Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

Online Multimodal Ensemble Learning using Self-learned Sensorimotor Representations

Martina Zambelli and Yiannis Demiris

Personal Robotics Lab, Dept. of Electrical and Electronic Engineering, Imperial College London, UK.
{m.zambelli13, y.demiris}@imperial.ac.uk.

Abstract

Internal models play a key role in cognitive agents by providing on the one hand predictions of sensory consequences of motor commands (forward models), and on the other hand inverse mappings (inverse models) to realise tasks involving control loops, such as imitation tasks. The ability to predict and generate new actions in continuously evolving environments intrinsically requiring the use of different sensory modalities is particularly relevant for autonomous robots, which must also be able to adapt their models online. We present a learning architecture based on self-learned multimodal sensorimotor representations. To attain accurate forward models, we propose an online heterogeneous ensemble learning method that allows us to improve the prediction accuracy by leveraging differences of multiple diverse predictors. We further propose a method to learn inverse models on-the-fly to equip a robot with multimodal learning skills to perform imitation tasks using multiple sensory modalities. We have evaluated the proposed methods on an iCub humanoid robot. Since no assumptions are made on the robot kinematic/dynamic structure, the method can be applied to different robotic platforms. *Keywords: Sensorimotor contingencies, online learning, ensemble learning, multimodal imitation learning.*

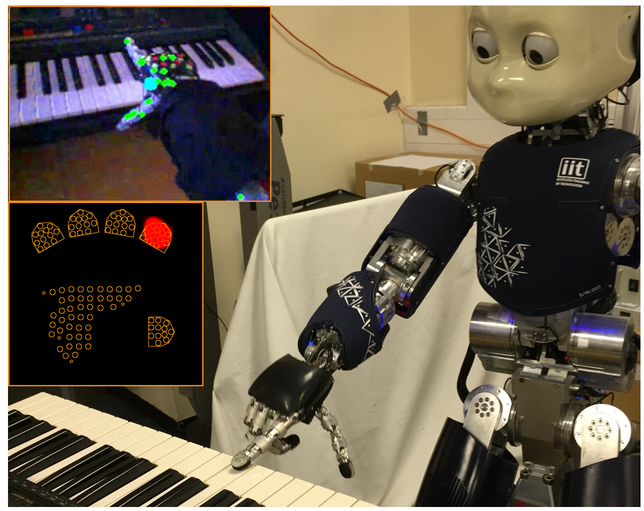


Figure 1: The iCub robot learns to interact with a piano keyboard through babbling and imitation. The top-left pictures show the view from the robot's left eye camera and the hand taxels activated when keys are pressed.

1 Introduction

Complex robots rely on internal models describing the kinematics and dynamics for controlling and planning actions; however, constructing analytical models of complex robotic platforms often presents critical difficulties and costs. In particular, analytical models might be inaccurate because they are based on assumptions that are not realistic, such as the complete rigidity of the links. Also nonlinearities and uncertainties of the system are often difficult to include. Another critical problem related to the use of analytical models is that they are highly specific for a particular robotic platform. This does not only limit their use to that particular platform, but also requires that the model must be completely revised in case of modifications of the platform, thus adding time, computational and economical costs.

These observations motivate the interest in endowing robots with learning capabilities, in order to enable them to build their internal models through learning processes [1–3] in which relations between actions and associated changes in sensory input, are involved. Through autonomous learning, robots can develop autonomous behaviours and formulation of decisions. Another benefit of self-learned models is that, in principle, they can update during the life-time of the robot, avoiding re-modelling in case, for example, of damaged parts or hardware failures. Because the internal models are learned by the robot continuously, they can handle changes in the robot morphology or in the robot sensory system, while eliminating the need for explicit analytical model formulation and dealing with model drifts [4]. Contrary to more classical control approaches based on hand-crafted kinematic and dynamic models, methods based on self-learning of sensorimotor representations can achieve complex behaviours, such as imitation tasks, without the need of explicit model formulation and without solving inverse kinematics problems, while giving the flexibility and adaptability of learned models, as well as the possibility to apply the same method on different robotic platforms.

