Active Robot Learning for Efficient Body-Schema Online Adaptation

1st Gonçalo Cunha
Institute for Systems and Robotics
Instituto Superior Técnico
Lisbon, Portugal
goncalocarvalhocunha@gmail.com

***Abstract*—Humanoid robots have complex bodies and kinematic chains with several Degrees-of-Freedom (DoF) which are difficult to model. Learning the parameters of a kinematic model can be achieved by observing the pose of the robot links during prospective motions and minimising the prediction errors. This thesis proposes a movement efficient approach for estimating online the body-schema of a humanoid robot arm in the form of Denavit-Hartenberg (DH) parameters. A cost-sensitive active learning approach based on the A-Optimality criterion is used to select optimal joint configurations. The chosen joint configurations simultaneously minimise the error in the estimation of the body schema and minimise the movement between samples. This reduces energy consumption, along with mechanical fatigue and wear, while not compromising the learning accuracy. The work was implemented in a simulation environment, using the 7 DoF arm of the iCub robot simulator. The hand pose is measured with a single camera via markers placed in the palm and back of the robot's hand. It is proposed a pose dependent noise model, to reduce the impact of non-uniform measurement noise, and a non-parametric occlusion model, to avoid choosing joint configurations where the markers are not visible, thus preventing worthless attempts. The results show cost-sensitive active learning has similar accuracy to the standard active learning approach, while reducing in about half the executed movement.**

***Index Terms*—cost-sensitive active learning, body-schema, calibration, humanoid, robotics**

I. Introduction

Robots are generally deployed to have a fixed behaviour in low uncertainty environments (e.g. factories) and they rely on their body-schema to accomplish many of their tasks. [1] defines body-schema as an ”implicit knowledge structure that encodes the body's form, the constraints on how the body's parts can be configured, and the consequences of this configuration on touch, vision and movement” and it is how the term is used in this thesis. In [2] several approaches are shown on how to represent the body-schema, as well as an interesting discussion about what is still missing in modern robots to be as robust as animals or human beings. Generally, robots require expensive and time consuming calibrations performed by experts since body parts: i) may not have the exact dimensions they should and ii) they may be affected by material wear and fatigue. Even with these offline calibration procedures, the presence of abnormal conditions or disturbances, such as changes in room temperature causing materials to expand or contract, may affect their performance if they lack the ability to adapt their models online.

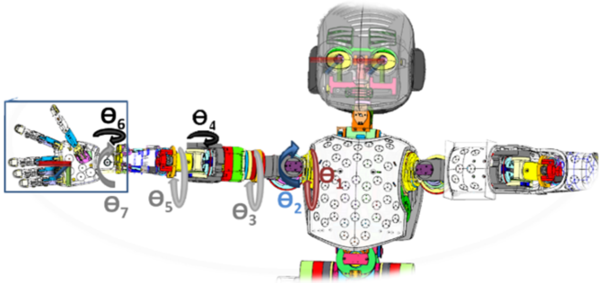


Fig. 1: iCub Computer Aided Design (CAD) model.

Humans are able to learn their own body-schema, which is a process started at a very young age, as stated in [3], by using information from the senses. This continuous learning and adaptation is what allows humans to be able to adapt to different conditions, and producing robots possessing similar behaviours is essential for many areas of robotics where an extended period of autonomous behaviour is required, such as exploration robots in inaccessible areas, rescue robots, social robots and for human-robot cooperation.

Online body-schema adaptation requires the use of machine learning algorithms, which require significant amounts of data, often provided by humans or, in the context of body-schema learning, by performing random movements and obtaining measurements. This may result in acquiring irrelevant data or data which does not improve the models, wasting human, computing and time resources. Active learning is a sub-field of machine learning which aims to reduce the amount of training data required to build a model, with a certain precision. This is done by having the learning algorithm decide which data it wants to label or sample next. A general introduction for this area of research can be found in [4]. Cost-sensitive active learning is a concept also explored in [4]. The main idea is that the learning task may be associated with other costs which do not decrease necessarily if the amount of training data reduces. As an example, robots designed to perform chemical experiments, such as [5], use optimisation algorithms to decide the next experiment to perform based on the learning potential, as well as the monetary costs. Active learning has been used in several works and, empirically, seems to succeed. In the context of robotics and body-schema learning, requiring less data means the robot is able to adapt faster to whichever unpredictable conditions it has to face.

The calibration problem consists of the estimation of a set of parameters associated with the arm's physical characteristics. The parameter estimation (i.e. calibration routine) requires samples obtained from arm movement.

The calibration routine should make use of samples from the position and orientation of the hand, i.e. the hand pose, while knowing the readings of proprioceptive sensors (joint encoders). An active learning approach chooses the best joint configurations in order to reduce the number of samples, as well as reduce the amount of movement needed which would improve time and energy efficiency as a consequence.

The proposed method will be compared with random selection of values for the joints and with a conventional active learning approach, which only aims to reduce the number of samples. The comparison will be made by assessing the ability of the methods to reduce error in the body-schema, the number of samples needed and the amount of movement needed.

II. Related Work

Recent works have succeeded in employing different strategies for body-schema adaptation, such as [6], [7] and [8]. All these works successfully adapt their body models to account for the robot's body errors, but they fail at choosing the most informative samples to do so. Using active learning could improve results and reduce the number of samples.

Active learning methods have been employed and have shown empirical success in multiple areas of robotics, such as [9], [10], [11] and [12]. In the context of body-schema learning, [13] used active learning to estimate a kinematic model of a serial robot and [14] used an intrinsically motivated goal exploration mechanism to learn inverse models in high-dimensional redundant robots. All these works have shown the advantages of using active learning, since less iterations of the learning algorithms were needed to achieve a certain error threshold. However, the main focus was to minimise the number of samples, instead of focusing on minimising the actual effort or movement needed, since the computation time is only part of the learning tasks for robots.

In [15] and [16], the authors proposed criteria for active touch point selection to estimate object shapes which consider both error reduction and exploration costs. These works exploited cost-sensitive active learning, being able to minimise the accumulated path length needed for accurate estimation with low impact on the number of touches necessary. This is similar to what is desired on this work on body schema adaptation. Indeed, the cost of an exploration action is related to the required movement to perform it.

A. Contributions

This work aims to create a calibration routine which can be performed autonomously by the robot, with no human intervention, using active learning for sampling and movement efficiency. The purpose of the calibration routine is to improve the robot's knowledge about its own body-schema. This consists of estimating a set of parameters associated with the physical properties of its body, such as the length of the forearm or the length of the upper arm, called the Denavit-Hartenberg (DH) parameters. The estimation of these parameters will be done using observations of the pose (position and orientation) of the hand, using visual input. This will be tested using the iCub arm, shown in Fig. 1, which has seven rotational joints.

