Active Learning of Object and Body Models with Time Constraints on a Humanoid Robot

Arturo Ribes, Jesús Cerquides, Yiannis Demiris, Ramón López de Mántaras

***Abstract*—In this paper we propose an active learning approach applied to a music performance imitation scenario. The humanoid robot iCub listens to a human performance and then incrementally learns to use a virtual musical instrument in order to imitate the given sequence. This is achieved by first learning a model of the instrument, needed to locate where the required sounds are heard in a virtual keyboard layed out in a tactile interface. Then, a model of its body capabilities is also learnt, which serves to establish the likelihood of success of the actions needed to imitate the sequence of sounds and to correct the errors made by the underlying kinematic controller. It also uses self-evaluation stages to provide feedback to the human instructor, which can be used to guide its learning process.**

***Index Terms*—active learning, humanoid robot, music performance imitation, imitation learning, multimodal learning**

I. Introduction

In recent years, active learning has gained a lot of interest from the machine learning community. Active learning is a technique where the learner is capable of interactively querying an oracle for the label of a desired input in order to obtain a labelled training sample [1]. Typically, an oracle is a human with extensive knowledge of the domain at hand. The aim is to reduce the number of needed training samples by careful selection of the questions asked to the oracle.
This is particularly important for the robotics community, as it endows robots with the ability of actively explore their environment, given that robots can take decisions about their own actions.

Also, in the developmental robotics field of research, active learning methods are of paramount importance for the incorporation of intrinsic motivation strategies, to drive the learning process towards situations of increasing complexity and to address the problem of exploration-exploitation trade-off.

In traditional approaches to active learning, the learner would *ask the oracle* the label for a particular query. However, in robotics, it is in fact the robot who *asks the environment*, given that the robot can decide which actions to execute. This ultimately helps in obtaining data which improves the knowledge that the robot has about the environment by maximizing some internal criteria.

With this mindset, interesting research experiments have been conducted in order to solve a variety of tasks taking advantage of active learning methods, like learning control models for high-dimensional and redundant robotic arms [2][3], grounding of relational symbols [4] or learning through human-robot interaction [5], among others.

For this work, we focus in the more general kind of social interaction, where the human supervisor provides an exemplar set of goals for the learner to discover. Specifically, we exemplify this problem in a music imitation scenario, where the robot must learn how to use a virtual keyboard presented in a tactile interface in order to imitate a sequence of musical notes provided by the human, as can be seen in Figure 1. Those sequences are given in the form of an ordered list of musical notes and durations, e.g. $G = \{C^{0.5}, F^{0.5}, D^{0.5}, E^{1}\}$, so the robot needs to learn which actions will ultimately produce such a sequence of sounds.
This problem is particularly interesting for two reasons. First, it does not require the human to be able to use the musical instrument given to the robot, as long as the robot has perceptual means for matching the sounds provided by the human to those produced by the instrument. Second, the robot can discover whether its inability to reproduce the given sequence comes from lack of knowledge about the instrument itself or about its own body motions, that is, the inability to move its body in a timely fashion or with enough precision.

Fig. 1. iCub interacting with the virtual keyboard shown by the Reactable tactile interface. The finger is used to control the virtual object, which is used by our software to know which sound to play.

The time constraint posed by this kind of scenario justifies

Arturo Ribes is with the Learning Systems department at IIIA-CSIC (UAB), Campus de la UAB, E-08193, Bellaterra, Barcelona (Spain) and the Personal Robotics Lab at the Imperial College of London, London, UK. e-mail: aribes@iiia.csic.es.

Jesús Cerquides is at IIIA-CSIC (UAB), Campus de la UAB, Bellaterra, Barcelona (Spain).

Yiannis Demiris is with the Personal Robotics Lab at the Imperial College of London, London, UK.

Ramón López de Mántaras is at IIIA-CSIC (UAB), Campus de la UAB, Bellaterra, Barcelona, Spain.

the use of active learning. This is because there is an interval of time where the robot learner is able to perform inference about which action to execute next, as the acquisition of new samples is governed by an external process. But it also has drawbacks, as the robot may not have enough time to decide on a particularly useful command action nor have time to execute it as planned, so this time constraint induces an interesting trade-off.

Although in this work we approach the problem of learning object properties using an active learning methodology, we also tackle the problem of dealing with black-box control dynamics uncertainties, namely, the uncertainties involved in moving the hand to a desired location with a precise timing of the movement.

For this particular problem, we used an architecture where the residual error of the inverse kinematics algorithm, for which we do not have any control, is fed into a model which learns to make predictions that will be used later to provide corrections to compensate for design or calibration errors.

Our contributions in this paper are as follows. First is to show how a dexterous robot benefits from an active learning strategy to explore an object's properties in order to achieve a sequence of goals proposed by a human supervisor in a perceptual modality. Secondly, how the uncertainties in the robot control algorithms can be modelled incrementally to compensate for design or parametrization errors, avoiding the need to fine-tune those control algorithms.

The robot is also capable of providing a self-evaluation of its own capabilities in terms of how likely it is at succeeding in the imitation task. This feedback is very useful for the human-robot interaction, as it provides a hint about how difficult the task may be for the robot for its current knowledge level.

The rest of the paper is organized as follows. The next section presents related work in the areas of developmental and active learning, and also relevant research for the purpose of utilizing the robot arm control residual error to correct its actions.

Then, we present our proposed architecture for the task at hand and the active learning strategy applied. After the system is described, we provide experimental results that support the use of the presented approach in this kind of problems. Finally, we present our concluding remarks and possible lines of future work.

II. Related work

In the active learning literature, a distinction is made between the sampling strategies and the strategies used to select one of the sampled candidate queries. Sampling refers to the method of obtaining the instances to be queried, where we can use a sequential sampling method by drawing samples from some distribution, or pool-based sampling, where we have a pool of samples, usually fixed, and then the learner ranks the samples according to a selection criteria [1].

While the later is commonly found in video or image retrieval tasks, where the learner has a huge unlabelled corpus of samples, the former seems to fit more into the robotics scenario, where the robot can perform specific actions in order to get samples containing an expected high value of information.

More emphasis is put in the sample selection strategy, that is, which measures can be used to decide if a particular sample, either coming from a predefined pool of instances or sampled from some distribution, is worth asking the oracle for its label. Particularly interesting among the different selection measures are the expected error reduction and the expected variance reduction. Although the latest objective is to have the lowest prediction error possible, both criteria are related in the sense that it is assumed that decreasing the predictive variance one can expect that the expected error will also exhibit a decrease [6].

Until now, we only talked about a learner that tries to actively choose its own learning data, but we did not say anything about the utility or purpose of the acquired knowledge. This is a very important issue, because a robot, particularly a humanoid robot with a high number of degrees of freedom, can be used for a variety of tasks, yet only some of them will be of particular utility.

In that case, social guidance comes in very handy, because provides cues or starting points for the robot learner to bootstrap its own learning process. Human interaction comes in different forms and at different levels of supervision, ranging from providing exemplar goals to achieve, to full-fledged demonstrations using the body of the robot in the *Learning-from-Demonstration* (LfD) scenario. In-between possibilities are intermittent interventions to guide learning if the human supervisor considers it appropriate.

The ability to imitate others starts at early years in development, involving a series of mechanisms linking sensory perceptions with particular motor configurations [7]. Results from neuroscience suggest that perception and action are deeply intertwined and also play a crucial role in the development of the agent [8][9][10].

Our work focuses in the autonomous active exploration of objects using a humanoid robot, while learning also about its own body limitations. We take into consideration the effects of the robot embodiment as a crucial part of the learning process, given that the manipulation capabilities of the robot affect directly the kind of sensory perceptions the robot will receive.

Recently many researchers have put much effort into the development of cognitive architectures that support online learning of object affordances. The concept of affordances, coined originally by J.J. Gibson in [11], makes reference to the relationship between perceptions and actions that an object elicits. In this sense, many reasearch works focus on the sensorimotor learning of those relationships at early stages of development [12][13][14].

Often, the environments that the robots deal with or the complexities in the robot body themselves make the autonomous exploration process cumbersome. In this sense, LfD [15] addresses this problem by providing the system with solutions to a particular problem and allowing the robot to map its internal models to conform to those demonstrations [16][17][18]. However, while this approach is very successful for certain tasks, it usually requires an explicit mapping between the demonstrator and robot body schemas and a definition beforehand of the task to be solved. Active learning strategies have been also successfully applied to LfD in [5].

From the perspective of developmental robotics, the task itself is to learn from the environment a series of skills in an autonomous way [19]. The drive to direct learning towards certain areas of the space of skills comes from what is termed as internal or *intrinsic* motivation [20][21]. It can be seen as a form of active learning, where the robot explores those areas in its sensorimotor space where some measure obtained from its internal models is improved, and not by an extrinsic measure coming from a task definition. [22].

Several works focus mainly in the action part of sensorimotor models, that is, the exploration of behaviour parameters that are expected to provide the robot with data containing high information value [23][24][25]. On the other hand, exploration can be focused in the perception part, also referred to as goal exploration [26][3][22], because it uses goals encoded as specific perceptions to choose actions that drive the system towards obtaining such perceptions.

In the latter case, although usually is the robot who is able to self-generate goals based on previous experience [26][3], there is also space for human-robot interaction to provide candidate goals. In those works, the goals provided by the human are used by the robot in order to bootstrap the goal space [27], i.e. as starting points to generate potentially useful goals, which later can be used to aid or guide the self-generation of other goals when the learning progresses to more mature stages.

Our work belongs to this latter category of problems, where a human subject provides a set of goals the robot should learn to reproduce with proficiency, guiding the exploration of the object it is interacting with. The applied active exploration strategy is similar to the one proposed by [4], where a probabilistic model is exploited in order to provide an estimate of expected reduction in the predictive distribution entropy. However, our models are based on Gaussian Mixture Models (GMM), which naturally support multi-modal and multivariate predictive distributions, and also, in contrast to the classification nature of [4], our problem is a regression one. Similar in its modelization is the work by [28], as they use GMMs to learn the sensorimotor maps. Despite of that, they use an active learning exploration strategy based on the modeling of the prediction error, rather than an information based measurement.

