each contact with a planar surface, by performing an extensive set of experiments with a simulated iCub robot. Interestingly, our techniques are robot-independent, and they could be used on any robot equipped with joint angle position sensors and contact sensing capabilities at the end-effector. Contacts can be detected not only with pressure sensors but also with force, acceleration or motor current sensing.

The rest of the paper is organized as follows. In Section II, we highlight the most relevant related work. Then, in Section III we describe the problem, while in Section IV we explain in detail our proposed solution. In Section V, we describe

[1] Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal. rodrigo.n.zenha@tecnico.ulisboa.pt , {alex,jasv}@isr.tecnico.ulisboa.pt

the simulation setup, the robotic platform used and details on the solution implementation. Finally, in Section VI, we present the experimental results, and in Section VII we draw our conclusions and sketch future work.

## II. RELATED WORK

Robots and humans need to have an accurate model of their bodies, in particular of arms and hands in order to be able to reach and manipulate objects in their workspace. To do so, robots typically rely on an analytical model which enables them to perform motion control.

A number of works have been developed in order to make use of sensory feedback, mainly visual, so as to actively estimate the robot's hand pose and visually controlling reaching tasks [11], [12], [13], [14]. The techniques proposed in previous works provide an efficient way of controlling motion towards a specific non-occluded object, but they do not help the robot to make use of model-based planners to control more general motions. We can find in literature methods that compensate for local errors during the robot's manipulator trajectory (*e.g.*, using visual servoing) [11], [15], however, these works do not consider sensorimotor data to actively learn an internal model representation. Body schema learning has been widely studied. In [16], we can find a work review of this topic up to 2010. In robotics, we can formulate the adaptation of the robot body schema as a self-calibration problem, where the robot's internal model parameters are estimated (learned) online during motor experience. Some works first attempt to solve the body schema and hand pose estimation by using either a camera attached to a moving end-effector matching points in the scene from consecutive frames [17], or Laser Tracking Systems (LTS) where a specific marker is placed on the end-effector tracking it to achieve a robust robot calibration [18]. Both cases make use of sensors that are not usual to humanoid robots.

Contact information was used in [19] to develop an off-line automatic kinematic chain calibration resorting to self-touch events, which was proven to be highly effective to optimize the robot's model Denavit-Hartenberg (DH) parameters. One setback of this method is that it does not consider the joint angle measurements inaccuracies; also, a sensitive skin covering the robot's body is needed to detect self-touch events, which might not be available on every robotic platform.

On-line solutions using on-board cameras have been studied [20], [21], [22], in which visual markers are used to easily detect the end-effector position; the inclusion of additional parts into the kinematic chain (*i.e.*, tools) has been considered as well [23]. Methods to perform on-line calibration, *e.g.*, during reaching tasks, based on contact information have also been studied. That is the case in [24] where they set to improve the robot's estimate of its configurations using an implicit *Manifold Particle Filter* informed by contact sensors during periods of persistent contact. In addition, goal-directed strategies in which the robot learns about its internal model during the execution of a goal-driven task, such as the ones presented in [25] (goal-directed exploration) and [26] (goal babbling), have been observed to enhance the body schema learning, for example, by reducing the time necessary for robot calibration convergence.

Vicente *et al.* [27] compare data from the robot's vision, proprioception and a realistic 3D computer graphics model of the robot [28] in order to estimate simultaneously the robot's hand pose and kinematic model, making use of GPU programming to implement a *Particle Filter*. Kolev *et al.* [29] combine data from proprioception, motion capture position and orientation, and force sensors, acquired while interacting with objects in its workspace, so as to build an internal simulation of the robot's configurations (using MuJoCo, a physics engine for model-based control) [30], and estimate the model parameters values that better explain the visualized results.

We propose a novel approach to adapt the robot's body schema using proprioception and haptic sensors which is capable of being executed online during robot's normal operations. Information about contacts on known planar surfaces feed a low computational cost estimation method (Extended Kalman Filter) enabling real-time body schema adaptation.

## III. PROBLEM STATEMENT

### A. Body Schema Modeling

The adaptation of the body schema can be seen as an internal process that occurs in the mind of the robot using its own perception to achieve a better internal model representation of the self. We focus on the calibration of the arms and hands by exploiting haptic perception on the fingertips. The body schema, from a robotics perspective, is embedded as the kinematic chain from the root reference frame to the end-effector and it is used to derive the forward model (to predict from the motor commands where the end-effector is in the 3D task-space): $\mathbf{T}^{\mathrm{e}} = \mathcal{K}(\boldsymbol{\theta})$; and the inverse model (to anticipate the motor commands to reach a given target in the 3D task-space): $\boldsymbol{\theta} = \mathcal{K}^{-1}(\mathbf{T}^{\mathrm{e}})$. Let us define $\mathbf{T}^{\mathrm{e}}$ as a 4x4 roto-translation matrix which encapsulates the pose (position and orientation) of the end-effector on the root reference frame. Due to modulation errors we can only get an estimation of the robot kinematics function ($\widehat{\mathcal{K}}(\cdot)$) based on the joint angles ($\boldsymbol{\theta}$) can be used to retrieve it:

$$\mathbf{T}^{\mathrm{e}} = \widehat{\mathcal{K}}(\boldsymbol{\theta}), \tag{1}$$

where $\mathcal{K}(\boldsymbol{\theta})$ is the true robot kinematics. However, due to the existence of calibration errors (bias), the real joints angles are different from the ones read from proprioception (joints encoders): $\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{p}} + \boldsymbol{\beta}$, where $\boldsymbol{\theta}$ are the real angles values, $\boldsymbol{\theta}^{\mathrm{p}}$ are the joint encoders readings (proprioception) and $\boldsymbol{\beta}$ are the angular offsets. To better estimate the robot kinematics we reformulate Eq. (1) to account for the joint offsets estimate $\widehat{\boldsymbol{\beta}}$ :

$$\mathbf{T}^{\mathrm{e}} = \widehat{\mathcal{K}}(\boldsymbol{\theta}^{\mathrm{p}} + \widehat{\boldsymbol{\beta}}). \tag{2}$$

The parameter vector and our state (to be estimated recursively) is defined as follows:

$$\boldsymbol{\beta} = [\beta^1 \beta^2 \dots \beta^N]^T, \tag{3}$$

where $N$ is the number of degrees of freedom of the robot's manipulator. Assuming the joint offsets to be slowly varying in time, we define the system's next state to be equal to the previous one adding a multivariate zero-mean Gaussian noise $\boldsymbol{\varepsilon}_t$, which allows us to account for small changes on the system state:

$$\boldsymbol{\beta}_t = \boldsymbol{\beta}_{t-1} + \boldsymbol{\varepsilon}_t. \tag{4}$$

### B. Observation Model

The observation model relates the system state $\boldsymbol{\beta}$ with a single measurement ($z_k$) from the tactile sensors. We assume that there is a planar surface described by:

$$x \cdot n_x + y \cdot n_y + z \cdot n_z - d = 0, \tag{5}$$

where $\mathbf{n} = [n_x, n_y, n_z]^T$ define the plane's normal vector ($||\mathbf{n}|| = 1$), and $d$ is the plane's minimum distance to the robot root reference frame. In a simulation environment, both $\mathbf{n}$ and $d$ are known *a priori*. On real robot experiments, vision sensing can be used to estimate the surface's pose on the robot vicinity, *e.g.*, using the Aruco board marker [31] or computing a planar fit on depth point clouds from stereo vision.