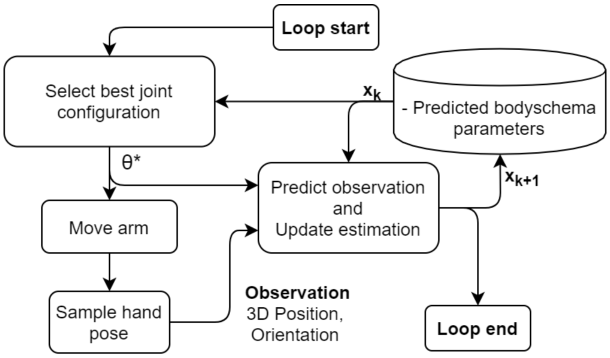

Loop start
Select best joint configuration
x_k
- Predicted bodyschema parameters
θ*
Predict observation and Update estimation
x_{k+1}
Move arm
Sample hand pose
Observation
3D Position, Orientation
Loop end

Fig. 2: Key steps in the structure of the required program to use active learning for joint value selection and observations of the hand pose to estimate the robot's physical parameters, $\boldsymbol{x}$. The subscript k indicates the algorithm's iteration and $\boldsymbol{\theta}^*$ is the selected joint configuration.

Similarly to [15] and [16], we argue that using active learning to reduce the number of samples taken may not be the best approach, since some of the best samples may require unnecessary long movements, increasing execution time and energy spent. The main contribution of this work is a cost-sensitive active learning approach for body-schema learning, which chooses the best joint angles, $\boldsymbol{\theta} = \begin{bmatrix}\theta^{(0)} & \theta^{(1)} & \cdots & \theta^{(n)}\end{bmatrix}$, to sample the hand pose, aiming to reduce the number of samples required and the required movement. The proposed calibration routine is composed of the key steps shown in Fig. 2.

III. Methods

This Section presents the main methods used to build a system capable of performing the steps in Fig. 2.

This system requires a recursive estimator to update the estimation of the DH parameters after each sample, for which it is used an Extended Kalman Filter (EKF), explained in Section III-A. To guarantee sampling and movement efficiency, a cost-sensitive active learning criterion must be defined, which is presented in Sections III-B and III-C. As already mentioned, this thesis proposes acquiring samples of the hand pose for the calibration procedure using visual input. Since the hand may not be detected by the cameras, in Section 3.6 an occlusion model is explained used to predict whether the hand should be visible or not, based on previous attempts, using non-parametric smoothed beta distributions.

A. Extended Kalman Filter

The EKF, explained in detail in [17], allows recursive parameter estimation of systems represented by a nonlinear model, which is the case for the relation between the DH parameters, $\boldsymbol{x}$, and the hand pose, $\boldsymbol{z}$, given by the forward kinematics at a particular joint configuration, $\boldsymbol{\theta}$, $h(\boldsymbol{x},\boldsymbol{\theta})$. For more information regarding the computation of the forward kinematics using the DH parameters, see [18]. A measurement of the hand pose, $\boldsymbol{z}_k$, is modelled by

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k,\boldsymbol{\theta}_k) + \boldsymbol{v}_k, \tag{1}$$

where $\boldsymbol{\theta}_k$ represents the joint encoder values, $\boldsymbol{v}_k$ represents the measurement noise and k is the time-step. In the EKF, it is also considered a non-linear state transition function, f, $\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, u_k) + \boldsymbol{w}_k$, where u_k is a control input of $\boldsymbol{x}_k$ and $\boldsymbol{w}_k$ is the associated process noise. Since in this work $\boldsymbol{x}$ are the DH parameters of the iCub's arm, they are approximately constant in time, only affected by the process noise

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{w}_k. \tag{2}$$

It is assumed the measurement and process noises are Gaussian, uncorrelated and zero-mean, have no cross-correlation, and have known co-variance matrices, $\boldsymbol{R}_k$ and $\boldsymbol{Q}_k$. Therefore,

$$E[\boldsymbol{v}_k\boldsymbol{v}_k^T] = \boldsymbol{R}_k \tag{3}$$

$$E[\boldsymbol{w}_k\boldsymbol{w}_k^T] = \boldsymbol{Q}_k \tag{4}$$

$$E[\boldsymbol{w}_k\boldsymbol{v}_l^T] = 0, \quad \forall \quad k,l. \tag{5}$$

The EKF is divided in two steps: 1) Predict and 2) Update.

1) Predict: This step predicts the state $\boldsymbol{x}$ and the prediction variance $\boldsymbol{P}_{k+1|k}$ at time $k+1$, using only information available at time k. These predictions are given by

$$\hat{\boldsymbol{x}}_{k+1|k} = \hat{\boldsymbol{x}}_{k|k} \tag{6}$$

and

$$\hat{\boldsymbol{P}}_{k+1|k} = \hat{\boldsymbol{P}}_{k|k} + \boldsymbol{Q}_k. \tag{7}$$

2) Update: This step updates the state, $\boldsymbol{x}$, and co-variance, P, using a combination of the prediction and the new observation obtained of the hand pose, $\boldsymbol{z}_k$. They are given by

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_{k+1}[\boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_{k+1|k},\boldsymbol{\theta}_k)] \tag{8}$$

and

$$\boldsymbol{P}_{k+1|k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{K}_{k+1}\boldsymbol{S}_{k+1}\boldsymbol{K}_{k+1}^T, \tag{9}$$

where $\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k}\boldsymbol{H}_{k+1}^T\boldsymbol{S}_{k+1}^{-1}$ is the Kalman gain with $\boldsymbol{S}_{k+1} = \boldsymbol{H}_k\boldsymbol{P}_{k+1|k}\boldsymbol{H}_k^T + \boldsymbol{R}_{k+1}$, and $\boldsymbol{H}$ is the jacobian matrix of the observation function in (1), with respect to $\boldsymbol{x}$, $\boldsymbol{H} = \frac{\partial h}{\partial \boldsymbol{x}}$.

B. Active Learning

An active learning problem is one where the learner has the ability to select its own training data, either from i) a limited pool of unlabelled data (pool-based sampling), ii) by looking at a stream of input data and deciding whether to query or discard it (selective sampling) or iii) by generating its own queries (query synthesis) [4]. This Section will focus on query synthesis, since this work is a regression problem, and this approach has been shown to work when the measurements do not depend on human interpretation [19].

The goal of an active learner is to identify what is the optimal instance to query or action to perform. There are several criteria to achieve this. For instance, expected entropy reduction, expected error reduction, expected variance reduction or expected model change. As shown in [19], the right criterion may depend on the problem itself and if the wrong one is chosen, it may even perform worse than querying randomly. After choosing the appropriate criterion, one must choose an appropriate cost function, $C(\cdot)$, and then the next instance to query, $\boldsymbol{\theta}$, is given by

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmin}} \quad C(\boldsymbol{\theta}). \tag{10}$$

We want to choose, $\boldsymbol{\theta}$, for the next action, which will most reduce the expected error in the estimation model.