From the perspective of kinematics control, the above mentioned research obviates the errors coming from the action execution comparing the desired with the obtained results, modelling the system as a hole and treating this as system noise. Another approach is the modeling of the residual error after an analytical model has been applied [29].

The other contribution of our work is the integration of a body model in order to provide corrections for the actions of the robot based on the errors between its intentions and the perceived results of its executed actions. To the best of our knowledge there is little research done in this sense, with a particularly similar works being [30][31], where they use a Gaussian Process to model the system noise obtained from an analytical model of the robotic system. The system proposed in [32] introduces a recurrent loop which models the errors of a fixed control element based in the internal motor commands. However, in the control problem studied here, we do not have access to these internal commands.

In our experiments, active exploration is performed with an iCub interacting with a visuo-tactile interactive interface, the Reactable, where a GUI is displayed showing a virtual keyboard and emitting sounds at a rhythm defined by the position of a tactile controlled virtual object. The experimental combination of both systems, iCub and the Reactable, for HRI or more generally, a multi-modal interface, has been explored in active event recognition [33] and in task imitation based on language descriptions [34].

III. Cognitive Architecture

In this section we explain to the reader the proposed architecture, first at the sensorimotor level, and then we continue with the cognitive level, where we detail the kind of models that are involved in the system and how they are learnt.

The imitation of the musical sequence performed by the human requires the robot to learn about two kinds of information: goals and means, that is, the robot must know where to find the musical notes in the keyboard and also judge how to reach those positions from the point of view of its own body capabilities. We can say that the main task of the robot is *to be able to imitate the note sequence*, but also has an implicit subtask, which is, *to be able to judge from a subjective point of view whether or not it can execute the given sequence* due to the time constraints it poses and the motor capabilities of the robot, which may or may not allow it to perform fast enough movements.

Besides the modules involved in perception and action execution, which will be described after introducing the musical interface that we developed to be used in our experiments, we divided the previously mentioned knowledge into two different models: one containing information about how the instrument works, and the other about how the robot body works. A schematic layout of the architecture proposed is depicted in Figure 2. The goals are fed into the model of the instrument to obtain a set of goal actions X_{GOAL} to be executed by the robot controller, which is represented as a black box. After the controller does its internal works, the hand ends up in a position represented as X_{REAL}, which is the one that the instrument uses in order to produce the sound. Both the sound and the end-effector position are fed into the model through learning connections. Also, in order to learn the body capabilities, the desired action X_{GOAL} and its results X_{REAL} are fed into the model of the body, which is used later to provide corrections to the actions the robot wants to execute.

A. Musical interface

Before describing the perceptual system, first we must illustrate the experimental scenario so as to give the reader a picture of how the information flows are interconnected and the chain of events that generate them.

The interaction is produced between the iCub robot and a musical instrument, which is implemented as a virtual keyboard presented in the tactile interface of the Reactable, shown in Figure 3. The underlying software produces musical events with different notes and tempos, determined by the

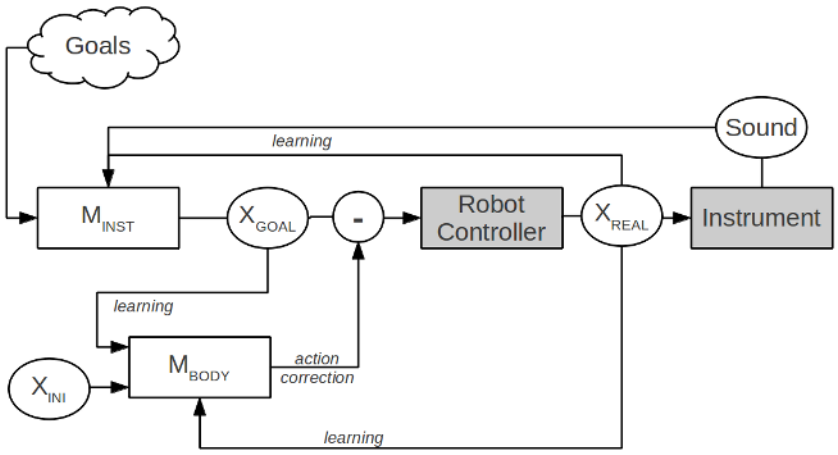

Fig. 2. General schema of the architecture proposed. The white boxes represent the models that are learnt by the robot, while the gray boxes represent closed system where the robot is just an observer. The circles represent variables.

position of the object in the interface. This object is moved by the iCub by dragging its finger over the tactile surface.

The musical events are sound samples from a real musical instrument obtained from an online database[1] and produced at a given tempo. The row where the object is placed defines the duration of the sound, and the note itself is given by the column.

For example, if at the time of a musical event the object is placed in the location depicted by Figure 3, the keyboard will play the note *D#* for the duration corresponding to a quarter note, which depends on the global tempo of the song. After this duration, a new event will be produced and the software will retrieve again where the object is positioned and play the corresponding event accordingly. In Figure 4 we provide a temporal representation of an example note sequence of five pairs note-duration, $S = \{(A, 1), (D, 1), (E, 0.5), (D, 2), (A, 1)\}$.

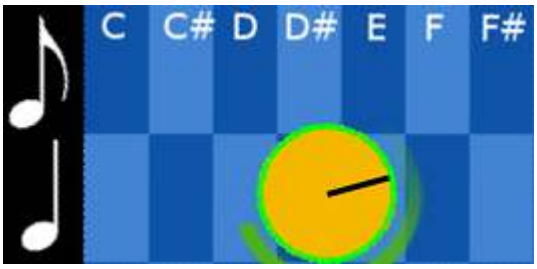

Fig. 3. Virtual keyboard interface for music interaction. The object, shown as a yellow circle, can be moved around by dragging it using the finger. Each cell changes both the note produced and the tempo in which it is emitted.

B. Perception

The perceptual system of the robot is composed of two modules, one for auditive perception and another for proprioception. In terms of auditive information, the robot perceives a vector description of the musical event. As stated before, an event is described by a note and its duration. We cannot use directly the sound wave as is, so first we extract some features using the YAAFE Library [35].

The selected features for sound representation are the *Mel-Frequency Cepstrum Coefficients (MFCC)* due to their successful application in many works concerning instrument and music identification [36][37][38]. MFCCs are computed by means of a non-linear transform of the logarithm of the power spectrum, called *cepstrum*. This non-linear transform maps the spectrum into a more perceptually suitable representation, thus it has been widely used in many research papers. The result is encoded using a Discrete Fourier Transform (DFT), for which only the N first coefficients are retained. In our experiments, we use the first 20 coefficients, which showed enough representation power in our empirical evaluations.

Given that the musical events are single notes, we observed that we can specify the duration of the event by the time between the onset of two consecutive notes, known as the Inter-Onset Interval (IOI). In order to compute this feature, first of all we extract an "onset feature" from the sound sequence again, using the YAAFE library. This feature gives a time-series which contains peaks where the power of the audio signal has an abrupt increase, in our case corresponding to the onset of a note. By detecting the local maxima of this time-series, we obtain the approximated starting time of the event. From that, we can compute the current tempo in beats-per-minute (BPM) or the IOI, which is the temporal feature used in our experiments to establish the duration of the current event. This feature proved to be very useful, as the localisation error of the computed IOI is lower than $15ms$ compared to the usual IOIs used, ranging from $0.5s$ to $2s$.

Having described the timbre and temporal features used in sound perception, we faced two types of issues. First, the MFCCs are sampled at $88Hz$ and computed over overlapping windows of approximately $23ms$ of length. This has the problem of a sample not carrying enough representative power to distinguish between musical notes. In order to aggregate the information of consecutive coefficient vectors, we project

[1]We used guitar and banjo samples from the UK Philharmonia Orchestra website at http://www.philharmonia.co.uk/explore/make_music

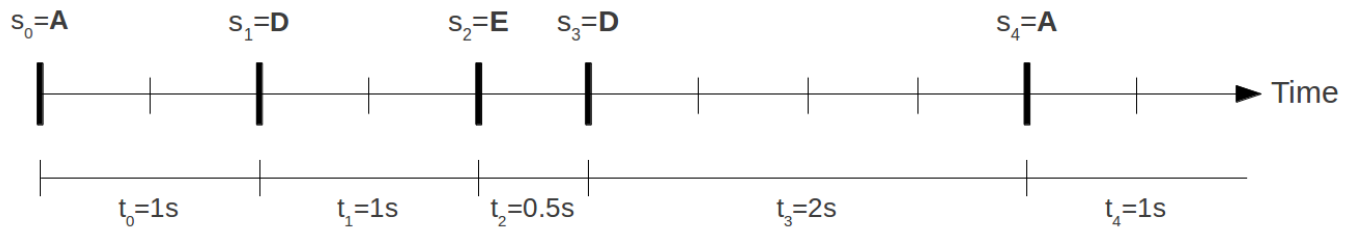

Fig. 4. Temporal representation of a sequence of musical events of the form $S = \{(s_0, t_0), (s_1, t_1), (s_2, t_2), (s_3, t_3), (s_4, t_4)\}$. The note is given by s_n, while the duration is given by t_n.

each of the windows into a GMM to obtain a fixed-sized vector. This approach has been used before in [39] for music classification and is called Bag-of-Features representation, similar to the widely known Bag-of-Words representation in the document retrieval literature.

After we have the BoF vectors for the short-time windows, we perform max-pooling over a longer time window, which captures the temporal variations of the timbre characteristics along the duration of the sound event. The GMM used to project the MFCC was learnt beforehand using the same incremental learning techniques used in our experiments. Having exposed it to a random sequence of musical notes we ended up with a GMM containing 38 components, so the resulting max-pooled BoF vector representation is 38-dimensional. Given that the sliding windows are not aligned with the sound wave and that the notes have different durations, we have some uncertainty in the mapping of the MFCC to the BoF, but this is handled by the probabilistic representation of the instrument model, as will be explained in detail below.