When a contact occurs, we are ensuring that the arm's end-effector position ($\mathbf{x}^e = (x^e, y^e, z^e)^T$) respects Eq. (5). However, due to errors in the kinematic model, each set of coordinates $\widehat{\mathbf{x}}_k^e = (\widehat{x}_k^e, \widehat{y}_k^e, \widehat{z}_k^e)^T$ at an instant $k$, follows the equation:

$$\widehat{x}_k^e \cdot n_x + \widehat{y}_k^e \cdot n_y + \widehat{z}_k^e \cdot n_z - d = \alpha_k, \tag{6}$$

where $\alpha_k$ is the error produced by the model inaccuracies.

The observation model is then defined as:

$$z_k(\boldsymbol{\theta}_k^p + \widehat{\boldsymbol{\beta}}_t) = \underbrace{\widehat{\mathbf{x}}_k^e \cdot \mathbf{n} - d}_{\alpha_k} + \delta_k, \tag{7}$$

where $\widehat{\mathbf{x}}_k^e$ is retrieved using the forward kinematics (Eq. (2)), $\widehat{\boldsymbol{\beta}}_t$ are the offsets estimation at time instate $t$ and $\delta_k$ is the error associated to an observation.

## IV. Approach

We develop a body schema adaptation strategy estimating the angular offsets $\boldsymbol{\beta}$ by exploiting contact constraints obtained at each end-effector contact with a surface in the robot's vicinity. The strategy devised can be divided into two steps: i) a motor babbling exploration strategy towards the target planar surface, stopping when tactile stimuli in the index finger occurs; and ii) a learning phase where an Extended Kalman Filter is fed with multisensory input (*i.e.*, proprioception, surface characteristics and haptic feedback) adapting the state $\boldsymbol{\beta}$.

### A. Workspace Exploration

The robot performs a motor babbling movement towards the desired surface using the joint space ($\boldsymbol{\theta}$) as the input command. The generation of the babbling movement is based on the geometric Jacobian ($\mathbf{J}_e(\boldsymbol{\theta})$) of the manipulator:

$$\dot{\mathbf{x}}^e = \mathbf{J}_e(\boldsymbol{\theta}) \cdot \dot{\boldsymbol{\theta}} \;\; ; \;\; \mathbf{J}_e(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{J}_e^v(\boldsymbol{\theta})_{3\times N} \\ \mathbf{J}_e^\omega(\boldsymbol{\theta})_{3\times N} \end{bmatrix}_{6\times N}, \tag{8}$$

which relates the $N$ joint velocities ($\dot{\boldsymbol{\theta}}$) with the end-effector task-space velocities ($\dot{\mathbf{x}}^e$). The sub-matrices $\mathbf{J}_e^v(\boldsymbol{\theta})$ and $\mathbf{J}_e^\omega(\boldsymbol{\theta})$ describe the linear and angular task-space velocities for a given $\dot{\boldsymbol{\theta}}$, respectively. To ensure that the babbling movement is in the direction of the target surface we test the condition:

$$(\mathbf{J}_e^v(\boldsymbol{\theta}) \cdot \dot{\boldsymbol{\theta}})^T \cdot \mathbf{n} \leq 0, \tag{9}$$

where $\mathbf{n}$ is the surface normal. If the condition is not satisfied, the motor command is discarded ensuring the babbling to be towards the target surface. Moreover, the exploration stops when a contact is detected by the haptic sensors.

### B. Parameter Estimation - Non-Linear Least Squares

A first approach to estimating the robot joint offsets, defined in Eq. (3), is to formulate the problem in a Non-Linear Least Squares (NLS) setting. This method enables a system to fit a set of $m$ observations with a non-linear model with $N$ unknown parameters (provided that $m \geq N$). It is an inherently offline method since it follows a batch estimation approach (all observations data must be retrieved before an estimation can be computed).

To find the best unknown parameters values which describe a set of observations, the NLS regression is cast as an optimization problem over an objective function, which somehow depends on the unknown parameters ($f(\boldsymbol{\beta})$):

$$\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta}} \frac{1}{2} \sum_{k=1}^{m} r_k(\boldsymbol{\beta})^2. \tag{10}$$

where $r_k(\boldsymbol{\beta}) = y_k - z(\boldsymbol{\theta}_k^p + \boldsymbol{\beta})$ ($y_k$ are the measured data corresponding to $\boldsymbol{\theta}_k^p$). The nonlinearity arises from $z(\boldsymbol{\theta}_k^p + \boldsymbol{\beta})$ (see Eq. (7)).

The solution to Eq. (10), $\boldsymbol{\beta}^*$, needs to be refined iteratively, *i.e.*, the values are obtained by successive approximations. For NLS problems, full batch methods are more accurate and stable as they iteratively refine the linearization point, compared to recursive estimation techniques.

The Gauss-Newton method is frequently used to solve NLS problems. The method generalizes Newton's method, an iterative algorithm for finding the roots of a non-linear function. It relies in a search direction vector $p^{GN}$ to iteratively update the $\widehat{\boldsymbol{\beta}}$ values:

$$\widehat{\boldsymbol{\beta}}_t = \widehat{\boldsymbol{\beta}}_{t-1} - p^{GN}, \tag{11}$$

and $p^{GN}$ is determined from solving the following equation:

$$\nabla^2 f(\widehat{\boldsymbol{\beta}}_{t-1}) \cdot p^{GN} = -\nabla f(\widehat{\boldsymbol{\beta}}_{t-1}). \tag{12}$$

Given that the Jacobian $J(\boldsymbol{\beta})$ of a set of $m$ residuals $r(\boldsymbol{\beta})$ is defined as

$$J(\boldsymbol{\beta}) = \begin{bmatrix} \nabla r_1(\boldsymbol{\beta})^T & \nabla r_2(\boldsymbol{\beta})^T & \dots & \nabla r_m(\boldsymbol{\beta})^T \end{bmatrix}^T, \tag{13}$$

the elements of the gradient of $f(\boldsymbol{\beta})$ are given by:

$$[\nabla f(\boldsymbol{\beta})]_j = \frac{\partial f(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{m} r_i(\boldsymbol{\beta}) \frac{\partial r_i(\boldsymbol{\beta})}{\partial \beta_j}, \tag{14}$$

and it follows that the gradient is the vector

$$\nabla f(\boldsymbol{\beta}) = J(\boldsymbol{\beta})^T r(\boldsymbol{\beta}). \tag{15}$$