1) A-optimality Criterion: The A-optimality criterion is proposed by [13] for active selection of the joint angles, $\boldsymbol{\theta}$. This criterion can be used, since the EKF linearises the observation function around $\hat{\boldsymbol{x}}_k$. It consists of defining the cost function as the expected mean squared error of the robot parameters, $\boldsymbol{x}$. Since the observation noise from (1) is considered to be Gaussian, the cost function is given by the expected trace of the covariance matrix of $\boldsymbol{x}$, given the previous observations, $\boldsymbol{z}_{1:k}$, for joint configurations $\boldsymbol{\theta}_{1:k}$,

$$\begin{aligned} C_0(\boldsymbol{\theta}) &= \mathbb{E}\left[(\hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x})^T(\hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x})|\boldsymbol{z}_{1:k},\boldsymbol{\theta}_{1:k}\right] \\ &\approx \mathbb{E}\left[tr(\boldsymbol{P}_{k+1})|\boldsymbol{z}_{1:k},\boldsymbol{\theta}_{1:k}\right]. \end{aligned} \tag{11}$$

Since the robot should estimate the hand pose using visual input, it needs to find the hand. Some adaptations were made to (11) to avoid the selection of joint configurations where the hand is behind the robot's back or hidden by some other parts of the body. As seen in Fig. 5b, positive values of $\boldsymbol{x}$ for the hand position are not desirable due to the robot $\boldsymbol{x}$ axis pointing backwards. In order to discourage this, a term was added to the cost function from (10) to penalise positive values for $\boldsymbol{x}$ in the predicted position, $\hat{\boldsymbol{p}}$, for a given joint configuration $\boldsymbol{\theta}$. Therefore it was adapted to

$$C(\boldsymbol{\theta}) = \frac{C_0(\boldsymbol{\theta})}{\bar{\eta}} + a \cdot \arctan(b \cdot \hat{\boldsymbol{p}}_x), \tag{12}$$

where $a, b > 0$ are tunable, and $\bar{\eta}$ is the likelihood of the hand being detected by the cameras for the given joint configuration $\boldsymbol{\theta}$. This is further explained in Section III-D.

Optimising the cost function (12) requires a global optimisation algorithm, since it is not easily differentiable, for which it is used the DIRECT algorithm, proposed in [20].

2) Computing $C(\boldsymbol{\theta})$: The information used to compute the co-variance $\boldsymbol{P}$ is shown schematically in Fig. 3. The predict and update steps of the EKF are computed with (7) and (9). This is possible since the prediction and update of the co-variance matrix do not depend on the sample taken, only on the jacobian $\boldsymbol{H} = \frac{\partial h}{\partial x}$ computed at the selected joint configuration, $\boldsymbol{\theta}$, and on the measurement noise co-variance, $\boldsymbol{R}$.

C. Cost-Sensitive Active Learning

In this work, the sample acquisition cost is related to the movement performed by the arm, since (10) selects the next

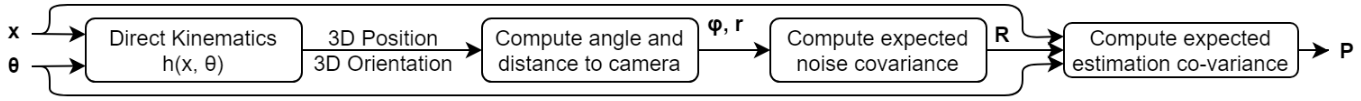


Fig. 3: Steps to compute the expected estimation co-variance, given the current estimation of the DH parameters, $\boldsymbol{x}$, and a joint configuration, $\boldsymbol{\theta}$. ϕ and r represent the angle and distance of the hand to the camera, respectively, according to Fig. 5a.

configuration to where the arm moves to. This work mitigates the amount of movement performed in two separate ways.

1) Unconstrained Optimisation: To penalise the amount of movement in the calibration routine, one can change (10) to

$$\boldsymbol{\theta}_k^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad C(\boldsymbol{\theta}) + \gamma d(\boldsymbol{\theta}, \boldsymbol{\theta}_{k-1}^*), \tag{13}$$

where $d(\boldsymbol{\theta}, \boldsymbol{\theta}_{k-1}^*)$ is a distance, representing the cost associated to moving to the next joint configuration, $\boldsymbol{\theta}$, accounting for the previous one, $\boldsymbol{\theta}_{k-1}^*$, and γ is a parameter which can be changed according to the relative weight intended. Increasing γ will increase the probability of $\boldsymbol{\theta}_k^*$ being closer to $\boldsymbol{\theta}_{k-1}^*$, thus reducing the accumulated movement of the arm.

2) Constrained Optimisation: Reducing the amount of arm movement during the calibration routine can also be done by constraining the optimisation problem. The choice of the next joint configuration is given by

$$\boldsymbol{\theta}_k^* = \underset{\boldsymbol{\theta} \in \left[\boldsymbol{\theta}_{k-1}^* - \boldsymbol{\Delta}, \boldsymbol{\theta}_{k-1}^* + \boldsymbol{\Delta}\right]}{\operatorname{argmin}} \quad C(\boldsymbol{\theta}), \tag{14}$$

where $\boldsymbol{\theta}_{k-1}^*$ is the previous joint configuration and $\boldsymbol{\Delta}$ is a vector of size equal to the number of joints, n. Considering normalised joint values in the interval $[0, 1]$, $\boldsymbol{\Delta}$ is defined as $\boldsymbol{\Delta} = \delta \cdot \boldsymbol{1_n}$, where $\boldsymbol{1_n}$ is a unit vector of size n and δ is a parameter that defines the relative movement every joint can perform around the current configuration. This is stricter than (13), since it is guaranteed that the next joint configuration will be inside the n-dimensional box $\left[\boldsymbol{\theta}_{k-1}^* - \boldsymbol{\Delta}, \boldsymbol{\theta}_{k-1}^* + \boldsymbol{\Delta}\right]$.

D. Non-parametric Occlusion Model

When a new joint configuration is selected, the hand may or may not be visible to the cameras. If the hand is not visible, a sample cannot be obtained to update the EKF. Consequently, the same joint configuration is selected again when solving (10), since the cost function from (11) remains unaltered. Using non-parametric smoothed beta distributions removes this issue, by having the robot retain the knowledge about the successful and unsuccessful attempts at sampling the hand pose and using it to predict the likelihood of a new joint configuration yielding a successful sample.

In [21], a kernel based non parametric approach is used to predict the probability of a successful grasp to extrapolate past data to new unseen features. The same method is applied in this work to predict the likelihood, η_*, of having a visible marker to the cameras in a particular joint configuration, $\boldsymbol{\theta}_*$, given the past k observations, $\boldsymbol{Y} = \{y_1, y_2, \ldots, y_k\}$, in other configurations, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k\}$, where each y_i contains the number of successful and unsuccessful sampling attempts, S_i and U_i, respectively.