The second issue comes from the fact that we cannot know the duration of an event until the next event occurs, which means that our incremental learning algorithms will be always one step behind the current perceptions. Later it will be shown that, given the incremental nature of our models, we can soon provide estimates of the sound and duration of the event by knowing only the position where the object is located at the time when we predict the event will occur.

The proprioceptive information comes from the iCub encoders and the estimation of the fingertip position in robot-centred coordinates. When moving the hand to a designated position, we confront two sources of uncertainty, one given by the movement itself, as neither the inverse kinematics solver nor the motor actuators reach the desired position, and the other source is the iCub hand being under-actuated and controlled by cables, thus the uncertainty in the fingertip position estimate is quite high. Both sources of uncertainty are handled in the body model, which will be described in detail later.

C. Actions

The iCub robot is placed at a fixed position in front of the tactile interface. The actions that the robot is able to perform are reaching movements by sliding its finger on the surface of the table, which drags the virtual object that is shown in the interface, viewed as a yellow circle in Figure 3. In order for the visual interface to map the position of the robot hand to the object, the robot must calibrate its body coordinates with the local coordinates of the virtual keyboard. This process is done at the beginning of the experiments and consists of placing the virtual object in a set of predefined positions, corresponding to the four corners of the keyboard, and then a series of random positions which render the final calibration more robust. Using a graphical interface to control the hand of the robot in task-space, we direct the hand of the robot to the marked locations, establishing a relationship from the set of obtained task-space coordinates of the robot to the corresponding coordinates in the virtual keyboard.

We define the actions commanded to the robot as *reaching a given position at a desired time*, consisting of a 2-D position vector x^g in the task-space of the hand and a desired movement time t. The orientation, pose and height of the hand is kept constant. This action is given to a modified cartesian controller of the iCub robot, which partitions the whole trajectory into a series of way-points, thus making the motion smoother and safer for the robot. The setup is shown in Figure 1, where we see the iCub controlling the virtual object in the tactile interface of the Reactable.

D. Musical Instrument Model

In order to be able to interact with the musical instrument, the robot needs to acquire a model of it. We decided to use a probabilistic distribution $p(X, S)$, where we define X as the position of the finger in the task-space of the robot and S as the feature representation of a given musical event. That model can be used to answer three kinds of questions:

- *Which sound will I perceive if I touch position **X**?*. This corresponds to the forward model of the instrument, that is, which is the output *S* for a given input *X*.
- *In which positions can I find sound **S**?*. This corresponds to an inverse model of the instrument, that is, which are the inputs X that give as output S. Consider that the result is not a single point but a distribution over inputs, potentially multi-modal, as different keys of the instrument may produce an equivalent sound.
- *How likely is that if I touch position **X** I will perceive sound **S**?*. In this case, we are asking the model to provide estimates of the likelihood of a given *position-sound* pair. This is particularly useful when evaluating candidates for exploration, which may be in areas of relatively high

entropy, thus, potentially leading to false positives when estimating goal positions. It will become clearer when explaining the active learning process.

By conditioning in either one of the variables, we obtain answers for the first two questions from the list above, that is, we may want to know the location distribution for sound s_i using the conditional distribution $p(X|S = s_i)$ or the most likely sound vector to be perceived if we place the finger in position x_i using the other conditional distribution $p(S|X = x_i)$.

Note that the distributions used in this paper are given in a general way and therefore could be implemented by using different models than the ones shown in the rest of this paper. We chose to represent this model using a Gaussian Mixture Model, as it can be incrementally and efficiently learnt from a stream of samples and, more importantly, can represent multi-modal distributions. Furthermore, being a generative model, it can be easily turned into a conditional distribution, so the three questions described in the previous list are parsimoniously represented in one single model.

We can define the instrument model by the joint density $p(X, S)$, captured by a GMM as shown in the following equation:

$$\mathcal{M}^t_{INST} \triangleq p(X, S|\mathcal{D}^t) = \sum_i^N p(X, S|c_i, \mathcal{D}^t)P(c_i|\mathcal{D}^t) \quad (1)$$

This corresponds to the likelihood of the pair (X, S) being observed, as captured by the current state of the model $\mathcal{M}^t_{INST}$ at time t, provided that the model is learnt incrementally using dataset $\mathcal{D}^t$.

In the case of the conditional distributions, for a GMM, the location distribution for a goal sound is implemented by Equation 2 and Equation 4 refers to the most likely sound at a particular location. Note that we drop the term $\mathcal{D}^t$ from these equations for readability purposes.

$$p(X|S = s_i) = \sum_i^N p(X|S = s_i, c_i)P(c_i|S = s_i) \quad (2)$$

$$p(S|X = x_i) = \sum_i^N p(S|X = x_i, c_i)P(c_i|X = x_i) \quad (3)$$

$$\hat{s}(x_i) = \arg\max_s p(S = s|X = x_i) \quad (4)$$

Given the fact that the boundaries of keys are sharp, that is, there is an abrupt change in the class of sound perceived in the boundary of a key of the virtual keyboard, we are bound to have errors by using a GMM to encode the distribution. We could use another family of distributions which might better approximate the kind of regions in this problem, but we did not want to be conditioned by this restriction, therefore resulting in an ad-hoc model which hinders the generalizability of the methods used to other tasks. In fact, we can combine both Equation 2 and Equation 4 to obtain a sample of positions which are highly likely to produce the expected sound.

$$X(s) = \{x_i \sim p(X|S = s) \mid err(s, \hat{s}(x_i)) < \epsilon\} \quad (5)$$

where $err(s, \hat{s}(x_i))$ is just an error function which is thresholded to establish how close the vector representations of sound classes need to be in order to be considered equivalent. In our experiments, this function is the Euclidean distance between both vectors.

$$err(s_a, s_b) = \| s_a - s_b \| \quad (6)$$

E. Body Model

For any movement we command the robot to do, we are likely to be affected by the pitfalls of robot control, that is, uncertainties that we cannot or we do not want to control, like kinematic solvers, PID controllers and mechanical properties of the robot system itself. This problems cause that the final position we would like to reach is, though very close, not the same as we commanded. We have to deal with this uncertainty in our system, and we do so by learning a probabilistic model which learns these uncertainties and enables the system to reason using this information.

Let us remember the definition of an action command, which is to *move from an initial position x^i to a goal position x^g in t seconds*. In practice, after the execution of an action, given that the hand reaching controller is not perfect, there is an error between the goal position x^g and the actual reached position x^r, as well as between the commanded time t and the actual time $\hat{t}$ needed by the robot to reach x^r. With that definition at hand, we would like to model the uncertainty in both variables, goal position and reaching time errors, that is the difference between the commanded and the actual values of these two variables. These differences are captured in the variables $\Delta X^g = x^r - x^g$ and $\Delta t = \hat{t} - t$.

We thus decide to represent the body capabilities as the distribution of the error variables $\Delta X^g, \Delta t$, accounting for the results of an action, and the action parameters X^i, X^g, T. This is defined as the model $\mathcal{M}^t_{BODY}$ learnt up to time t from dataset $\mathcal{D}^t$.

$$\mathcal{M}^t_{BODY} \triangleq p(\Delta X^g, \Delta t, X^i, X^g, T|\mathcal{D}^t) \quad (7)$$

As happens with the instrument model, this formulation is rather general and, thus, can be represented with different probabilistic models. However, for consistency with the model used in our experiments, we provide the equations assuming that we used a GMM.
Then we can infer, given the current position x^i, a goal position x^g and a reaching time t, which is the distribution of expected end positions by sampling from

$$\begin{aligned} \hat{x^g} &= x^g + \Delta x^g \\ \Delta x^g &\sim p(\Delta X^g|X^i = x^i, X^g = x^g, T = t) = \\ &\sum_m^N p(\Delta X^g|c_m)P(c_m|X^i = x^i, X^g = x^g, T = t) \end{aligned} \quad (8)$$

Conversely, the expected reaching time is obtained as a mixture of univariate normal distributions using the following equation:

$$p(\Delta T|X^i = x^i, X^g = x^g, T = t) \tag{9}$$

Such a distribution is used to compute the probability of reaching the destination before the next event occurs. For example, let us assume that the next event is due to happen in $T_{max} = 1s$, and the action is issued with a reaching time of $t = 0.7s$. That leaves us with an error margin of $0.3s$. Now we can use the *cumulative distribution function (CDF)* of the mixture of univariate normal distributions obtained from Equation 11 to check the probability that the error in reaching time is less than $0.3s$ as a measure of confidence of the robot reaching on time the required location.

In our experiments, this model is implemented using a GMM, so the equations for the goal position error and time error are as follows:

$$p(\Delta X^g|X^i = x^i, X^g = x^g, T = t) = \sum_m^N p(\Delta X^g|c_m)P(c_m|X^i = x^i, X^g = x^g, T = t) \tag{10}$$

$$p(\Delta T|X^i = x^i, X^g = x^g, T = t) = \sum_m^N p(\Delta T|c_m)P(c_m|X^i = x^i, X^g = x^g, T = t) \tag{11}$$

F. Active learning strategy

In a typical active learning setup for classification, the learner is most concerned about choosing a good learning sample, so it has to decide using one or different strategies which is the input vector and then ask the oracle to provide a label for it.

This is particularly suited in applications where we do not have a good dataset of the environment, usually because the labelling costs are very high. In that case, it is beneficial to invest some time in crafting a good question so the learner gets a higher return in terms of the information contained in the resulting training sample.

Many works from active learning literature use measures of intrinsic motivation based on the uncertainty of the model [40] or its prediction error [41]. However, in developmental robotics those measures are often dismissed because they make assumptions about the learnability of the underlying function, sometimes leading to pathological behaviours like focusing on unlearnable parts of the state space or exploring areas governed by uncontrollable randomness.