Moreover, assuming $r_i(\boldsymbol{\beta}^*) = 0$ close to a solution $\boldsymbol{\beta}^*$, the Hessian of $f(\boldsymbol{\beta})$ is given by:

$$\nabla^2 f(\boldsymbol{\beta}) = \nabla r(\boldsymbol{\beta}) \nabla r(\boldsymbol{\beta})^T = J(\boldsymbol{\beta})^T J(\boldsymbol{\beta}), \tag{16}$$

and the search direction defined in Eq. (12) has the following solution:

$$p^{GN} = -(J(\boldsymbol{\beta})^T J(\boldsymbol{\beta}))^{-1} J(\boldsymbol{\beta})^T r(\boldsymbol{\beta}). \tag{17}$$

### C. Parameter Estimation - Extended Kalman Filter

Our goal is to estimate the robot joints offsets defined in Eq. (3). For that purpose, we use an Extended Kalman Filter (EKF), commonly used for online parameter estimation. By assuming that each $\beta^i$ is distributed through multivariate normal distributions (with mean $\mu$ and covariance $\boldsymbol{\Sigma}$), we are in conditions of using the EKF. Using the proposed dynamic Eq. (4) and observation model Eq. (7), we get the following EKF equations:

1) Prediction:

$$\overline{\boldsymbol{\beta_t}} = \widehat{\boldsymbol{\beta}}_{t-1} + \boldsymbol{\varepsilon_t}, \tag{18a}$$

$$\overline{\boldsymbol{\Sigma}}_t = \boldsymbol{\Sigma}_{t-1} + \mathbf{Q}_t, \tag{18b}$$

2) Kalman Gain and Update:

$$\mathbf{K}_t = \overline{\boldsymbol{\Sigma}}_t \mathbf{H}_t^T (\mathbf{H}_t \overline{\boldsymbol{\Sigma}}_t \mathbf{H}_t^T + \mathbf{R}_t)^{-1}, \tag{18c}$$

$$\widehat{\boldsymbol{\beta}}_t = \overline{\boldsymbol{\beta_t}} + \mathbf{K}_t(\mathbf{0} - \mathbf{z}_t), \tag{18d}$$

$$\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \overline{\boldsymbol{\beta_t}}, \tag{18e}$$

where $\widehat{\boldsymbol{\beta}}_t$ and $\Sigma_t$ are the current offsets estimation and covariance matrix, respectively. $\mathbf{Q}_t$ and $\mathbf{R}_t$ are positive semi-definite covariance matrices and $\mathbf{z}_t$ and $\mathbf{H}_t$ encapsulate a set of observations.

When a contact is detected and the joint encoders readings are retrieved, we are in conditions of performing a measurement and acquire an observation $z_k$ as described in Eq. (7). At each contact event, we compute the current observation measurement ($z_k$) as well as $\mathbf{H}_k$ which is obtained from its the gradient: $\mathbf{H}_k = \nabla z_k(\boldsymbol{\theta}_k^p + \widehat{\boldsymbol{\beta}}_t)$. We can rewrite it as:

$$\mathbf{H}_k = \mathbf{n} \cdot \mathbf{J}_e^v(\boldsymbol{\theta}_k^p + \widehat{\boldsymbol{\beta}}_t), \tag{19}$$

where $\mathbf{J}_e^v(\boldsymbol{\theta}_k^p + \widehat{\boldsymbol{\beta}}_t)$ is the sub-matrix defined in Eq. (8).

We combine and evaluate 3 different new data incorporation strategies:

*1) Aggregation of Multiple Observations:* Coupling together a varying number of contact constraints ($k$) before a filter update step ($t$). Depending on how many measures we want to couple together, matrices $\mathbf{H}_t$ and $\mathbf{z}_t$ are subject to changes in their respective dimensions:

$$\mathbf{H}_t = \begin{bmatrix} \mathbf{H}_{k-n} & \cdots & \mathbf{H}_{k-1} & \mathbf{H}_k \end{bmatrix}^T, \tag{20a}$$

$$\mathbf{z}_t = \begin{bmatrix} z_{k-n} & \cdots & z_{k-1} & z_k \end{bmatrix}^T, \tag{20b}$$

where $t$ is the instant when the filter performs an estimation step, and the number of contact events detected up until $t$ is $n+1$. When we use only one observation for an estimation update step ($n = 0$) we have $\mathbf{H}_t \equiv \mathbf{H}_k$ and $\mathbf{z}_t \equiv z_k$.

*2) Estimation Differential Entropy Evaluation:* Upon each contact, we get $\mathbf{z}_t$ and $\mathbf{H}_t$ and compute the predicted next step estimation covariance matrix $\boldsymbol{\Sigma}_t$. Following the approach used in [32], we decide to incorporate the new data if the current estimation differential entropy decreases when compared with the previous estimation:

$$\frac{1}{2} \log_e \frac{|\boldsymbol{\Sigma}_{t-1}|}{|\boldsymbol{\Sigma}_t|} > 0, \tag{21}$$

and discard new data that does not bring innovative information to the estimation. Here $|.|$ denotes a matrix determinant and $\boldsymbol{\Sigma}_{t-1}$ is the current estimation covariance matrix.

*3) Anti-Windup Control (A-W):* A common problem in recursive parameter estimation is windup in system state covariance matrix $\boldsymbol{\Sigma}_t$ during periods of poor excitation. This phenomenon happens when some eigenvalues of $\boldsymbol{\Sigma}_t$ increase uncontrollably to large values when highly correlated data is incorporated, making the filter oversensitive to noise and with long transients when relevant data is obtained, leading the system to incorrect estimates. In [33] a technique is described in order to avoid windup when using recursive parameter estimation methods, such as the Recursive Least Squares (which can be recast into a EKF) by controlling the parameter random walk covariance matrix, $\mathbf{Q}(t)$, so as to get $\boldsymbol{\Sigma}_t$ to achieve a constant pre-defined covariance matrix, $\mathbf{P}_d$, thus avoiding it to get unacceptable large eigenvalues. We use the same technique adapted to the EKF framework:

$$\mathbf{Q}(t) = \frac{\mathbf{P}_d \mathbf{H}_t \mathbf{H}_t^T \mathbf{P}_d}{\mathbf{R}(t) + \mathbf{H}_t^T \mathbf{P}_d \mathbf{H}_t}. \tag{22}$$

## V. Experimental Setup

The algorithms developed are tested in an iCub simulator [34]. iCub is an humanoid robot developed in the context of the EU project RobotCub (2004-2010) and since then it has been adopted by more than 35 laboratories worldwide. iCub is open-source, with the hardware design, software and documentation all available for free. iCub provides proprioception information about its joint encoders readings, and it is also possible to attach an artificial skin which gives us feedback (together with tactile fingers) about established contacts. To communicate with the simulator we use the YARP middleware which allow us to build a robot control system as a collection of programs communicating in a peer-to-peer way. The iCub internal model is provided within the YARP/iCub software framework.