The Bayes rule is used to obtain the posterior

$$p(\eta_* | \boldsymbol{\theta}_*, \boldsymbol{\Theta}, \boldsymbol{Y}) \propto p(\boldsymbol{Y} | \eta_*, \boldsymbol{\theta}_*, \boldsymbol{\Theta}) p(\eta_* | \boldsymbol{\theta}_*, \boldsymbol{\Theta}), \tag{15}$$

where the prior $p(\eta_* | \boldsymbol{\theta}_*, \boldsymbol{\Theta})$ is a Beta distribution, with parameters a_0 and b_0, $Be(a_0, b_0)$, and the likelihood term,

$$p(\boldsymbol{Y} | \eta_*, \boldsymbol{\theta}_*, \boldsymbol{\Theta}) = \prod_{i=0}^{k} p(y_i | \eta_*, \boldsymbol{\theta}_*, \boldsymbol{\theta}_i), \tag{16}$$

is the likelihood of the observation, y_i, at each configuration, $\boldsymbol{\theta}_i$, given $\boldsymbol{\theta}_*$ and the success rate, η_*. For each observation, [21] models the likelihood as a binomial distribution

$$p(y_i | \eta_*, \boldsymbol{\theta}_*, \boldsymbol{\theta}_i) = Bin(S_{*i}; \eta_*, S_{*i} + U_{*i}), \tag{17}$$

where $S_{*i} = K(\boldsymbol{\theta}_*, \boldsymbol{\theta}_i) S_i$ and $U_{*i} = K(\boldsymbol{\theta}_*, \boldsymbol{\theta}_i) U_i$ are the number of successful and failed sampling attempts at the joint configuration $\boldsymbol{\theta}_*$, given the attempts at $\boldsymbol{\theta}_i$, propagated by a kernel function, K. The books [22] and [23] provide more information on kernel functions. It was used a squared exponential kernel so that the diffusion process keeps $S_* = S_i$ and $U_* = U_i$ for $\boldsymbol{\theta}_* = \boldsymbol{\theta}_i$ and decreases as $\boldsymbol{\theta}_*$ is farther in the joint space. Using the deduced posterior in [21],

$$\begin{aligned} & p(\eta_* | \boldsymbol{\theta}_*, \boldsymbol{\Theta}, \boldsymbol{Y}) \\ & \qquad = \mathrm{Be}\left(\eta_*; \sum_{i=0}^{k} S_{*i} + a_0, \sum_{i=0}^{k} U_{*i} + b_0\right). \end{aligned} \tag{18}$$

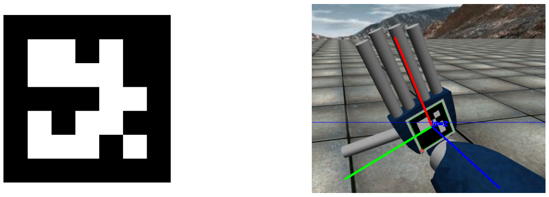
Finally, the predicted sampling success mean probability is

$$\bar{\eta}_* = \frac{\sum_{i=0}^{k} S_{*i} + a_0}{\sum_{i=0}^{k} S_{*i} + a_0 + \sum_{i=0}^{k} U_{*i} + b_0}. \tag{19}$$

This result is used to select joint configurations more likely to have a visible marker in the cost function from (12).

IV. Experimental Setting

This Section describes the implementation and performed experiments. Sampling the hand pose using visual input is done using ArUco markers, shown in Section IV-A. These are placed on the back and palm of the hand, providing a method to detect its position and orientation. Section IV-B gives some details regarding the iCub simulator. Section IV-C establishes the metrics used for comparisons and results. Finally, the proposed experiments are divided in two settings. The first one, explained in Section IV-D, assumes the hand pose can be retrieved, for instance, from an external vision-based algorithm and the real hand pose is directly retrieved on the simulation. The second setting, explained in Section IV-E, is more realistic, since the hand pose is retrieved by using the iCub eyes (cameras) and fiducial markers placed on the hands.



(a) An ArUco marker. (b) ArUco marker pose estimation.

Fig. 4: ArUco marker and pose detection in the iCub simulator. The red, green and blue lines represent the x, y and z axes.

A. ArUco Module

The ArUco module is based on the ArUco library [24], which serves for detection of square fiducial markers (ArUco markers), like the one in Figure 4a. The ArUco module provides functions for marker creation, marker detection and marker pose estimation, in addition to functions to draw the detected markers and their orientation in the input image for visualisation, as shown in Figure 4b.

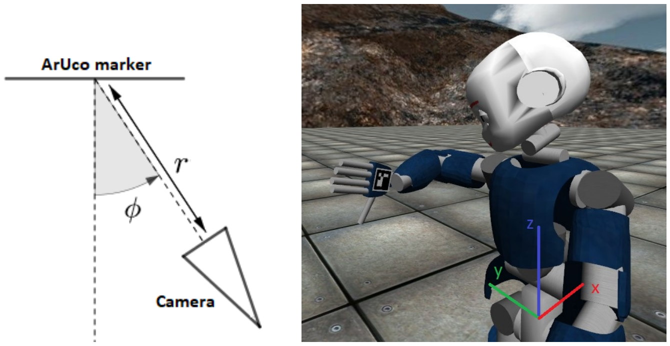
Each ArUco marker encodes a unique ID by a binary code, given by different patterns, and belongs to a set of markers, referred to as dictionary. The set of markers in the dictionaries follow a criterion to maximise inter-marker distance, to avoid possible errors. Only two markers are used to measure the hand pose. One for the palm of the hand and one for the back of the hand, therefore the smallest 4×4 bits predefined dictionary is used, to reduce the inter-marker confusion rate.

As stated in [24], the marker pose with respect to the camera is estimated by minimising the re-projection error of the corners. This measures how close the real image is to the 2-D projection of the 3-D estimate.

The ArUco module, available in OpenCV[1], is responsible for acquiring observations of the hand pose.

1) Measurement Noise Model: In the research field of estimation and tracking, it is crucial to have knowledge about what is the expected error for a given sample and if the expected error is constant for every sample or if it heteroscedastic, meaning it is not constant in time or in space. This allows to give more importance to a sample which has a lesser amount of error expected and vice-versa.

From Section IV-A, it is clear that the accuracy of the pose obtained depends highly on camera calibration parameters and on the amount of pixels the marker occupies in the image. It is of paramount importance to estimate the expected measurement error for each sample. For instance, a marker positioned closer to the camera will yield a more reliable estimate than a marker which is farther away. Having prior knowledge about this allows estimators, in this case the EKF, to know how much to rely on a particular sample and change the measurement co-variance from (3) accordingly.



(a) ArUco marker pose in relation to the camera. (b) ArUco marker placed on the right hand of the iCub simulator.

Fig. 5: ArUco marker detection by the camera in the iCub simulator. The distance from the camera to the marker is represented by r and ϕ represents the angle between the perpendicular direction of the marker and the direction defined by the centre point of the marker and the camera position.

Measurement error and variance in fiducial markers have been studied in various works, such as [25] and [26]. They show how measurement error and variance change with the distance and angle to the camera. This information allowed building a rough predictor of the error, given a particular sample. Both the works achieve similar conclusions:

- The closer to the camera the fiducial marker is, the better is the pose estimation.
- The rotation estimation seems to be worse when the marker is directly facing the camera and the systematic error and variance are lower when the marker is angled 30º to 50º from the camera.