For this reason, other measures based on the gradient or *progress* of this quantities are proposed [20] [42]. This corresponds to a decrease of the variance or learning progress.

Our approach is based on an information theoretic measure for intrinsic motivation. We consider interesting learning about areas which may result in a decrease of the predictive entropy. Thus, the robot is endowed with a drive to explore positions where it expects that will lower its predictive entropy after learning about them. In contrast to approaches where the error is considered, either empirical [3] or expected [42], we consider that, in problems where the distribution of outcomes may be multi-modal, an approach based in uncertainty reduction is more suitable.

The approach presented in [4] also use the expected predictive entropy reduction. However, the authors use a model which relies on Gaussian Processes, so their posterior predictive distributions are inherently uni-modal. Our approach, by making use of a GMM, overcomes this limitation as this kind of model can represent multi-modal distributions.

Predictive entropy, as defined by the following equation:

$$H(X|S = s) = \int_{\mathbb{R}^D} P(X|S = s) log(P(X|S = s))\,\mathrm{d}x \tag{12}$$

is a function related to the variance of the distribution, although more suitable for multi-dimensional and multi-modal predictive distributions like the one given by Equation 2. Thus, a reduction in entropy can be seen that as a reduction in variance.

Given that in our experiments our models are GMM, this poses a problem, as there is no closed form for computing its entropy without making some assumptions. Therefore, we approximate it by using an upper-bound of the entropy, which consists of a weighted sum of the entropies of the individual Gaussian components [43]. This upper-bound is formally defined by the following equation:

$$H(X|S = s) \leq \sum_i^N \omega_i \cdot (-log\,\omega_i + \frac{1}{2} log\big((2\pi e)^D |\mathbf{C}_i|\big) \tag{13}$$

where D is the dimensionality of the distribution, N is the number of components in the mixture model, $|\mathbf{C}_i|$ is the determinant of the covariance matrix of component i and $\omega_i = P(X|S = s, c_i), \forall i \in 1..N$ is the weight of the component i, equivalent to the probability that a sound perception s is matched to mixture component i. This measure is very fast to compute, as most of the terms can be cached to speed up computation.

In order to overcome the complexity of computing the determinants of the covariance matrices, we exploit the fact that each training point only will update very few model components, so we maintain a cache of inverse matrices and determinants to accelerate computations.

For a distribution where there is no significant overlap between the mixture components, the real entropy is very close to its upper bound.

We consider the task to be dependent on a given set of goal sounds G to be discovered, defined as a subset of the possible sounds $\mathcal{S}$ that can be produced by the musical instrument, i.e. $G \subset \mathcal{S}$. Algorithm 1 provides the steps to retrieve a candidate position, given the active learning strategy to follow, the set of goals G and the current model $\mathcal{M}^t_{INST}$, used to extract the sampling distributions.

First, we obtain the entropy of the current predictive distribution $H(X|S = G, \mathcal{M}^t_{INST})$. Then, we sample a set

Algorithm 1 Retrieve a candidate position

Input: strategy = {RAND, PRIOR, POST}, G, $\mathcal{M}^t_{INST}$
$\mathcal{H}^t \leftarrow H(X|S = G, \mathcal{M}^t_{INST})$
$weights \leftarrow []$
if strategy = PRIOR **then**
    $candidates \leftarrow sampleFrom(P(X))$
else
    $candidates \sim sampleFrom(P(X|S = G))$
for all x_i **in** $candidates$ **do**
    $\mathcal{M}^{t+1}_{INST} \leftarrow update(\mathcal{M}^t_{INST}, \{x_i, \hat{s}(x_i)\})$
    $weights(i) \leftarrow \mathcal{H}^t - H(X|S = G, \mathcal{M}^{t+1}_{INST})$
$c \leftarrow SoftMax(weights)$
Send action based on x_c

of position candidates $\mathbf{x}^c$, depending on the strategy used, either from the prior distribution over positions $P(X)$ or from the distribution of positions conditioned on the goals we must discover $P(X|S = G)$. Once we have the set of candidates, we have to compute, for each of the candidates, the expected decrease in predictive entropy by simulating the possible outcomes of executing an action based in the candidate location. We approximate it by taking the most likely sound for the candidate position being evaluated, thus, we have that:

$$H(X|S = G, x_i, \mathcal{M}^t_{INST}) = H(X|S = G, update(\mathcal{M}^t_{INST}, \{x^c_i, \hat{s}(x^c_i)\})) \quad (14)$$

where $update(\mathcal{M}_{INST}, \{x, s\})$ is an operation that incorporates the data sample $\{x, s\}$ into the model $\mathcal{M}_{INST}$, returning the updated model. $\hat{s}(x^c_i)$ computes the most likely sound for a given position, and is obtained from Equation 4. The expected decrease in predictive entropy for each of the candidates, is used as a weight in order to stochastically select a candidate by means of the softmax function [44]. In Figure 5 there is an schematic depiction of the whole process.

Then we compared a baseline method with two alternative active learning methods. The baseline method consists of just taking a random sample from Equation 2, which at the beginning amounts to a uninformative flat prior distribution, and constructing an action based on the sampled position. The active learning methods are differenced by the way they sample the potential candidates. One is to sample, as in the baseline method, from the distribution over actions conditioned on the goal perceptions we desire to obtain. The other is to sample directly from the prior and let the weights based on entropy reduction decide which candidate to take.

IV. Experimental Results

In this section, we present the experimental setup used to answer two main research questions, namely, the impact of applying active learning strategies in the number of needed training samples and how to improve the control of the robot by means of learning an action correction model. Regarding the first question, we are interested in situations where the robot uses part of the time between the acquisition of two consecutive data samples to infer a potentially good learning candidate. Particularly, we focus in applying an active learning strategy which helps in reducing the amount of training data needed by the robot to reach a desired level of competence. Our results showed a significant improvement of the presented active learning strategy when compared with a random selection strategy.

The other research question deals with the physical nature of the studied system. Given the complexity inherent in solving the kinematic equations used to control the robot hand, it is very difficult to tune the controllers in order to reach the desired locations. This causes location errors that potentially hinder the actual performance of the robot. A machine learning methodology is applied to overcome these limitations and its impact is assessed in the experiments proposed here, showing that for complex predictive distributions where the choice of action is not clear, taking advantage of a model about how the its body behaves provides a benefit to the robot.

The cognitive architecture described above was implemented in the iCub platform, a 53 degrees of freedom (DoF) humanoid robot [45], using its upper torso and only one of the arms. The motor control was done using a cartesian controller, which given an action specified as desired end-effector position and execution time, internally solves the inverse kinematics problem [46]. Given the intricacies of motor control over a flat surface, we used a modified finger sliding controller built on top of the cartesian controller for smoother and precise control of the fingertip.[2]

The robot frame of reference was calibrated to the Reactable, a visuo-tactile interactive interface, in order to map the coordinates of the robot end-effector to the coordinates received from the tactile interface.

Due to the inherent difficulties in calibration using vision, we decided to directly calibrate the hand of the robot to the local system of reference of the experimental interface shown in the Reactable screen, so we ended up with one calibration matrix instead of two. In any case, there were calibration errors which our system learnt effectively and minimized their negative effects.

The software was implemented using the YARP middleware [47] for tasks related to the iCub control and sensor data acquisition. ROS [48] was used for the learning related tasks and as integration tool for all the modules. Experiments shown in Figure 6 were executed in the iCub Simulator [49] in order to experiment with the parameters of the model and tune the algorithms.

A. Learning the Instrument Model

First of all we evaluated how the robot finds the different notes required to imitate the sequence given by the human. We compared the learning performance of the active learning strategy with a baseline, which is defined as reaching a random location the current model expects to contain a goal sound.

[2]Thanks to Ugo Pattacini for providing the base code for the sliding controller.

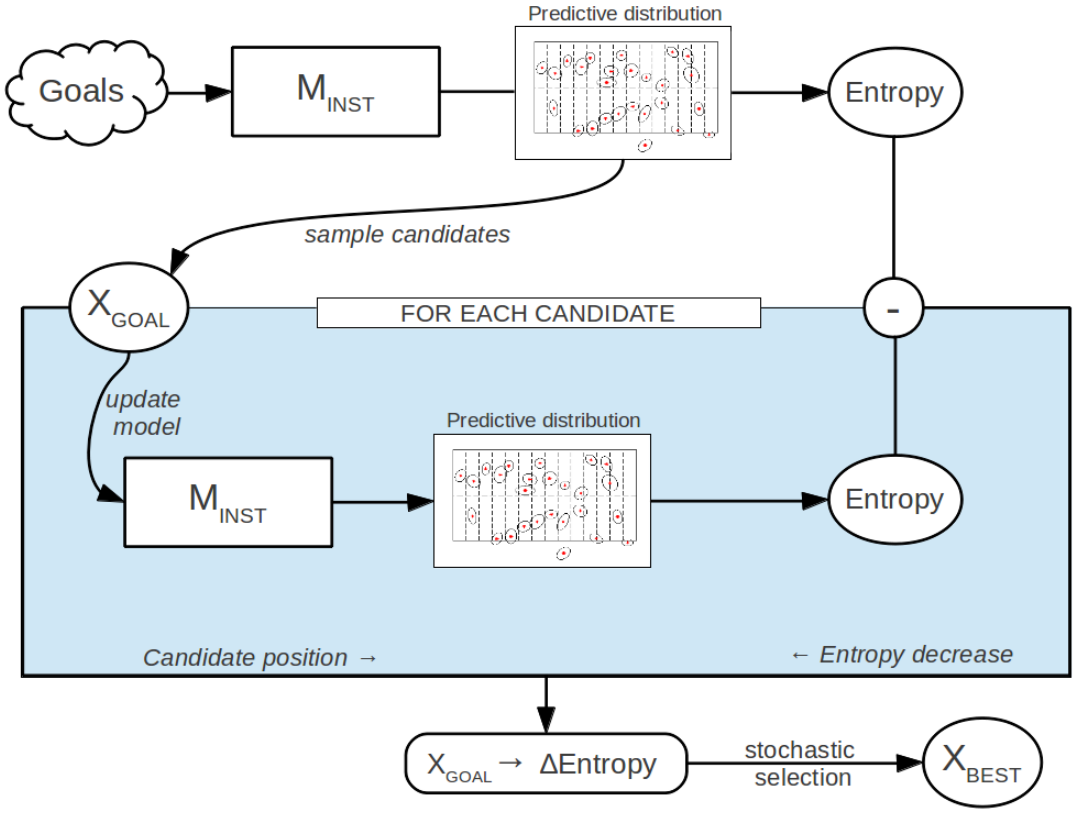

Fig. 5. Schematic of the active learning strategy. Using the predictive distribution extracted from the current model $\mathcal{M}^t_{INST}$, we compute the predictive entropy. Then, we sample a set of position candidates to explore. Each candidate is used to simulate an update of the model, so we can obtain the new predictive distribution and compute its entropy, which is used to give a score to each candidate according to the decrease in predictive entropy. A candidate is sampled stochastically according to these scores.