The simulation setup is composed of three reachable surfaces in the iCub vicinity with *a priori* known normal vectors and minimum distances, relative to the robot's root frame, which provide a rich environment for the robot to acquire contact constraints: $\mathbf{n}^1 = (\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2})$ and $d^1 = \frac{-\sqrt{2}}{8}$m, $\mathbf{n}^2 = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0)$ and $d^2 = \frac{-3\sqrt{2}}{10}$m, $\mathbf{n}^3 = (0, -1, 0)$ and $d^3 = -0.05$m. For the first set of experiments, we make the robot constantly reach for the surface described by $\mathbf{n}^1$ and $d^1$. We then perform a second set of experiments where the contact events alternate between all three surfaces. Figure 1 shows the simulated iCub robot reaching for these surfaces.

We control the robot's left arm and define its left index fingertip to be the manipulator's end effector. We artificially simulate angular offsets on the seven DoF of the iCub's left arm fixating the joint of the index finger. We define $\boldsymbol{\beta} = \left[-11, 11, -7, -17, -7, -17, 7\right]^T$deg, whose values have the same order of magnitude of the calibration errors we typically encounter on the real robot. All simulations are performed relative to these offsets, except for the last set of experiments, where we use 2 different artificially introduced offsets sets in order to test the strategy robustness.

An interesting feature of the iCub simulator is that the robotic kinematic model in use already reveals inherent inaccuracies, such as it would be expected from real robot experiments. This allows us to get some sense about the behavior of our calibration strategy in real robot experiments, more precisely, we are able to evaluate how non-considered inaccuracies affect the strategy global performance. Figure 2 shows relative frequency histogram of the Cartesian distances (metrical errors) values from 300 different contacts on all three surfaces mentioned before, assuming no joint's angular offsets. The obtained dataset mean is of 2.7 mm, with a standard deviation of 3.4 mm.

We present next the simulated results for the devised calibration strategy, comparing all new data incorporation methods (see subsection V-B), running a total of ten simulations for each method. For each experiment, we perform a total of 45 contacts (49 for the EKF 7-contact setting).

### A. Error Metrics

After each filter update, we compute the global estimation root mean squared error (RMSE) relative to the real offsets:

$$\text{RMSE}(\widehat{\boldsymbol{\beta}}) = \sqrt{\frac{1}{7}\sum_{i=1}^{7}(\widehat{\beta}^i - \beta^i)^2}. \tag{23}$$

After the calibration process is finished we make the end-effector reach for the a surface and compute the end-effector Cartesian error, considering the final $\widehat{\boldsymbol{\beta}}$ values. The Cartesian distance between the real end-effector position and the estimated one is obtained as follows:

$$E_{\text{Cart}}(\boldsymbol{\theta}^p, \widehat{\boldsymbol{\beta}}) = \sqrt{(x^e - \widehat{x}^e)^2 + (y^e - \widehat{y}^e)^2 + (z^e - \widehat{z}^e)^2}. \tag{24}$$

Remember that $\widehat{\mathbf{x}}^e$ is obtained from Eq. (2).

### B. New Data Incorporation methods

Next, we provide a detailed description of the different new data incorporations methods which are evaluated next in the results section. These methods result from combining one, two or three different strategies mentioned in IV-C:

**a) 7-Contact (7C):**

Upon each contact detection, $z_k$ and $\mathbf{H}_k$ are stored in $\mathbf{z}_t$ and $\mathbf{H}_t$, respectively. The system performs an estimation step after it collects 7 contact constraints (equal to the number of the iCub's arm DoFs) obtained from 7 contact events.

**b) Single Contact (SC):**

Upon each contact event, $z_k$ and $\mathbf{H}_k$ are fed to the filter and the systems perform an estimation update step.

**c) Single Contact with Entropy (SC-E):**

Similar to the previous technique, but now for each new data obtained the system computes the predicted next step estimation covariance matrix and evaluates whether or not the new observation actively contributes to the estimation differential entropy reduction using Eq. (21). If the condition is not satisfied, the new observation is discarded.

**d) Varying-Contact with Entropy (VC-E):**

Equivalent to the previous method, but each time a new observation fails to reduce the global estimation differential entropy, instead of being discarded, $z_k$ and $\mathbf{H}_k$ are added to the matrices $\mathbf{z}_t$ and $\mathbf{H}_t$, respectively. Every time new data is obtained, the system evaluates if $\mathbf{z}_t$ and $\mathbf{H}_t$ are able to reduce the entropy of the next step estimation. If so, an update step is performed regarding all previously stored observations.

**e) Single Contact with A-W (SC-AW):**

Equivalent to the Single Contact Estimation method, but $\mathbf{Q}_t$ is controlled with the anti-windup technique described in Eq. (22), rather than being a predefined matrix.

**f) Single Contact with Entropy and A-W (SC-EAW):**

The system discards every new observation data which fails to reduce the next step estimation differential entropy, and controls $\mathbf{Q}_t$ matrix so as to avoid estimation windup from uncorrelated measures.

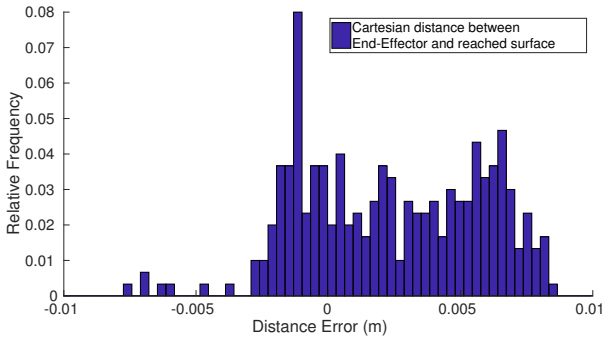



Fig. 2. Histogram of Cartesian distance error between end-effector and three different reached surfaces, for 300 reaching events.

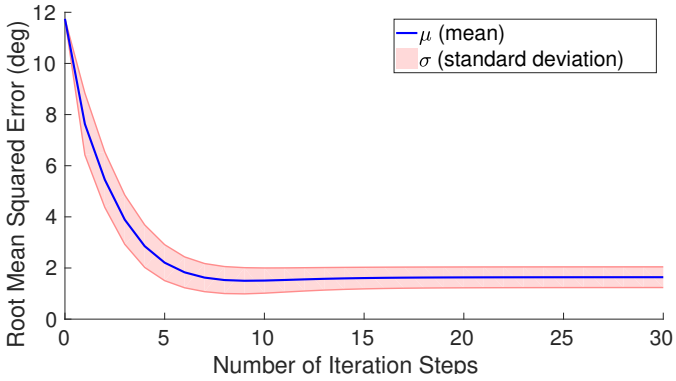
## VI. Results

We evaluate the results of $\boldsymbol{\beta}$ estimation for contacts in a single, and in three different surfaces[1]. All experimental results are shown in Tables I and II. Tables I depicts each estimation filter (and all new data incorporation strategies, for the EKF setting) global performances, providing the final RMSE($\widehat{\boldsymbol{\beta}}$) estimation error mean ($\mu[^\circ]$) and standard deviation ($\sigma[^\circ]$) over 10 experiments after 45 contact events. Table II shows the final $E_{\text{Cart}}(\boldsymbol{\theta}^p, \widehat{\boldsymbol{\beta}})$ error mean ($\mu$[m]) and standard deviation ($\sigma$[m]) for the same methods, number of experiments and number contacts. We resort to Section V-B to identify the methods during the results presentation.