We model the measurement noise co-variance matrix as

$$\boldsymbol{R}_i = \sigma_i^2 \boldsymbol{I}, \tag{20}$$

where σ_i represents the standard deviation of the i-th measurement and $\boldsymbol{I}$ represents the identity matrix. Given the studies in the mentioned works, a very rough approximation of the expected error can be made as a function of camera distance and camera angle to the marker, in order to change σ_i^2 according to the measurement obtained. The mentioned works showed the error seemed to have a quadratic relation with distance and it was lower when the camera is angled 45º from the marker, increasing towards 0º and 90º. Given the distance, r, and angle, ϕ, to the camera, as in Figure 5a, an expression was created to compute a value for σ_i^2

$$\sigma_i^2 = ar^2 + b\left(\phi - 45\right)^2, \tag{21}$$

where a and b are adjustable coefficients, which were selected by trial and error, with an initial guess based on the previously mentioned studies. This is by no means the most accurate error prediction and improving this prediction could improve the final results. Nevertheless, it was essential for improving the performance of the EKF, as will be shown in Section V-C1.

B. iCub Simulator

In order to test the implementation, the open-source simulator for the humanoid robot iCub will be used, presented in

[1] A tutorial of how to use the module is at https://docs.opencv.org/trunk/d5/dae/tutorial_aruco_detection.html.

Resolution	$\boldsymbol{f_x}$	$\boldsymbol{f_y}$	$\boldsymbol{c_x}$	$\boldsymbol{c_y}$
640 × 480	514.681	514.681	320.0	240.0

TABLE I: Image resolution and intrinsic parameters of the cameras of the iCub simulator. These parameters are very similar to the parameters from the actual iCub cameras.

Link	**0**	**1**	**2**	**3**	**4**	**5**	**6**
$\boldsymbol{a}$ [mm]	0	0	-15	15	0	0	62.5
$\boldsymbol{d}$ [mm]	-107.74	0	-152.28	0	-137.4	0	16
$\boldsymbol{\alpha}$ [°]	90	-90	-90	90	90	90	0
$\boldsymbol{o}$ [°]	-90	-90	-105	0	-90	90	180

TABLE II: Actual Denavit-Hartenberg parameters of the iCub right arm in the iCub simulator. These parameters are very similar to the actual DH parameters of the actual iCub robot.

[27]. This simulator was designed to replicate the physics and dynamics of the real robot, as close as possible.

The iCub library[2] provides interfaces to interact with the simulator, either by sending commands or reading information. It allows to send commands to move all joints connecting the rigid bodies, contains proprioceptive information about the joint encoders and it possesses 2 cameras to act as eyes. The properties of the cameras are in Table I.

The Yet Another Robot Platform (YARP) middleware [28] is used to communicate with the robot, which allows building a robot control system as a collection of programs communicating in a peer-to-peer way.

C. Comparison Metrics

In this Section, a few metrics will be defined for performance evaluation of the different methods, regarding prediction errors and the amount of movement done by the arm.

In order to measure prediction errors, both for position and orientation, a set of N points are generated in the joint space, using a uniform distribution, at each iteration. The forward kinematics is computed and, from the resulting matrix, both the predicted 3D position in the Cartesian space, $\hat{\boldsymbol{p}}$, and rotation matrix, $\hat{\boldsymbol{R}}$, are obtained. The respective errors are evaluated at each of these joint configurations. To obtain an overall error estimate of the calibration, the arithmetic mean of both position and orientation errors is computed.

1) Average Position Error: For each of the N configurations, the position error is computed using the euclidean norm and the expression for the average position error results in

$$\frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{p}_i - \hat{\boldsymbol{p}}_i\|. \tag{22}$$

2) Average Orientation Error: For each of the N configurations, the orientation error is computed using,

$$d(\boldsymbol{R}_A, \boldsymbol{R}_B) = \sqrt{\frac{\|logm(\boldsymbol{R}_A^T \boldsymbol{R}_B)\|_F^2}{2}} \cdot \frac{180}{\pi} \quad [°] \ , \tag{23}$$

Position	**Orientation**	
	Unitary axis	**Angle**
2 mm	8%	5°

TABLE III: Standard deviations of the added Gaussian noise to the simulated samples for the geometric simulation setting. After the error is added to the 3 components of the unitary axis, the axis is re-normalised.

where $logm$ is the principal matrix logarithm and $\|\cdot\|_F$ is the Frobenius norm. The average error results in

$$\frac{1}{N}\sum_{i=1}^{N} d(\boldsymbol{R}_i, \hat{\boldsymbol{R}}_i) \quad [°] \ . \tag{24}$$

3) Accumulated Joint Movement: For each iteration of the algorithm, the accumulated joint movement is given by the sum of the movements of each individual joint up to the current iteration. So, at iteration K, the accumulated joint movement is given by

$$\sum_{i=1}^{K} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}\|_1 \quad [°] \ , \tag{25}$$

where $\|\cdot\|_1$ represents the $l1$ norm and $\boldsymbol{\theta}_0$ is the known initial joint configuration.

D. Geometric Simulation Setting

In the iCub Simulator, the DH parameters of the robot are known exactly. This provides a reliable way to sample the hand pose. It removes the error associated with the use of the ArUco markers, so it allows a quicker way to test the initial system implementation composed of the EKF, the cost-sensitive active learning methods, the DIRECT algorithm, connection to the YARP server and use of the iCub interface. It can also serve as validation for the use of the cost-sensitive active learning methods and it can be thought of as a setting where a more reliable way of sampling is available.

1) Sampling the hand pose: As already mentioned, the DH parameters of the iCub robot in the simulation environment are known exactly and are in Table II. In order to obtain measurements of the pose of the hand without relying on its visual sensors, these values can be used to compute the exact pose, by computing the forward kinematics. Random Gaussian noise is added to each sample, after the exact pose is computed, with standard deviations according to Table III.

E. Graphical Simulation Setting

In order to test the system in a more realistic setting, a camera of the iCub Simulator will be used to acquire the hand pose, using the Aruco markers, described in Section IV-A.

The YARP middleware provides interfaces to interact with the environment of the simulator, including adding and moving custom objects with custom textures. By adding planes to the environment with the ArUco markers as textures, one can write a small script to move and rotate the markers to be placed on the palm and on the back of the hand, since the kinematics of the arm are known exactly, as can be seen in Fig. 5b.

[2] The library is available in the git-hub repository https://github.com/robotology/icub-main.

Method	Joint Selection
Random (**R**)	Uniform random selection
Active Learning (**AL**)	Solves (10)
Unconstrained Cost-Sensitive Active Learning (**UCSAL**)	Solves (13)
Constrained Cost-Sensitive Active Learning (**CCSAL**)	Solves (14)

TABLE IV: Summary of the different used joint selection methods.

# of samples per experiment	# of experiments	Uniform distribution width		Initial arm pose
		Linear	Angular	
50	50	46 mm	54°	Random

TABLE V: Conditions of the performed experiments with and without fiducial markers. The uniform distribution width is regarding the random initialisation of the DH parameters.

V. Results

This Section shows the results obtained by the cost-sensitive active learning methods and the comparison with random exploration and with the standard active learning method. The results from Section V-B were obtained with simulated samples and the results from Section V-C were obtained using the ArUco markers for sampling.

A. Experimental Conditions

The calibration routine from Fig. 2 was performed using four different methods for the selection of the joint values of the iCub arm, which are summarised in Table IV. The first method is Random (R), which selects random joint configuration uniformly distributed, which usually serves as a base line for comparison with active learning. The second method is the Active Learning (AL) method, which does not consider movement costs. The third and fourth methods are the cost-sensitive approaches proposed in this work, Unconstrained Cost-Sensitive Active Learning (UCSAL) and Constrained Cost-Sensitive Active Learning (CCSAL).