Due to the randomness inherent in the active learning method, we performed a series of experiments in order to track the performance over the whole learning process. The evaluation measure used is the time, specified in terms of number of samples needed, to reach a desired average precision level. Then we performed a non-parametric hypothesis test in order to assess the statistical significance of our experimental results. In our experiments, precision is defined as the proportion of predicted locations that are expected to produce a specific goal sound. We computed it by first sampling a set of 1000 locations from the model, and then checking whether or not these locations were inside the correct region producing the sound being evaluated.

Given that the model was learnt with very few exemplars and with little prior knowledge, we observed that when computing the sampling distribution for obtaining candidate positions, we faced a problem of exaggerated differences in the probabilities of some components generating the data, most likely caused by the high-dimensionality of the perception vector description compared with the size of the dataset $\mathcal{D}^t$ used until time t and the components having used very few training exemplars to learn their parameters.

In order to normalize the mixture component likelihoods to obtain a distribution vector to sample from, we used a transformation based in the one proposed in [50], which works by mapping the normalized likelihood values to the range $[10^{-K}, 1]$, where K is a prefixed value which basically states the maximum difference in orders of magnitude between the highest and lowest confidence measures the model can provide. In our experiments this parameter was set to 10 after an empirical evaluation. However, we did not observe a high sensitivity on this parameter unless high values were chosen, e.g. $K > 50$.
This was done by first transforming the likelihood values to a logarithmic scale, then linearly mapping the lowest and highest value to a range of $[-K, 0]$. After that, we just mapped back and normalized the result.
As described in [50], this mapping does not exaggerate the relative differences in belief, nor does alter the relative ordering in mixture component likelihoods.

Our proposed entropy-based active learning uses two sampling strategies which we also compared in our experiments. In order to get the sample of candidates to be explored, we could sample from the distribution conditioned on the set of n goals G, using the following formula:

$$x_i \sim p(X|S = G)\ s.t.\ G = \{g_0, \ldots, g_{N-1}\} \qquad (15)$$

or by uniform sampling from the prior distribution over

positions:

$$x_i \sim p(X) \tag{16}$$

This prior distribution is defined as a uniform distribution over the range of positions that the robot is able to reach safely, so it effectively corresponds to the robot arm working space. We show the results in Figure 6, with the corresponding histograms for the distribution of the number of samples needed to reach a precision of 60%. It can be clearly seen that the best strategy is to use active learning sampling from the prior distribution, as the distribution conditioned on the goals offers a bias, thus not very suitable particularly in early stages of the learning process.

In order to observe how the model changes over time as new regions of the instrument were explored, in Figures 7 and 8, we show two different examples of exploration sequences. The one in Figure 7 corresponds to the unimodal case, while the one in Figure 8 refers to the multimodal case. The scatter plots show, for each of the four goal sounds given to the robot, a sample of positions obtained from the model at three different stages of learning. These stages correspond to the robot having explored 20, 80 and 200 locations, respectively. It has to be noted that the shown examples are selected for illustrative purposes, and that the variability of the obtained models in terms of the number of training samples, that are needed to have a desired level of performance, can be seen in Figure 6. The multimodal case corresponds to a more difficult problem, where the goal sounds can be found in two separate regions, thus making the predictive distribution inherently multimodal. The specifics of this problem are explained in Section IV-E.
It can be seen that at the early stages of learning, some of the goals remained undiscovered, and the ones discovered correspond to broad regions which expand beyond the sharp boundaries of the virtual keys, while more mature stages show that all goals have already been discovered, thus corresponding to *narrowing down* the boundaries of the discovered regions.

We also provide results for the precision estimated by the model, that is, we use the model to judge whether or not the expected perception belongs to the goal we desire to obtain. The early stages were found to be over confident, due to poor boundary definitions, which is normal given the Gaussian nature of the underlying model. However, later stages proved more accurate in judging whether a point sampled from the posterior distribution over positions given the specific goals will produce the expected perception.

B. Learning the Body Model

We also evaluated the performance of the body model. In this case, we allowed the robot to perform reaching movements associated with the goals that it needed to imitate. This was done after the instrument model was learnt, so as to guarantee that the robot was confident enough to retrieve valid candidate positions.

After some data was acquired and a body model learnt, we evaluated the accuracy of the error predictions made by the model by comparing them with a series of test reaching movements.

For each instrument model learnt from the evaluation of Section IV-A, we obtained a body model by performing series of imitative actions as described in the next experiment. Then we performed the evaluation of the corrections using these pairs of instrument and body models, obtaining the datasets needed to empirically show how the spatial reaching error is accurately predicted by the corresponding body model.

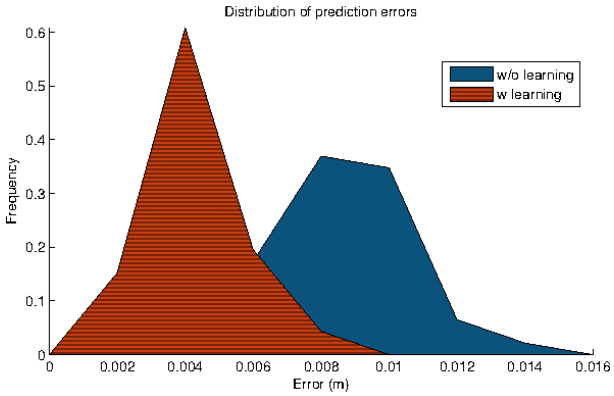

Fig. 9. Distribution of reaching errors with and without learning a body model. It can be clearly seen that the body model predictions are accurate enough to be used as corrections to enhance the performance of the reaching actions.

In Figure 9 it can be clearly seen that the body model predictions are accurate enough to be used as corrections in order to alleviate the effects of the calibration error in the inverse kinematics controller of the robot. Almost all the predictions kept the reaching error below $5mm$, which is the lowest bound our robot controller used to consider a reaching movement finished, so any improvement on that is considered as pure chance. However, with no learning, a lot of errors were above $1cm$, so high that in many occasions the robot ends up out of the region that produces the desired perception.

C. Imitation of the sequence

After learning both models, the robot was ready to try to imitate the given sound sequence. We divided the sequence in series of pairwise goal sounds. For example, if the goal sound sequence was:

$$G = \{C^{0.5}, F^{0.5}, D^{0.5}, E^{1}\}$$

provided that C, F, D and E are the musical notes and 0.5 and 1 are the tempos, expressed in seconds, we obtained the following pairwise goal sequence:

$$G^{PW} = \{(C^{0.5}, F^{0.5}), (F^{0.5}, D^{0.5}), (D^{0.5}, E^{1}), (E^{1}, C^{0.5})\}$$

By using the model of the instrument to obtain the positions and times from this sequence, we transformed the list G^{PW} into a list of action commands. However, the process of