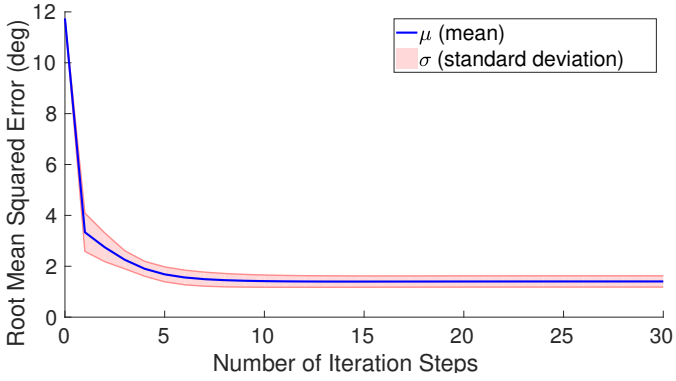
### A. Contacts over a single surface

We begin by testing the Gauss-Newton batch estimation outcome for 45 contact constraints on a single surface. This experiment allow us to understand what is the tangible estimation lower bound, given the number of contact constraints considered, which later is used to evaluate the performance of the EKF adaptive filter. Fig. 3(a) shows the 10 experiments mean outcome of the RMSE($\widehat{\boldsymbol{\beta}}$) evolution after each iterative step taken, where the mean $\mu[^\circ]$ is represented by a blue line, and the respective standard deviation $\sigma[^\circ]$ is represented on a shaded red region. The method is observed to reduce the estimation error by $86\%$ and keeping a low $\sigma[^\circ]$ for all the iteration steps. Given these results, we are able to conclude, with some confidence, that after 45 contact events in a single surface, the global $\boldsymbol{\beta}$ estimation error lower bound value is of $\mu[^\circ]$ = 1.64 deg (see Table I). Furthermore, looking at Table II, we observe the $E_{\text{Cartesian}}(\boldsymbol{\theta}^p, \widehat{\boldsymbol{\beta}})$ value to be $\mu$[m] = 7 mm, the same order of magnitude seen in section V for the errors' values of a robotic system with no joints' angular offsets.

[1]a video of the proposed approach can be found in `http://soma.isr.tecnico.ulisboa.pt/vislab_data/ICDL2018bodyschema/ICDL2018bodyschema.mp4`




(a) Contacts over a single surface.




(b) Contacts over three different surfaces.

Fig. 3. RMSE($\widehat{\boldsymbol{\beta}}$) (mean and standard deviation) over 10 experiments for $\boldsymbol{\beta}$ estimation using the Gauss-Newton method with 45 contacts.

Next, we evaluate the results of $\boldsymbol{\beta}$ estimation with the EKF, starting with the method **7C** (7-Contact Estimation). Looking at Fig. 4(a) we see the RMSE($\widehat{\boldsymbol{\beta}}$) evolution over the 10 experiments, through 49 contact events, corresponding to a total of 7 filter update steps. The estimation error decreases slowly over each estimation step, and the model inaccuracies are reduced by $50\%$ ($\mu[^\circ]$ = 5.87deg) after 35 contacts (5 estimation update steps).

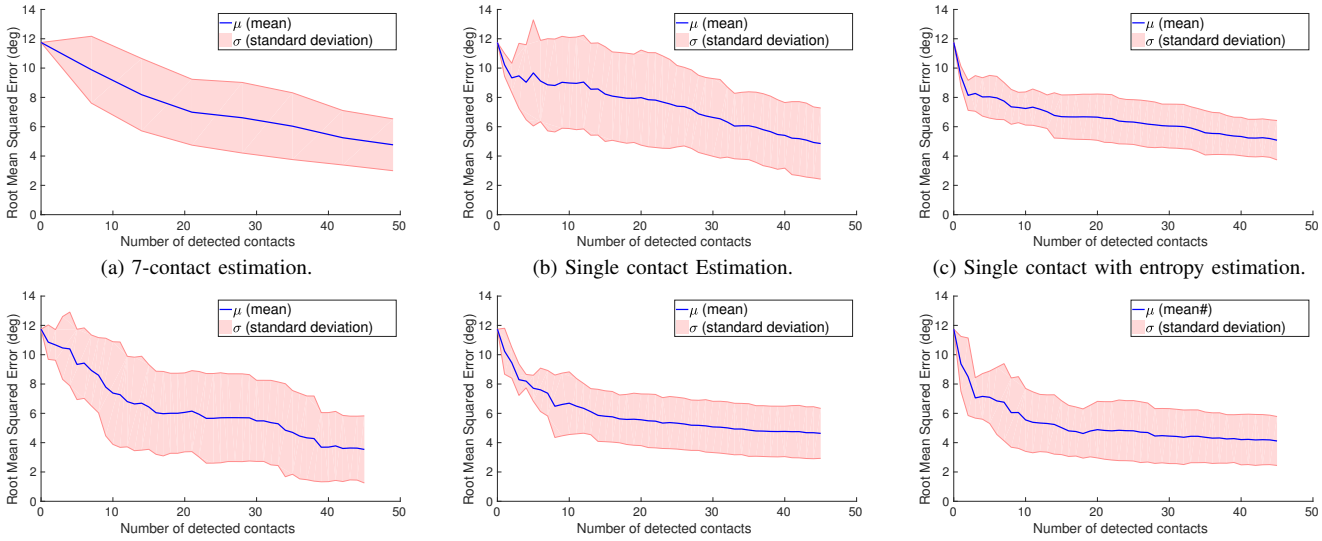
The results for method **SC** can be seen in Fig. 4(b), which show a worse performance. The system keeps a slow steady error descend during the whole estimation, reducing the estimation absolute error by $50\%$ after 37 contacts. Moreover, $\sigma[^\circ]$ remains relatively high for all $t$, due to the slow reaction of the filter against estimation steps taken in the wrong direction during periods of poor excitation.