A random initial joint configuration for the right arm of the robot is set and the algorithm from Fig. 2 executes for all different methods. The main loop of Fig. 2 runs 50 times, which means 50 samples are taken of the hand pose. At each new run, the DH parameters of the robot are initialised with values from a uniform distribution, where the means are the actual values of the DH parameters from Table II and the width of the distribution is according to Table V. In order to obtain solid conclusions, the results displayed are the average of 50 repetitions of each experiment.

B. Geometric Simulation Results

By observing Fig. 6, the first relevant observation is the improved performance of active exploration, comparing to random exploration. R often leads to uninformative samples, resulting in small error decrements. Looking at Fig. 7, it shows the AL method does, roughly, double the movement of R and it is more efficient in the amount of movement performed. For instance, after moving the arm for 2×10^5 deg, the R method has a lower error than the AL method. However, for the same number of iterations (assuming a constant cost for each sample) the AL is more efficient. This is not desirable when trying to sample efficiently since, even if the R method would require more samples, it may have needed less movement to achieve the same results.

When considering the number of samples taken in Fig. 6, both the cost-sensitive methods, UCSAL and CCSAL, can perform very similarly to the AL method, with the selected values for γ and δ, regarding (13) and (14), which were $\gamma = 10^{-5}$ and $\delta = 0.4$, respectively. In the results from Fig. 7, they reveal to be much more advantageous since it took less than half the movement and the same amount of iterations to achieve similar results. When comparing the two cost-sensitive methods, their lines almost overlap in the error plots from Figs. 6 and 7, indicating that both methods allow similar amounts of exploration and exploitation.

C. Graphical Simulation Results

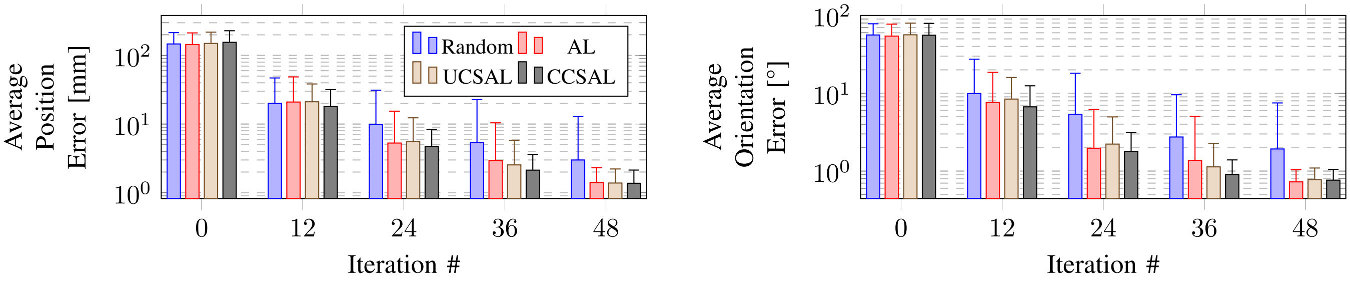
The results in Figs. 8 and 9 were taken using the same methods from Section V-B.

By observing the plots from Fig. 8, there are two major differences, when comparing them to the results without the use of fiducial markers for sampling, in Fig. 6. Firstly, the final errors obtained for the position and orientation errors are much larger. The pose measurement method using fiducial markers introduces a significantly larger amount of noise into the system, resulting in worse estimations. This could be improved by using cameras with a better resolution, which are not available in the iCub simulator. The second difference is regarding the position errors. In Fig. 8a, the random method seems to perform much closer to the active learning methods than in Fig. 6a. This may be, again, due to the limitations imposed by the pose measurement method. When using the fiducial markers, it seems there is not much of a margin to improve after a few samples and the random method catches up rather quickly. Oppositely, in Fig. 8b, the random method is significantly worse at estimating the hand orientation than the active learning methods. This may mean that choosing a joint configuration more likely to have a visible marker in a favourable pose, as explained in Section IV-A1, is more important for estimating orientation than position.

The efficiency of the cost-sensitive methods, UCSAL and CCSAL, seems to be less noticed in Fig. 9 than in Fig. 7, but it is still fairly noticeable regarding the evolution of the orientation errors. The selected values for γ and δ, regarding (13) and (14), were $\gamma = 10^{-5}$ and $\delta = 0.5$, respectively. Both cost-sensitive methods require less movement than the standard active learning method, AL, to achieve the same results in the same number of iterations, meaning they are still advantageous with a non ideal measurement system.

1) Predicting Measurement Noise: This Section shows the impact of the noise model from (21) on the EKF.

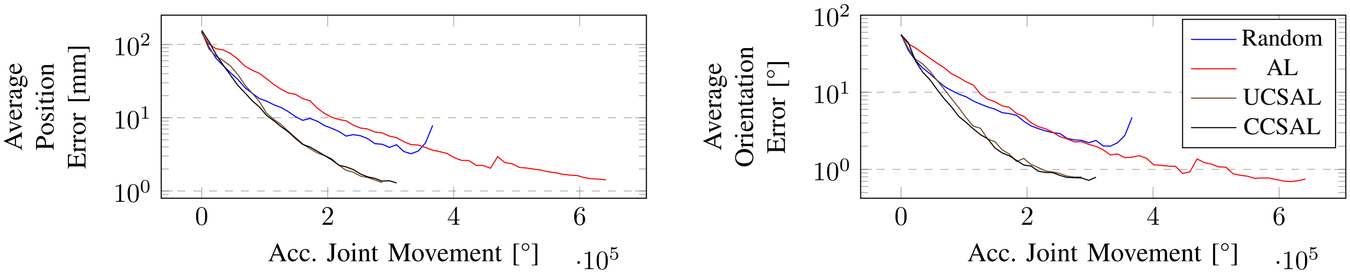
Three different methods are used to test the noise model: Naive EKF, Naive AL, and AL. All three methods use active learning, but they are not cost-sensitive. The results still apply



(a) Mean position error in millimetres with respect to the loop iterations of the calibration routine.

(b) Mean orientation error in degrees with respect to the loop iterations of the calibration routine.

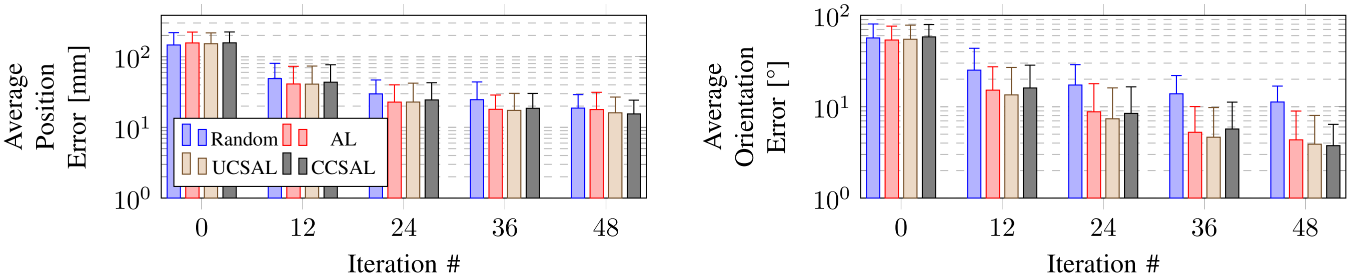
Fig. 6: Geometric simulation results of the different joint selection methods regarding the mean position and orientation errors, with respect to the loop iterations of the calibration routine.