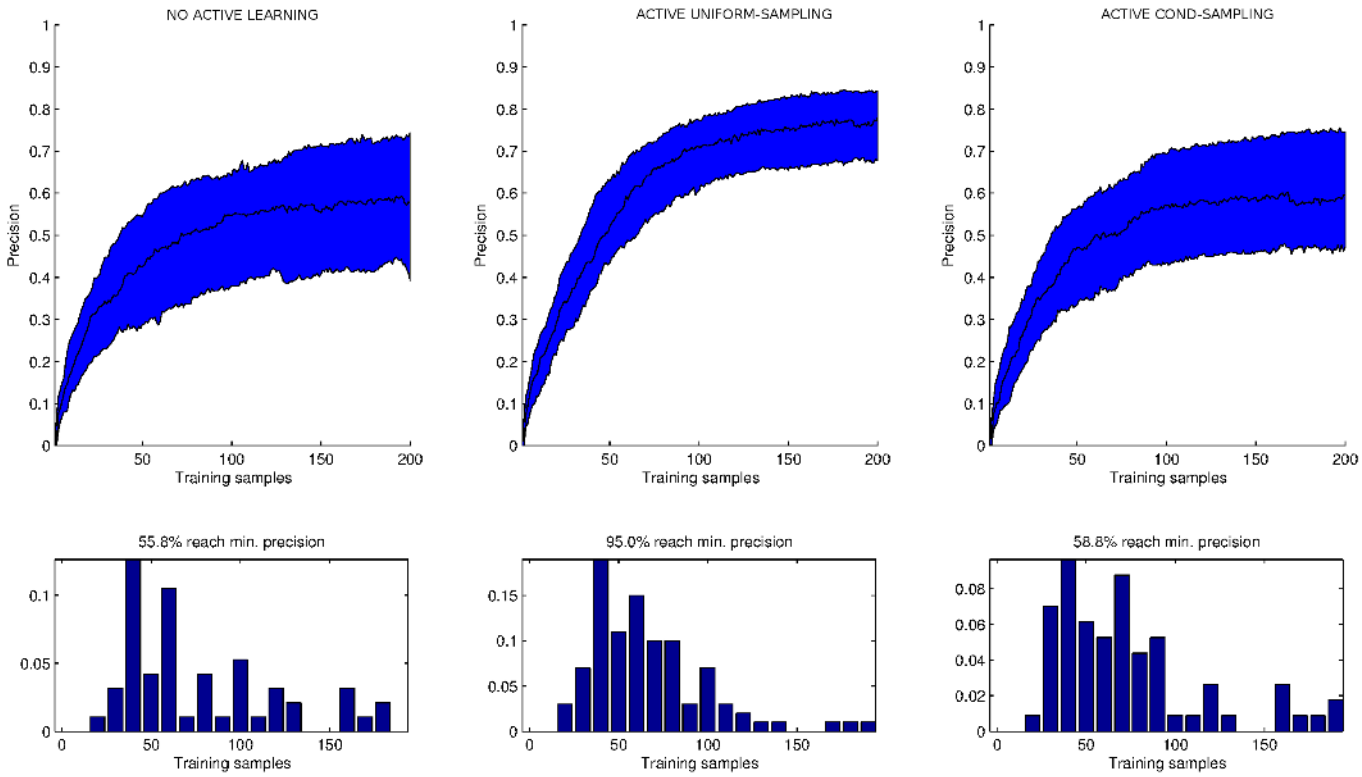

Fig. 6. Results for the three strategies aplied to discover goal sound regions. NO ACTIVE LEARNING consists in taking a random sample from the distribution conditioned on the goals. The active learning strategies are UNIFORM-SAMPLING, which takes the sample candidates from the prior distribution on locations, and COND-SAMPLING, which uses the distribution conditioned on the goals to obtain the candidates.

imitating the sequence was slightly different. We assumed that if the robot failed to reach a desired goal position, it had to start again from the beginning. This obviously induced a bias on the first goal having a lot of trials, while the latest one was only tried after the previous ones have been correctly reached, but the resulting precision probabilities were accordingly normalized taking into account this issue.

As explained in Section III-B, the perception of the robot in terms of the sound was one step behind, meaning that when the sound for event evt_t started to be played, the robot actually perceived the sound for event evt_{t-1}.
For this reason, it used the current position of the hand at the time of the new event to infer the sound $\hat{s}$ that was expected to be playing, using Equation 4, and the duration of that sound in order to know the time $\hat{t}$ for the next event.
Then the robot checked if the expected sound $\hat{s}$ was any of the goals in G. This was done by choosing the goal sound that minimized a matching error function $err(\hat{s}, g_i)$:

$$\hat{i} = \arg\min_i err(\hat{s}, g_i) \; s.t. \; i \in 1 \ldots \#G$$

The error function used in our experiments is the same as Equation 6. Only matches below an error threshold were considered good, so if $err(\hat{s}, g_{\hat{i}}) < \epsilon$, the robot assumed that the current sound was indeed the goal $g_{\hat{i}}$. If not, the robot assumed it was in a wrong location and sent an action to go back to the first goal G_1.
Having identified the current goal, the robot extracted the next goal sound from the corresponding pairwise goal $G^{PW}_{\hat{i}}$.

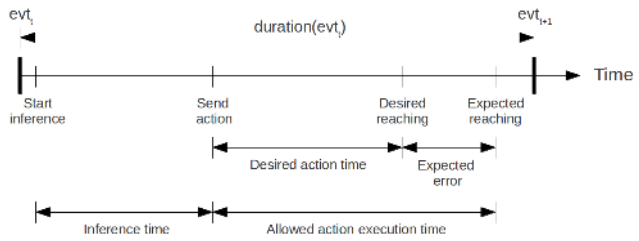

Fig. 10. Action inference process carried on by the robot. The robot considers a fixed action execution time. This, added to the expected temporal error of considered actions, gives the maximum time allowed for action execution. The remaining time is used for inference of the best action to execute.

Now the task was to find, for the next goal sound, a good candidate action, defined as a position and a reaching time. The process is illustrated in Figure 10. After computing the maximum remaining time T_{max} for the current sound being played, we had to allocate two time segments, one for inferring the action to be executed and another for actually executing the selected action. However, the real action execution time needs to take into consideration the uncertainty in the hand controller, so we also accounted for this temporal error.

Let us say, for example, that $T_{max} = 1s$, and that we set the time for action execution to $t_c = 0.7s$. It means that the robot only had $0.3s$ to spend on inferring the action and also to account for the temporal error that such action may have. An example can be seen in Figure 11, were the resulting action

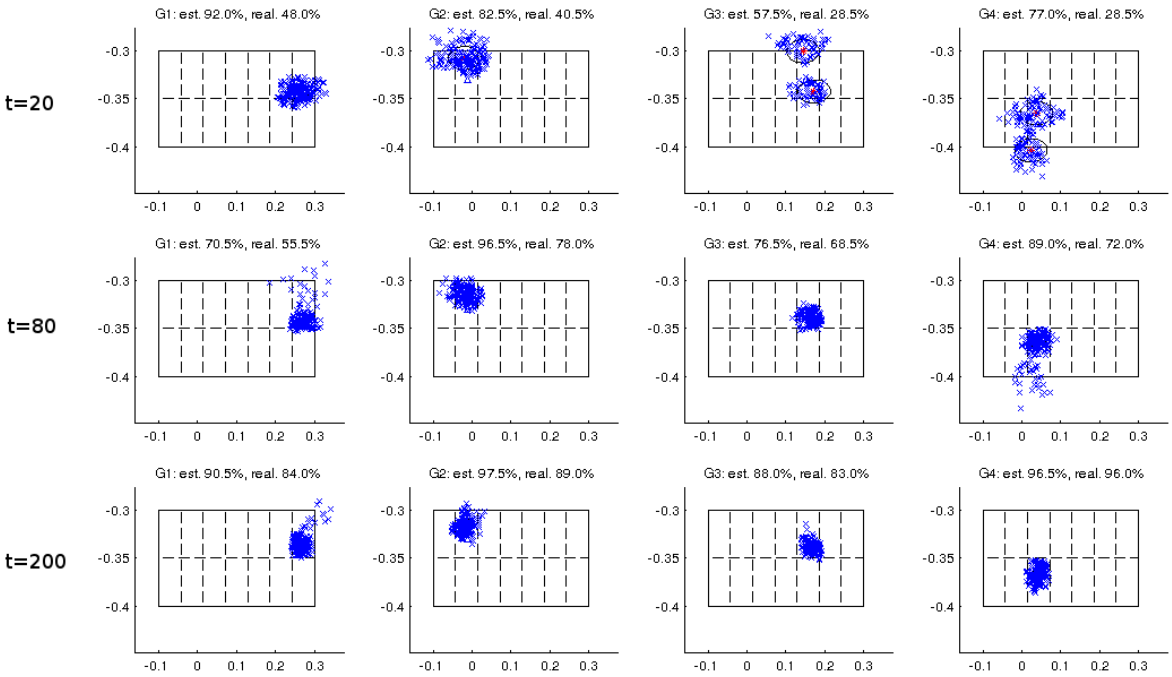

Fig. 7. Evaluation of an instrument model (unimodal distributions) at three different stages of learning, namely, after 20, 80 and 200 learning samples have been observed. Scatter plots for all 4 goals are shown, one per column, highlighting different stages in the learning process. Numbers on top of each plot show the estimated and real precision for the corresponding goal at that stage. At early stages, some of the goals remain undiscovered, meaning that the robot is still mainly exploring. As the interaction progresses, it can be seen that the learning focusses in discovered regions to better define the boundaries of the discovered goal regions.

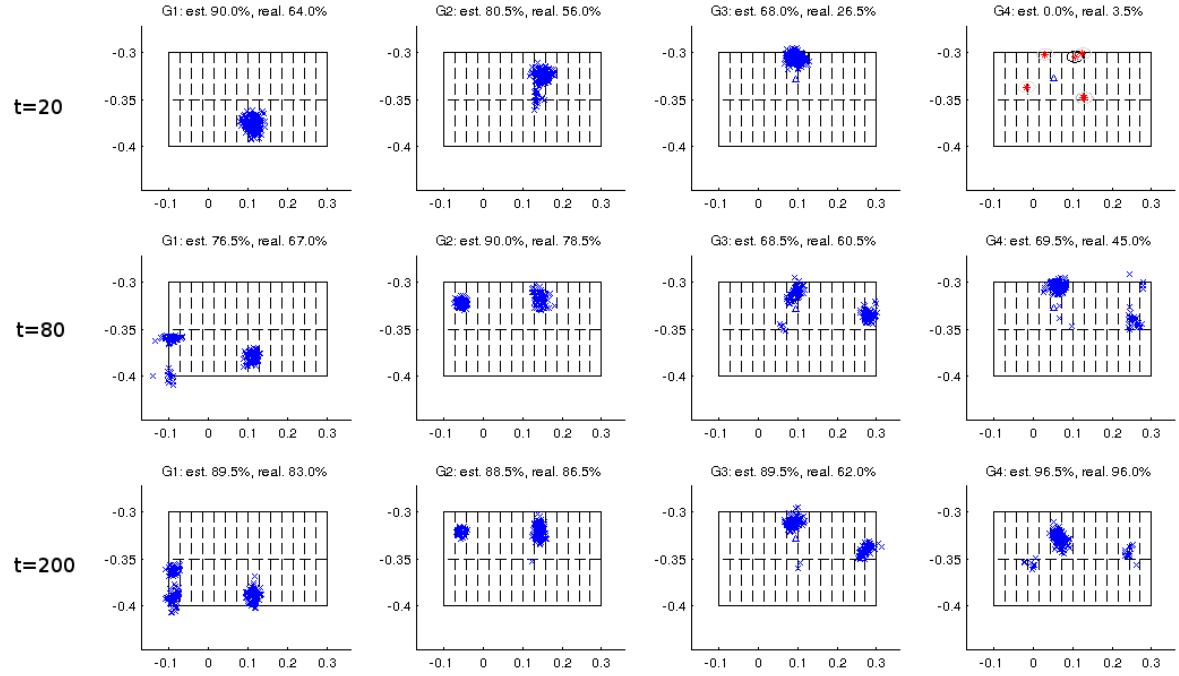

Fig. 8. Evaluation of an instrument model (multimodal distributions) at three different stages of learning, namely, after 20, 80 and 200 learning samples have been observed. Scatter plots for all 4 goals are shown, one per column, highlighting different stages in the learning process. Numbers on top of each plot show the estimated and real precision for the corresponding goal at that stage. In this case exploration is more difficult, as there are distant regions providing the same goal sounds. In early stages, although almost all the goals are discovered, not all the regions have been found, meaning the exploration is still ongoing. The later stages correspond to narrowing down and accurately defining the boundaries of the discovered regions.

candidates are displayed as white dots in the interface. There is also an online video showing the results of this experiment[3].