We try to improve method **SC** by evaluating the estimation differential entropy (Fig. 4(c)) or using the A-W control technique (Eq. (22)) (Fig. 4(e)). For the first solution (method **SC-E**), we conclude that it is able to stabilize the filter performance, since the global estimation $\sigma[^\circ]$ is reduced compared to the simple single contact setting (see Table I). However, this happens at the cost of the filter converging sooner to a minimum, since it is not able to easily find observations which reduce the estimation entropy. Furthermore, the estimation error does not increase at any moment, opposed to the simpler method **SC**. The later solution (method **SC-AW**) achieves a better performance as well, being able to stabilize the filter. Its key improvements are: i) higher pronounced error reduction slope until the 8th

TABLE I

Final RMSE($\boldsymbol{\beta}$) mean ($\mu$) and standard deviation ($\sigma$) values.

| Estimation method | | Three Surfaces | | One Surface | |
|---|---|---|---|---|---|
| | | $\mu[^\circ]$ | $\sigma[^\circ]$ | $\mu[^\circ]$ | $\sigma[^\circ]$ |
| NLS | Gauss-Newton | 1.40 | 0.22 | 1.64 | 0.41 |
| EKF | 7C (a.) | 3.36 | 1.41 | 5.05 | 1.76 |
| | SC (b.) | 2.30 | 1.00 | 4.85 | 2.42 |
| | SC-E (c.) | 5.00 | 2.06 | 5.08 | 1.34 |
| | VC-E (d.) | 2.41 | 1.02 | 3.53 | 2.28 |
| | SC-AW (e.) | 2.67 | 0.80 | 4.63 | 1.70 |
| | SC-EAW (f.) | 2.20 | 0.74 | 4.11 | 1.66 |
| Without calibration ($\widehat{\boldsymbol{\beta}} = \mathbf{0}_{7\times1}$) | | 11.74 | - | 11.74 | - |




(a) 7-contact estimation. (b) Single contact Estimation. (c) Single contact with entropy estimation.

(d) Varying-contact with entropy estimation. (e) Single contact with A-W estimation. (f) Single contact with entropy and A-W.

Fig. 4. RMSE (mean and standard deviation) over 10 experiment for each new data incorporation method described in V-B for $\boldsymbol{\beta}$ estimation using an EFK and contacts on one surface.

TABLE II

FINAL $E_{\text{CART}}(\boldsymbol{\theta}^p, \widehat{\boldsymbol{\beta}})$ MEAN ($\mu$) AND STANDARD DEVIATION ($\sigma$) VALUES.

| **Estimation method** | | Three Surfaces | | One Surface | |
|---|---|---|---|---|---|
| | | $\mu$[m] | $\sigma$[m] | $\mu$[m] | $\sigma$[m] |
| NLS | Gauss-Newton | 0.006 | 0.002 | 0.007 | 0.016 |
| EKF | 7C (a.) | 0.014 | 0.006 | 0.028 | 0.015 |
| | SC (b.) | 0.011 | 0.005 | 0.026 | 0.011 |
| | SC-E (c.) | 0.063 | 0.035 | 0.049 | 0.022 |
| | VC-E (d.) | 0.011 | 0.004 | 0.023 | 0.010 |
| | SC-AW (e.) | 0.012 | 0.008 | 0.022 | 0.012 |
| | SC-EAW (f.) | 0.011 | 0.010 | 0.020 | 0.012 |
| Without calibration ($\widehat{\boldsymbol{\beta}} = \mathbf{0}_{7\times1}$) | | 0.168 | - | 0.168 | - |

contact; ii) reduction the absolute error by 50% after 14 contacts, also improving the final solution (after 45 contacts) on $\boldsymbol{\beta}$ estimation (see Table I).

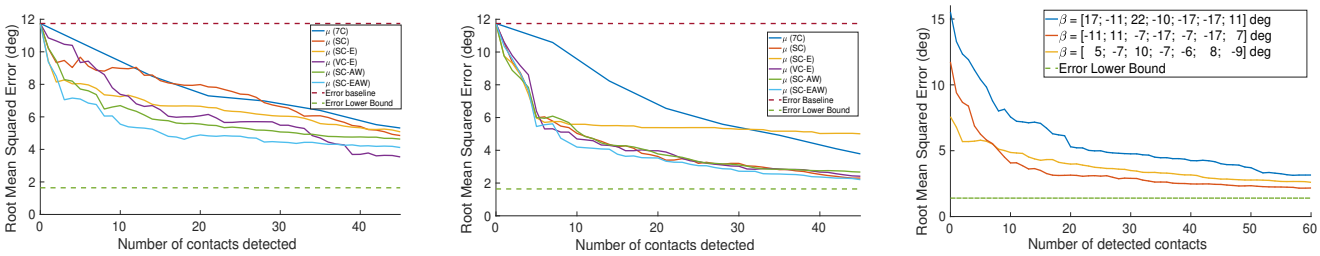
The method **VC-E** is a mixed model combining the entropy evaluation with the method **7C**. Then, we enable the filter to adapt coupling together a sufficient number of observations to perform an estimation update step with lower entropy. The estimation keeps a steady error descend up until the 16th contact (reducing the model inaccuracies by $50\%$), slowing the pace for the next contacts (see Fig. 4(d)).

Finally, in Fig. 4(f), we see the results of combining both the A-W control technique and estimation entropy evaluation on a single contact strategy (method **SC-EAW**). We are able to notice three key features: i) the system is able to converge to a lower overall estimation minimum (reducing the estimation error by $15\%$ compared to the single contact setting, and $65\%$ overall, after 45 contact events), ii) the overall experiments $\sigma[^\circ]$ is 30% lower compared to method **SC**, and iii) the system converges faster to a minimum (requires 10 steps to reduce the model inaccuracies by 50%). Combining both techniques we are able to get a more stable and precise filter. Fig. 5(a) helps us to visually compare the performance of all methods, as well as to evaluate the estimation error reduction relative to the estimation lower bound ($\mu[^\circ]$ = 1.64 deg) obtained from the batch estimation described in the previous subsection.

From Table. II we are also able to evaluate the results of the end-effector position error ($E_{\text{Cartesian}}(\boldsymbol{\theta}^p, \widehat{\boldsymbol{\beta}})$) improvements for the final joint's offsets estimated values. The initial end-effector position error, before calibration, is (approximately) $\mu$[m] = 168 mm. Since the **VC-E** method is the one which reduces the RMSE($\widehat{\boldsymbol{\beta}}$) the most, it would be expectable that the corresponding $E_{\text{Cartesian}}(\boldsymbol{\theta}^p, \widehat{\boldsymbol{\beta}})$ value would also be the lowest. However, that is not case because the **VC-E** method estimated poorly the offsets values corresponding to the joints which influence the kinematic chain motion the most, *i.e.* those closest to the kinematic chain root, opposed to the **SC-EAW** method, which is seen to reach the lowest $E_{\text{Cartesian}}(\boldsymbol{\theta}^p, \widehat{\boldsymbol{\beta}})$ value, reducing the end-effector position error by 88% ($\mu$[m] = 20mm).

### *B. Contacts over three different surfaces*

By broadening the robot spatial exploration in order to perform contacts on 3 surfaces, we expect an overall better $\boldsymbol{\beta}$ estimation performance, since contact constraints obtained in this manner provide to the filter richer information. Once again we evaluate the $\boldsymbol{\beta}$ estimation using the Gauss-Newton iterative method, for 45 contacts constraints. Fig. 3(b) shows the RMSE($\widehat{\boldsymbol{\beta}}$) evolution after each iterative step taken. It's interesting to notice that the results obtained for contacts on a single or three different surfaces are very similar (see Table I), suggesting once again that the tangible estimation error lower bound considered previously is very close to the




(a) Comparison between all data incorporation methods using contacts over one surface.

(b) Comparison between all data incorporation methods using contacts over three surfaces.

(c) 3 different offsets sets estimation using the SC-EAW method and contacts on 3 surfaces.