(a) Mean position error in millimetres with respect to the accumulated joint movement during the calibration routine.

(b) Mean orientation error in degrees with respect to the accumulated joint movement during the calibration routine.

Fig. 7: Geometric simulation results of the different joint selection methods regarding the mean position and orientation errors, with respect to the accumulated joint movement during the calibration routine.



(a) Mean position error in millimetres with respect to the loop iterations of the calibration routine.

(b) Mean orientation error in degrees with respect to the loop iterations of the calibration routine.

Fig. 8: Graphical simulation results of the different joint selection methods regarding the mean position and orientation errors, with respect to the loop iterations of the calibration routine.

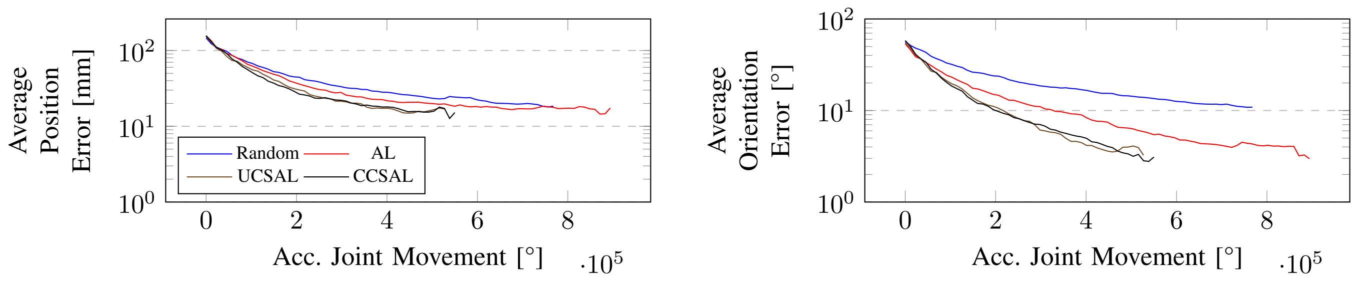
Method	**Measurement Noise Model**	
	EKF	**Joint Value Selection**
Naive EKF	No	No
Naive AL	Yes	No
AL	Yes	Yes

TABLE VI: Summary of the different experiments performed to evaluate the impact of the measurement noise model.

to the general case, since all methods are using the same active learning approach of solving (10).

All the tested methods follow the initial conditions summarised in Table V. The results are an average of 50 runs, where in each run the robot samples the hand pose 50 times. At each run, the DH parameters are initialised from a uniform random distribution, with widths as described in Table V.

In Naive EKF, the measurement noise co-variance matrix is constant through the entirety of the run, $\boldsymbol{R} = \sigma^2 \boldsymbol{I}$. The Naive AL method uses the noise model from (21) to change $\boldsymbol{R}$ only for the update step of the EKF. The AL method uses the noise model from (21) to change $\boldsymbol{R}$ for the update step of the EKF and to predict the best sample in the active learning step. All the methods are summarised in Table VI. The average position



(a) Mean position error in millimetres with respect to the accumulated joint movement during the calibration routine.

(b) Mean orientation error in degrees with respect to the accumulated joint movement during the calibration routine.

Fig. 9: Graphical simulation results of the different joint selection methods regarding the mean position and orientation errors, with respect to the accumulated joint movement during the calibration routine.

and orientation errors are plotted in Fig 10.

In Fig. 10, it is possible to see the differences in performance for both conditions. The Naive EKF approach achieves the least performance, since it is not capable of reducing the impact of measurements more likely to be wrong. The Naive AL approach performs slightly better, but by not predicting the expected error in a sample before selecting the next joint configuration, the EKF converges slower, since the samples obtained are more likely to be less reliable.

2) Marker Occlusion: Some statistics were obtained regarding the number of discarded samples in the 50 iterations of the calibration routine and they are shown in Figure 11. These are discarded due to occlusion of the marker or due to the ArUco module not being able to locate the marker in the camera images. Every time this happens, it means movement is performed, yet no sample is obtained. The statistics obtained are regarding the main results from Figures 8 and 9.

Figure 11 shows the active learning methods tend to suggest fewer arm configurations where the marker is hidden by its own body. This shows the impact of the smooth beta distributions, explained in Section III-D. It is crucial to avoid failed sampling attempts, since, after a few failed attempts, it is able to discourage those and the surrounding joint configurations in the joint selection step. The CCSAL method seems to discard slightly more samples than the other active learning methods. Since the CCSAL method reduces the search bounds to reduce movement, occasionally, the search space may be reduced to one where it is harder to find the markers, while the other methods are allowed to leave those regions quickly. Even though the CCSAL method discards more samples on average, the results from Figure 9 showed it is as efficient as the UCSAL method and even has lower standard deviation values. Reducing search space creates a trade-off between being able to search it more thoroughly, but increasing the chance of being in a region where it is hard to spot the markers.

Improving these results would be difficult without more prior knowledge. The smooth beta distribution could be trained before the calibration routine instead of starting with no prior knowledge, as done in this work, but having too many training data could increase the computation time significantly, since (15) has complexity $\mathcal{O}(n)$ and the DIRECT algorithm must compute it for every point it evaluates.

VI. Conclusions

This work proposed a cost-sensitive active learning approach to estimate the DH parameters of 7 joints of the iCub arm in order to prioritise movement efficiency.

The results show there is an advantage in discouraging or restricting movement during the optimisation stage. It is possible to reduce the movement performed by roughly half and still maintain the iteration wise performance. If movement efficiency is a priority, one can restrict the movement even more, at the cost of more iterations. It is worth mentioning, more iterations does not mean lower time-efficiency, since reducing the amount of time spent moving may make up for the extra computing time. Indeed, it will depend on the computing power and the speed at which the arm moves.

Using ArUco markers provided an insight to a more realistic setting, where the samples obtained are impacted by measurement error. Even in these conditions, the active learning showed robustness, still performing better than random exploration. The cost-sensitive methods were still more efficient than the standard active learning method and maintained a similar performance regarding the number of samples.