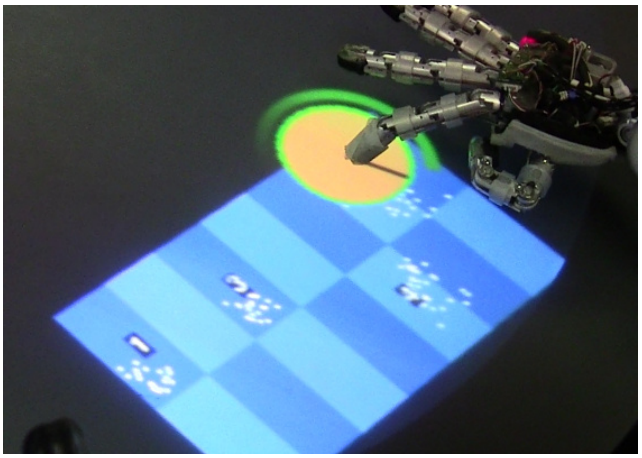

Fig. 11. Close up of the robot performing the imitation of the sound sequence. The virtual keys that were used to generate the sequence by the human are labelled as $1, 2, 3, 4$. White dots represent locations that are evaluated and filtered to select a good candidate for reaching.

D. Evaluation of the Instrument Model

As the robot has internal probabilistic models of how the instrument works, it can estimate how consistent its predictions are.

For this purpose, in order to evaluate how good the robot is at finding goal g_i, first we extracted a sample using Equation 2 and then evaluated each point by guessing the most likely sound $\hat{s}$ that should be heard at that location using Equation 4. The sounds were compared using $err(\hat{s}, g_i)$, and then we computed the percentage of correct guesses.

In our experiments we observed that, although this evaluation was usually too optimistic compared to the empirical evaluation, it did show the same trends as the empirical evaluation using the oracle, meaning that the derivative is very similar. In this way, this measure can be used as an estimate of its learning progress without the need of empirically assessing it through a new sequence of movements. Detecting a plateau in the learning progress is an indicative of convergence of the instrument model to stable predictions, which is the point where it should be confident enough to start performing the imitation of the sound sequence. Results can be seen in Figure 12 for an example learning trajectory.

E. Correcting reaching commands with the Body Model

Once the instrument model converged, the imitation of the sequence was tested. However, there are situations where the uncertainty about the end position of the hand undermines the performance of the robot.

In this case, the body model was used to keep track of this errors in different areas of the task space and provide estimates

[3]There is a video showing a performance of iCub using the tactile interface at https://www.youtube.com/watch?v=P1iWuzFfQn8

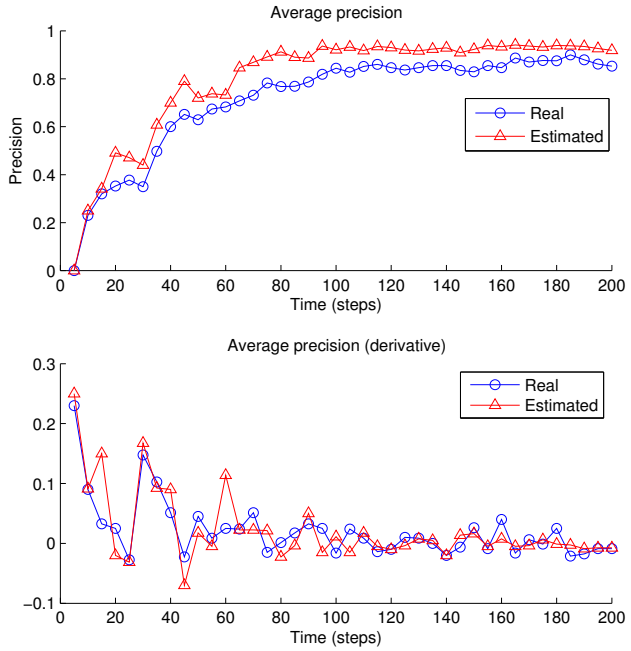

Fig. 12. Plot of instrument model evaluation results for an example of training sequence. The top plot shows the average precision for the four goals. We show the real precision obtained using the oracle and the estimation using the model at each time step. The bottom plot shows the derivative of the precision, where it can be seen that the trend in the estimated learning progress, seen as the change in estimated precision, follows more closely that of the empiric evaluation.

of where the real position of the hand will be if a particular action is executed. Then we used this in a feed-forward control loop to correct the action sent to the robot controller and minimize the impact of this error, as shown in Figure 13.

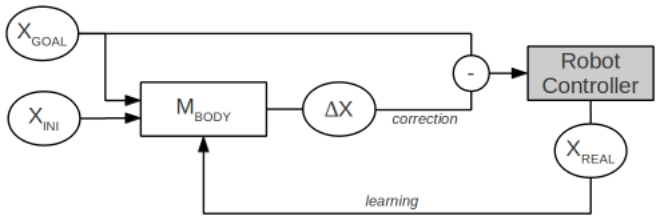

Fig. 13. Schematic of the action correction mechanism using the body model. The desired action and the current position of the end-effector is fed into the model, which provides corrections for the position, as well as an estimate of the temporal error in reaching that position.

Our initial experimental setup did not prove challenging enough to benefit from the corrections provided by the body model. For this reason, we increased the difficulty to evidence the two kinds of problems that our architecture is particularly suitable for. The change introduced was to increase the number of virtual keys, effectively reducing their individual size. The sounds produced were the same, but this time could be found in two different regions. The initial keyboard sequence of notes was A, B, C, D, E, F, G, with each virtual key having a size of about $4cm$, so it changed to $A, B, C, D, E, F, G, A, B, C, D, E, F, G$, resulting in each

virtual key decreasing its size to about $2cm$.
Not only this smaller keys resulted in an evident difficulty for the robot to find goal regions, as its reaching uncertainty region was therefore bigger in relation to key size, but also the predictive distribution of where each sound was found became multi-modal, which is a major difficulty for some models but not for the GMM used in our experiments.
However, the multi-modality present posed a decision problem for the robot. If we did not take reaching time into account, basically the robot tried to reach the location as fast as it could, resulting in many of the actions ending in an undesired location or simply not reaching them on time. Making use of the learnt body capabilities we had an effective filter for some of the candidates as the model considered them "*out of reach*" due to temporal constraints in the actions the robot could make.

Depending on the maximum velocity of action execution of robot actions and the distance of the different pairs of goals, using the corrections given by the body model provided a significant advantage over not using it. Figure 15 shows the success rate in reaching each of the four pairwise goals in the example demonstration, depicted in 14 using the numbers 1 to 4 to denote ordering.

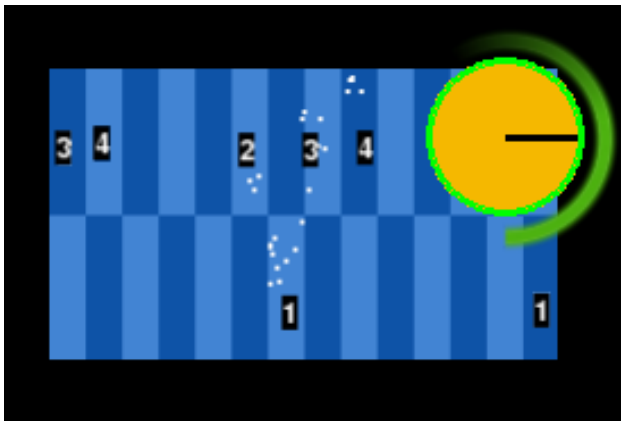

Fig. 14. Screenshot of the virtual keyboard interface showing the extended problem. It can be seen that goals marked with numbers 1 to 4 can be found in two different locations (object is over goal 2). The most difficult actions are movements from goal 2 to 3 and from goal 3 to 4.

V. Conclusions

In this work we proposed a system architecture which enables a humanoid robot to actively explore an object and obtain a model of how to use it for the purpose of achieving a set of goals given by a human supervisor. This applies to problems where goals are in the form of a sequence of perceptions that need to be obtained after executing a corresponding sequence of actions. In the proposed object model, as currently presented, only considers atomic actions, e.g. the end-points for a reaching behaviour. However, actions can encode the parameters of a full motion trajectory. In this way, the model should be extensible to more dynamic problems, like the execution of dance movements by teaching a series of goal body poses which serve as *key frames* for the whole motion sequence.

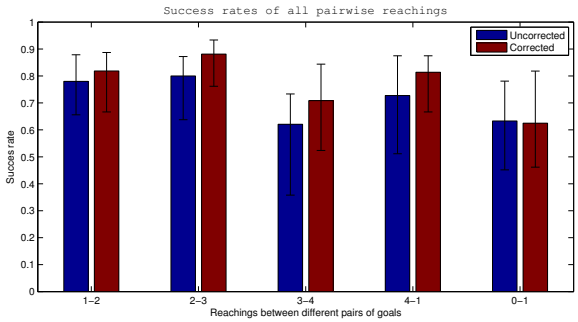

Fig. 15. Results for the evaluation of corrections using the Body GMM. Reachings are represented as pairs A-B, meaning a movement from goal A to goal B. It can be seen that for reachings 2-3 and 3-4, the corrections provide a significant improvement, due to the filtering in reaching time and a more accurate goal position estimation.