Fig. 5. $\boldsymbol{\beta}$ estimation performances comparison between all data incorporation methods and for contacts contacts events over one or three surfaces.

real one. We also notice that the $E_{\text{Cartesian}}(\boldsymbol{\theta}^p, \widehat{\boldsymbol{\beta}})$ error value ($\mu$[m] = 6 mm) has the same order of magnitude as the values obtained for a robotic system with no joint's angular offsets.

Looking at Fig. 5(b) we are able to visually analyze the different new data incorporation methods (EKF setting) performance in this scenario.

We can clearly assess from Fig. 5(b) that the methods **7C** and **SC-E** are the ones which benefit less from information acquired from contacts on 3 different surfaces since they both converge to the highest estimation errors minima. The **7C** setting has a steady slow error reduction slope due to not being able to quickly compensate for estimation steps given on wrong directions. The **SC-E** estimation converges early (5th contact) to a local minimum, not being able to easily find relevant observations from there.

The remaining methods, however, show relevant improvements relative to the single surface scenario. We are able to conclude that by enhancing the **7C** and **SC** methods with an estimation entropy evaluation (**7C** setting) or performing Anti-Windup control (**SC** setting) we improve the filter performance, whether by reducing the global mean estimation error or its $\sigma[^\circ]$ (see table I for detailed values).

The best results are obtained for the single contact setting with both estimation entropy evaluation and anti-windup control techniques, reducing both the estimation error by $45\%$ relative to the single surface scenario (and 80% overall), and presenting the lowest overall $\sigma[^\circ]$ value. Comparatively to the scenario with contacts over a single surface, this method is able to get much closer to the estimation error lower bound ($\mu[^\circ]$ = 1.40 deg). As a result of this, the $E_{\text{Cartesian}}(\boldsymbol{\theta}^p, \widehat{\boldsymbol{\beta}})$ value obtained is also lower (approximately by 50%) comparing to the previous scenario, reaching an error value of $\mu$[m] = 11mm, which is already very close to the error values observed for a robotic system with no joints' angular offsets.

In Fig. 5(c) we see the results of using the method **SC-EAW** for $\boldsymbol{\beta}$ estimation of 3 different artificially introduced offsets readings (10 experiments for each set). Up until the 60th contact event, all experiments reach an estimation minimum; moreover, independently of the true readings offsets, the filter is always able to reduce the estimation error to approximately $\mu[^\circ]$= 2.5deg, illustrating the reliability of the devised strategy.

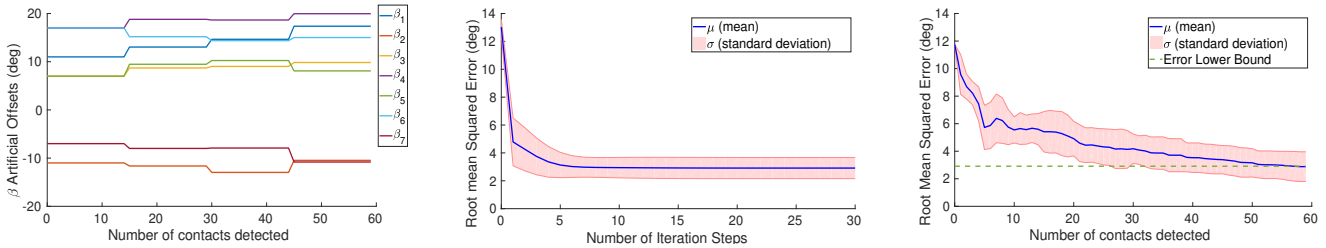
Finally, we set to evaluate the filter robustness when estimating slowly varying offsets values, over 60 contact events. To simulate this scenario we start by introducing the artificial $\boldsymbol{\beta}$ values set described in section V, but this time we induce random small changes to the offsets values after every 15 contacts events. Fig. 6(a) shows an example of the artificially introduced offsets evolution over different 60 contact events, taken from from one calibration run. The offsets values variation is achieved by the addition of random values with Gaussian distribution, with zero mean and a standard deviation of 1.7deg.

We start by evaluating the outcome of the Gauss-Newton method to the aforementioned scenario. Looking at Fig. 6(b) we can clearly see that the method converges to an RMSE($\widehat{\boldsymbol{\beta}}$) value of $\mu[^\circ]$ = 2.91deg (and $\sigma[^\circ]$ = 0.76deg), naturally higher than the one obtained for the fixed artificially introduced offsets scenario ($\mu[^\circ]$ = 1.40deg), since batch estimation techniques need to assume the system state being estimated to be constant in time. We can also presume that the this error would only increase if more contact constraints were retrieved.

Comparing these to the estimation results of the **SC-EAW** method (Fig. 6(c)) to the same scenario, we reckon that the system stabilizes around $\mu[^\circ]$ = 2.80deg, whose value is slightly lower than the results obtained for the Gauss-Newton (batch) estimation ($\mu[^\circ]$ = 2.91deg), and corresponding to an improvement of $76\%$ relative to the initial true offsets values. Moreover, the Cartesian error is observed to lower to $\mu$[m] = 0.014m (and $\sigma$[m] = 0.007m), close to results obtained for the same method for fixed $\boldsymbol{\beta}$ scenario. This experiment demonstrates that the system is capable of coping with small changes in $\boldsymbol{\beta}$ throughout time, achieving a low estimation error after 60 contact, with a low standard deviation ($\sigma[^\circ]$ = 0.73deg), and even outperforming the batch estimation setting.

## VII. CONCLUSIONS AND FUTURE WORK

We devise a novel approach for online body schema adaptation, implemented on the iCub humanoid robot, leading to subsequent improvements in the end-effector pose estimation. The robot's arm joints offsets estimation is performed using an EKF fed with contact constraints obtained during the execution of reaching tasks. We rely on typical humanoid robot's embedded sensors (tactile sensors and proprioception). Our strategy is inspired by the human incremental




(a) Slowly changing artificial offsets until 60 contacts events, taken from one calibration run.

(b) NLS setting with the Gauss-Newton iterative method. RMSE($\widehat{\boldsymbol{\beta}}$) (mean and standard deviation) evolution over 10 calibration runs.

(c) EKF setting with the **SC-EAW** method. RMSE($\widehat{\boldsymbol{\beta}}$) (mean and standard deviation) evolution over 10 calibration runs.

Fig. 6. Estimation performances of slowly varying $\boldsymbol{\beta}$ between the Gauss-Newton and **SC-EAW** methods for 60 contacts in three different surfaces.

learning process of their own body schema, as we use tactile feedback in order to enable the robot to learn about its internal model.

One can conclude from our experiments that by making the robot perform a spatial exploration on an information richer workspace (*i.e.*, availability on different surfaces) the robot is more effective in reducing its model inaccuracies. Overall, our simulation experiments show that we can reduce the model parameter inaccuracies up to $80\%$ by performing contacts on 3 different surfaces, and up to $63\%$ for contacts on a single surface.