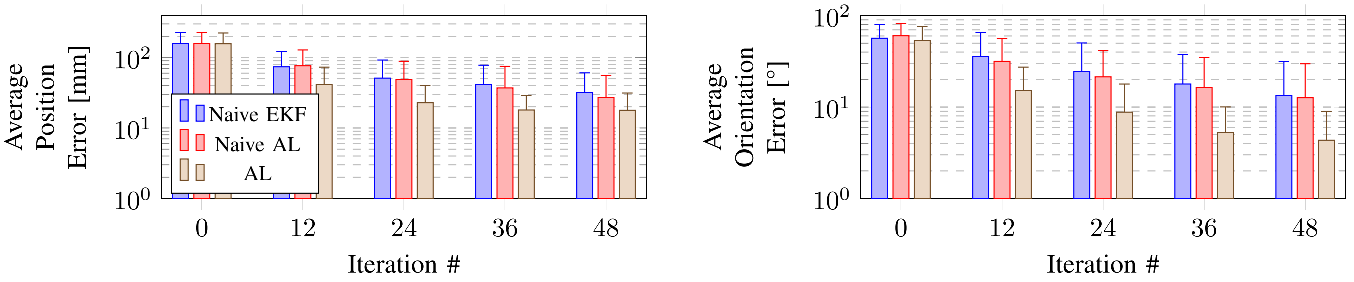
For future work, it would be interesting to proceed to an implementation on the actual iCub robot to evaluate how the methods perform in the real world. It could also be built a better model to predict the observation error to check if it could improve the results significantly.

References

[1] M. S. A. Graziano and M. M. Botvinick, "How the brain represents the body: insights from neurophysiology and psychology," *Common mechanisms in perception and action: Attention and performance XIX*, vol. 19, pp. 136–157, 2002. [Online]. Available: https://www.princeton.edu/~graziano/Papers/Attn_Perf19.pdf

[2] M. Hoffmann, "Body models in humans, animals, and robots," *arXiv preprint arXiv:2010.09325*.

[3] A. v. d. Meer, "Keeping the arm in the limelight: advanced visual control of arm movements in neonates." *European Journal of Paediatric Neurology*, vol. 1, no. 4, pp. 103–8, 1997. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/10728203



(a) Mean position error in millimetres with respect to the loop iterations of the calibration routine.

(b) Mean orientation error in degrees with respect to the loop iterations of the calibration routine.

Fig. 10: Graphical simulation results regarding the use of the proposed noise model from Section IV-A1 in the EKF and active joint selection. Naive EKF does not use the model at any point, while Naive AL uses the model only in the EKF, but not for active joint selection, and AL uses the noise model in both instances.

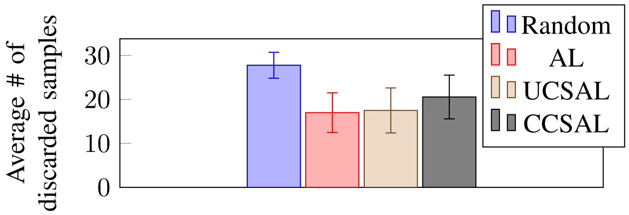


Fig. 11: Average number of discarded samples due to marker occlusion for each method in the graphical simulation results.

[4] B. Settles, "Active Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 6 2012. [Online]. Available: http://www.morganclaypool.com/doi/abs/10.2200/S00429ED1V01Y201207AIM018

[5] B. Burger, P. M. Maffettone, V. V. Gusev, C. M. Aitchison, Y. Bai, X. Wang, X. Li, B. M. Alston, B. Li, R. Clowes, N. Rankin, B. Harris, R. S. Sprick, and A. I. Cooper, "A mobile robotic chemist," *Nature*, vol. 583, no. 7815, pp. 237–241, 2020. [Online]. Available: http://dx.doi.org/10.1038/s41586-020-2442-2

[6] P. Vicente, L. Jamone, and A. Bernardino, "Online body schema adaptation based on internal mental simulation and multisensory feedback," *Frontiers Robotics AI*, vol. 3, no. MAR, 2016.

[7] R. Zenha, P. Vicente, L. Jamone, and A. Bernardino, "Incremental adaptation of a robot body schema based on touch events," *2018 Joint IEEE 8th International Conference on Development and Learning and Epigenetic Robotics, ICDL-EpiRob 2018*, pp. 119–124, 2018.

[8] K. Stepanova, T. Pajdla, and M. Hoffmann, "Robot Self-Calibration Using Multiple Kinematic Chains-A Simulation Study on the iCub Humanoid Robot," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1900–1907, 2019.

[9] U. Martinez-Hernandez, T. J. Dodd, M. H. Evans, T. J. Prescott, and N. F. Lepora, "Active sensorimotor control for tactile exploration," *Robotics and Autonomous Systems*, vol. 87, pp. 15–27, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2016.09.014

[10] U. Martinez-Hernandez, T. J. Dodd, and T. J. Prescott, "Feeling the Shape: Active Exploration Behaviors for Object Recognition with a Robotic Hand," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 12, pp. 2339–2348, 2018.

[11] Q. Lu, M. Van der Merwe, and T. Hermans, "Multi-Fingered Active Grasp Learning," 6 2020. [Online]. Available: http://arxiv.org/abs/2006.05264

[12] A. Ribes, J. Cerquides, Y. Demiris, and R. Lopez de Mantaras, "Active Learning of Object and Body Models with Time Constraints on a Humanoid Robot," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 1, pp. 26–41, 2015.

[13] R. Martinez-Cantin, M. Lopes, and L. Montesano, "Body schema acquisition through active learning," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1860–1866, 2010.

[14] A. Baranes and P.-y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robotics and Autonomous Systems*, vol. 61, no. 1, pp. 49–73, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2012.05.008

[15] T. Matsubara and K. Shibata, "Active tactile exploration with uncertainty and travel cost for fast shape estimation of unknown objects," *Robotics and Autonomous Systems*, vol. 91, pp. 314–326, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2017.01.014

[16] S. Ottenhaus, L. Kaul, N. Vahrenkamp, and T. Asfour, "Active Tactile Exploration Based on Cost-Aware Information Gain Maximization," *International Journal of Humanoid Robotics*, vol. 15, no. 1, pp. 1–21, 2018.

[17] N. Reid, "Estimation," *International Encyclopedia of Statistical Science*, pp. 455–459, 2011.

[18] P. Sanz, *Robotics: Modeling, Planning, and Control*, 2009, vol. 16, no. 4.

[19] J. O'Neil, "An Evaluation of Selection Strategies for Active Learning with Regression," *Dublin Institute of Technology*, 2015. [Online]. Available: http://arrow.dit.ie/scschcomdis

[20] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

[21] L. Montesano and M. Lopes, "Learning grasping affordances from local visual descriptors," *2009 IEEE 8th International Conference on Development and Learning, ICDL 2009*, pp. 1–6, 2009.

[22] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, 2006, vol. 2, no. 3.

[23] B. Schölkopf and A. J. Smola, *Learning with Kernels*. MIT, 2002.

[24] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2014.01.005

[25] D. F. Abawi, J. Bienwald, and R. Dörner, "Accuracy in optical tracking with fiducial markers: An accuracy function for ARToolKit," *ISMAR 2004: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, no. January 2004, pp. 260–261, 2004.

[26] K. Pentenrieder, P. Meier, and G. Klinker, "Analysis of Tracking Accuracy for Single-Camera Square-Marker-Based Tracking," *Proc. Dritter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*, no. August 2016, p. 15, 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.2719&rep=rep1&type=pdf

[27] 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*, 2008.

[28] G. Metta, P. Fitzpatrick, and L. Natale, "YARP: Yet Another Robot Platform," *International Journal on Advanced Robotics Systems*, 2006.