Event if a large set of goals can be accomplished by using the same object, usually most of them are not required by the task at hand, so the robot should not need to know everything about the object. By making the problem goal-based, we managed to allow the robot to focus exploration on a narrower set of actions. This is particularly useful for problems where the space of possible outcomes for actions is very big and we want the robot to quickly specialize in a subset of skills.

Also, in many real world robotics applications, the data used by the robot to learn using exploration behaviours arrives at a frequency such that there is enough time to apply inference techniques to actively choose actions based on current models. The kind of problems where the frequency at which consecutive data samples arrive is governed by an external process, as is the case of the music problem presented in this paper, makes our approach very suitable, given that the robot is able to use the time between data samples to plan an adequate action.

We illustrated this with an experiment based in the imitation of a sequence of musical notes played by a humanoid robot in a virtual keyboard displayed in a visuo-tactile interface.
Our results indicate that, by using an active learning strategy based in an information-theoretic measure, the robot was able to acquire the required knowledge faster than if using a random exploration strategy following only the predictions provided by the current model.

Moreover, the embodiment of the robot affects the interaction dynamics with the object it is exploring, in the sense of the actions not resulting in exactly the desired perceptions.
In our experiments, the robot has a reaching error that depends both on its physical body dynamics and also on the software controller that guides its hand to the desired location. Time constraints also play an important role, due to the fact that higher movement speeds result in higher spatial error.
The proposed architecture, integrating a model of the body constraints, takes advantage of such information to provide an error correction control module which predicts the expected result of the desired action and corrects the action to minimize that expected error. In problems where tuning the action controller is very difficult or impractical, introducing a model which learns control uncertainties and provides action corrections addresses the problem of fine-calibration of robot

controllers.

The robot can also give an estimate of the feasibility of the actions needed to accomplish the required goals. This may serve as a good indicator for the human supervisor about the difficulty of the given sequence subjective to the robot. This property not only alleviates the need to know exactly what actions the robot can or can not perform, but also serves as a communication tool because such subjective judgement is given when the robot is confident enough about the knowledge it has.

The evaluation of the correction module showed no significant improvement on a simple setup of the object, although with a more complex setup, where the robot can obtain the same goal in multiple locations, i.e. displaying multi-modal predictive distributions, some of the actions could not be performed under the desired time due to body constraints. Our probabilistic model successfully filtered such unattainable candidate actions, keeping the robot from executing unsafe operations.

ACKNOWLEDGMENT

This work was supported in part by the Generalitat de Catalunya to Consolidated Groups 2014 SGR Grant 118, the CSIC intramural project 201250E054 and the EU FP7 Project WYSIWYD under Grant 612139.

REFERENCES

[1] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, pp. 55–66, 2010.

[2] M. Rolf, J. J. Steil, and M. Gienger, "Online goal babbling for rapid bootstrapping of inverse models in high dimensions," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2. IEEE, 2011, pp. 1–8.

[3] 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.

[4] J. Kulick, M. Toussaint, T. Lang, and M. Lopes, "Active learning for teaching a robot grounded relational symbols," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 1451–1457.

[5] M. Cakmak, C. Chao, and A. L. Thomaz, "Designing interactions for robot active learners," *Autonomous Mental Development, IEEE Transactions on*, vol. 2, no. 2, pp. 108–118, 2010.

[6] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[7] A. N. Meltzoff, "Elements of a developmental theory of imitation," *The imitative mind: Development, evolution, and brain bases*, pp. 19–41, 2002.

[8] E. W. Bushnell and J. P. Boudreau, "Motor development and the mind: The potential role of motor abilities as a determinant of aspects of perceptual development," *Child development*, vol. 64, no. 4, pp. 1005–1021, 1993.

[9] A. H. Fagg and M. A. Arbib, "Modeling parietal–premotor interactions in primate control of grasping," *Neural Networks*, vol. 11, no. 7, pp. 1277–1303, 1998.

[10] E. Thelen, "Motor development as foundation and future of developmental psychology," *International journal of behavioral development*, vol. 24, no. 4, pp. 385–397, 2000.

[11] J. J. Gibson, "The ecological approach to visual perception," 1979, houghton Mifflin.

[12] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning about objects through action-initial steps towards artificial cognition," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 3. IEEE, 2003, pp. 3140–3145.

[13] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensory–motor coordination to imitation," *Robotics, IEEE Transactions on*, vol. 24, no. 1, pp. 15–26, 2008.

[14] A. Ribes, J. Cerquides, Y. Demiris, and R. L. De Mántaras, "Incremental learning of an optical flow model for sensorimotor anticipation in a mobile robot," in *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–2.

[15] S. Calinon, "Robot programming by demonstration," in *Springer handbook of robotics*. Springer, 2008, pp. 1371–1394.

[16] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *The International Journal of Robotics Research*, pp. 330–345, 2011.

[17] T. Cederborg, M. Li, A. Baranes, and P.-Y. Oudeyer, "Incremental local online gaussian mixture regression for imitation learning of multiple tasks," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 267–274.

[18] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation," *Robotics & Automation Magazine, IEEE*, vol. 17, no. 2, pp. 44–54, 2010.

[19] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection Science*, vol. 15, no. 4, pp. 151–190, 2003.

[20] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 2, pp. 265–286, 2007.

[21] P.-Y. Oudeyer, F. Kaplan, *et al.*, "How can we define intrinsic motivation?" in *Proceedings of the 8th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, 2008.

[22] S. Ivaldi, S. Nguyen, N. Lyubova, A. Droniou, V. Padois, D. Filliat, P.-Y. Oudeyer, and O. Sigaud, "Object learning through active exploration," 2013.

[23] A. Dearden and Y. Demiris, "Learning forward models for robots," in *IJCAI*, vol. 5, 2005, p. 1440.

[24] Y. Demiris and A. Dearden, "From motor babbling to hierarchical learning by imitation: a robot developmental pathway," 2005.

[25] R. Saegusa, G. Metta, G. Sandini, and S. Sakka, "Active motor babbling for sensorimotor learning," in *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*. IEEE, 2009, pp. 794–799.

[26] M. Rolf, J. J. Steil, and M. Gienger, "Goal babbling permits direct learning of inverse kinematics," *Autonomous Mental Development, IEEE Transactions on*, vol. 2, no. 3, pp. 216–229, 2010.

[27] A. Baranes, P.-Y. Oudeyer, *et al.*, "Bootstrapping intrinsically motivated learning with human demonstration," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2. IEEE, 2011, pp. 1–8.

[28] C. Moulin-Frier and P.-Y. Oudeyer, "Exploration strategies in developmental robotics: a unified probabilistic framework," in *Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on*. IEEE, 2013, pp. 1–6.

[29] P. Hemakumara and S. Sukkarieh, "Non-parametric uav system identification with dependent gaussian processes," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4435–4441.

[30] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "Gaussian processes and reinforcement learning for identification and control of an autonomous blimp." in *ICRA*, 2007, pp. 742–747.

[31] Y. Su, Y. Wu, H. Soh, Z. Du, and Y. Demiris, "Enhanced kinematic model for dexterous manipulation with an underactuated hand," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2493–2499.

[32] J. Porrill and P. Dean, "Recurrent cerebellar loops simplify adaptive control of redundant and nonlinear motor systems," *Neural computation*, vol. 19, no. 1, pp. 170–193, 2007.

[33] D. Ognibene and Y. Demiris, "Towards active event recognition," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 2495–2501.

[34] M. Petit, S. Lallée, J.-D. Boucher, G. Pointeau, P. Cheminade, D. Ognibene, E. Chinellato, U. Pattacini, I. Gori, U. Martinez-Hernandez, *et al.*, "The coordinating role of language in real-time multimodal learning of cooperative tasks," *Autonomous Mental Development, IEEE Transactions on*, vol. 5, no. 1, pp. 3–17, 2013.

[35] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard, "Yaafe, an easy to use and efficient audio feature extraction software." in *ISMIR*, 2010, pp. 441–446.

[36] J. Marques and P. J. Moreno, "A study of musical instrument classification using gaussian mixture models and support vector machines," *Cambridge Research Laboratory Technical Report Series CRL*, vol. 4, 1999.

[37] A. Eronen, "Comparison of features for musical instrument recognition," in *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the*. IEEE, 2001, pp. 19–22.

[38] P. Herrera, A. Yeterian, and F. Gouyon, "Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques," in *Music and Artificial Intelligence*. Springer, 2002, pp. 69–80.

[39] A. Berenzweig, B. Logan, D. P. Ellis, and B. Whitman, "A large-scale evaluation of acoustic and subjective music-similarity measures," *Computer Music Journal*, vol. 28, no. 2, pp. 63–76, 2004.

[40] S. Thrun and K. Möller, "Active exploration in dynamic environments," in *Advances in neural information processing systems*, 1992, pp. 531–538.

[41] S. Thrun, "Exploration in active learning," *Handbook of Brain Science and Neural Networks*, pp. 381–384, 1995.

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

[43] M. F. Huber, T. Bailey, H. Durrant-Whyte, and U. D. Hanebeck, "On entropy approximation for gaussian mixture random vectors," in *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*. IEEE, 2008, pp. 181–188.

[44] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.

[45] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*. ACM, 2008, pp. 50–56.

[46] U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini, "An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1668–1674.

[47] P. Fitzpatrick, G. Metta, and L. Natale, "Towards long-lived robot genes," *Robotics and Autonomous systems*, vol. 56, no. 1, pp. 29–45, 2008.

[48] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[49] 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.

[50] J. Cerquides and R. L. De Mántaras, "Tan classifiers based on decomposable distributions," *Machine Learning*, vol. 59, no. 3, pp. 323–354, 2005.