In future work we intend to test the developed strategy on the real-world scenario, making use of vision sensing to extract information about surfaces' pose on the robot vicinity. Another possible direction would be to implement a different adaptive estimation filter from the EKF, like the Unscented Kalman Filter or the Particle Filter, to cope with deviations from Gaussianity in the distribution of the estimated parameters.

## Acknowledgments

This work was partially supported by Fundação para a Ciência e a Tecnologia (project UID/EEA/50009/2013 and Grant PD/BD/135115/2017), and by EPSRC UK (project NCNR, National Centre for Nuclear Robotics, EP/R02572X/1). We acknowledge the support of NVIDIA Corporation with the donation of the GPU used for this research.

## References

[1] G. Berlucchi and S. Aglioti, "The body in the brain: neural bases of corporeal awareness," *Trends in Neurosciences*, no. 20(12), pp. 560–564, 1997.

[2] P. J. Marshall and A. N. Meltzoff, "Body maps in the infant brain," *Trends in Cognitive Sciences*, no. 19(9), pp. 499–505, 2015.

[3] R. Joseph, "Fetal brain behavior and cognitive development," *Developmental Review*, vol. 20, pp. 81–98, 2000.

[4] C. von Hofsten, "An action perspective on motor development," *Trends in Cognitive Sciences*, vol. 8, pp. 266–272, 2004.

[5] M. Pedersen, L. Nalpantidis, R. Andersen, C. Schou, S. Bagh, V. Kruger, and O. Madsen, "Robot skills for manufacturing: From concept to industrial deployment," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 282 – 291, 2016.

[6] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human-robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, 2018.

[7] T. Mettler, M. Sprenger, and R. Winter, "Service robots in hospitals: new perspectives on niche evolution and technology affordances," *European Journal of Information Systems*, no. 5, pp. 451–468, 2017.

[8] B. Robins, K. Dautenhahn, R. Te Boekhorst, and A. Billard, "Robotic assistants in therapy and education of children with autism: can a small humanoid robot help encourage social interaction skills?" *Universal Access in the Information Society*, vol. 4, no. 2, pp. 105–120, 2005.

[9] A. De Luca and F. Flacco, "Integrated control for phri: Collision avoidance, detection, reaction and collaboration," in *IEEE RAS & EMBS Int. Conf. on Biomedical Robotics and Biomechatronics*, 2012, pp. 288–295.

[10] 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, p. 1250027, 2012.

[11] A. Miller and B. Leibowitz, "Integration of vision, force and tactile sensing for grasping," *Int. J. Intell. Mach.*, vol. 4, no. 1, pp. 129–149, 1999.

[12] F. Reinhart and J. Steil, "Reaching movement generation with a recurrent neural network based on learning inverse kinematics for the humanoid robot icub," in *Int. Conf. on Humanoid Robots - Humanoids 2009*. IEEE, pp. 323–330.

[13] T. Schmidt, A. Newcombe, and D. Fox, "Dart: Dense articulated real-time tracking." in *Robotics: Science and Systems*, 2014.

[14] M. Peniak and A. Cangelosi, "Scaling-up action learning neuro-controllers with gpus," in *International Joint Conference on Neural Networks*. IEEE, 2014, pp. 2519–2524.

[15] T. Schmidt, K. Hertkorn, R. Newcombe, Z. Marton, M. Suppa, and D. Fox, "Depth-based tracking with physical constraints for robot manipulation," in *2015 IEEE ICRA*, pp. 119–126.

[16] M. Hoffmann, H. Marques, A. Arieta, H. Sumioka, M. Lungarella, and R. Pfeifer, "Body schema in robotics: a review," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 4, pp. 304–324, 2010.

[17] Y. Meng and H. Zhuang, "Autonomous robot calibration using vision technology," *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 4, pp. 436–446, 2007.

[18] Y. Bai, H. Zhuang, and S. Roth, "Experiment study of puma robot calibration using a laser tracking system," in *2003 IEEE International Workshop on Soft Computing in Industrial Applications, 2003.*, 2003, pp. 139–144.

[19] A. Roncone, M. Hoffmann, U. Pattacini, and G. Metta, "Automatic kinematic chain calibration using artificial skin: self-touch in the icub humanoid robot," in *2014 IEEE ICRA*, pp. 2305–2312.

[20] S. Ulbrich, V. de Angulo, T. Asfour, C. Torras, and R. Dillmann, "Rapid learning of humanoid body schemas with kinematic bezier maps," in *IEEE-RAS International Conference on Humanoid Robots*, 2009, pp. 431–438.

[21] L. Jamone, L. Natale, G. Metta, F. Nori, and G. Sandini, "Autonomous online learning of reaching behavior in a humanoid robot," *International Journal of Humanoid Robotics*, no. 9(3), pp. 1 250 017(1–26), 2012.

[22] L. Jamone, M. Brandao, L. Natale, K. Hashimoto, G. Sandini, and A. Takanishi, "Autonomous online generation of a motor representation of the workspace for intelligent whole-body reaching," *Robotics and Autonomous Systems*, no. 64(4), pp. 556–567, 2014.

[23] L. Jamone, B. Damas, N. Endo, V. Santos, and A. Takanishi, "Incremental development of multiple tool models for robotic reaching through autonomous exploration," *PALADYN Journal of Behavioral Robotics*, no. 3(3), pp. 113–127, 2013.

[24] M. Klingensmith, C. Koval, S. Srinivasa, S. Pollard, and M. Kaess,

"The manifold particle filter for state estimation on high-dimensional implicit manifolds," *2017 IEEE ICRA*, pp. 4673–4680.

[25] L. Jamone, L. Natale, K. Hashimoto, G. Sandini, and A. Takanishi, "Learning task space control through goal directed exploration," in *2011 IEEE ROBIO*, pp. 702–708.

[26] M. Rolf, "Goal babbling with unknown ranges: A direction-sampling approach," in *2013 IEEE ICDL*, pp. 1–7.

[27] 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.

[28] ——, "Robotic hand pose estimation based on stereo vision and gpu-enabled internal graphical simulation," *Journal of Intelligent & Robotic Systems*, vol. 83, no. 3-4, pp. 339–358, 2016.

[29] S. Kolev and E. Todorov, "Physically consistent state estimation and system identification for contacts," in *2015 IEEE-RAS on Humanoid Robots*, pp. 1036–1043.

[30] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ Int. Conf. on IROS*, pp. 5026–5033.

[31] S. Jurado, R. Salinas, J. Cuevas, and J. Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[32] J. Maye, H. Sommer, G. Agamennoni, R. Siegwart, and P. Furgale, "Online self-calibration for robotic systems," *The International Journal of Robotics Research*, vol. 35, no. 4, pp. 357–380, 2016.

[33] B. Stenlund and F. Gustafsson, "Avoiding windup in recursive parameter estimation," *Preprints of reglermöte 2002*, pp. 148–153, 2002.

[34] V. Tikhanoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale, and F. Nori, "An open-source simulator for cognitive robotics research: the prototype of the icub humanoid robot simulator," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*. ACM, 2008, pp. 57–61.