Some autonomous learning approaches to build internal models take inspiration from neuroscientific studies arguing that infants use self-exploration and self-stimulation to "calibrate" their sensorimotor and body representations [5]. Analogously,

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

a robot can explore its sensorimotor capabilities through self-exploration, or *motor babbling* [6]. The role of actions is constitutive in this learning process [7,8]. No internal representations of the world are designed *ad hoc* to generate sensory awareness. Instead, robots can learn internal models to predict sensorimotor representations, given the current sensory states and the motor commands; these are also known as *forward models*. These models have been related to the central nervous system that internally simulates the motor system in planning, control and learning [9]. The other type of internal model, commonly coupled with forward models, is the *inverse model*, which is a mapping in the opposite direction: given a target (goal) state and the current state, it provides the motor commands needed to reach the goal. Forward and inverse models have been employed to build biologically inspired control architectures [10–13].

Complex robots are usually equipped with multiple different sensors and require accurate sensorimotor representations to perform many tasks which often involve multiple sensory modalities. It is thus desirable to learn accurate forward models which can adapt to potential new conditions, and also to leverage the information available to the robot from multiple sensor modalities. Motivated by these observations, we focus on the one hand on online learning of forward models where the accuracy of predictions is improved by using multiple different predictors to form an ensemble learning structure. On the other hand, we develop a new method to achieve multimodal imitation learning on robots, using *self-learned* sensorimotor representations from visual, proprioceptive, tactile and auditory stimuli. This method is the generalisation of our previously presented approach to multimodal imitation [14]. This method is based on the construction of multimodal sensorimotor representations, does not require multiple demonstrations, nor training complex learning structures, and allows to perform on-the-fly multimodal imitation by combining self-learned sensorimotor representations.

The main contributions of this paper can be summarised as follows: *(i)* an online heterogeneous ensemble learning method is presented to achieve accurate predictions and learn accurate forward models, which are able to generalise on new data trajectories; *(ii)* a generalisation of the method proposed in [14] for learning inverse models based on multimodal self-learned sensorimotor representations is presented to achieve multimodal imitation learning tasks; *(iii)* a control architecture to learn and use sensorimotor representations is illustrated, where forward and inverse models can work in parallel; *(iv)* evaluation of the proposed architecture on a multimodal imitation task using an iCub humanoid robot [15] and a piano keyboard; since no assumptions on the morphology nor on the kinematics/dynamics of the robot are made, the proposed framework can be applied to different robotic platforms.

2 Background

Online learning of internal models: A real challenge in robot learning is devising algorithms that allow online adaptation. The importance of online learning in robotic applications is related to the fact that robots are required to interact in a continuously evolving environment; also, changing of contexts, such as tools being used or human to interact with, make online strategies attractive [16–18]; another motivation for online model learning is that it is difficult if not impossible to cover the complete state space with data beforehand [19]. Although many state-of-the-art learning paradigms still require batch processing, some works have proposed online strategies in the field of humanoid robot learning. A biologically inspired model for online and continuous learning of visuo-motor coordination has been proposed in [20], where dynamic self-organising maps associated through Hebbian links have been adopted for learning the visuo-motor coordination online on a Nao humanoid robot. An online learning approach to achieve reaching behaviour in a humanoid robot has been proposed in [16], where the receptive field weighted regression algorithm has been employed to learn online a representation of the robot's reachable space. In [17] an online strategy has been implemented to learn the kinematic structure of a humanoid robot, yielding the position of each segment and computing the associated Jacobians. However, these studies rely on learning structures which are not parameter free, and they are either goal/task-directed or aim at identifying specific kinematic parameters.

Sensorimotor exploration and learning: Inspired by neuroscientific studies arguing that infants use self-exploration and self-stimulation to "calibrate" their sensorimotor and body representations [5], autonomous learning approaches have been implemented on robots based on random self-exploration, or *motor babbling* [6]. Other studies claim that early movements are already goal-directed [21], and a number of approaches in robotic learning follow this approach [22,23]. In this study, we will follow the first approach and ground the learning methods on random motor-babbling.

Ensemble learning: Many ensemble learning algorithms have been used in robotics to solve problems such as localization, detection, recognition, decision making [24,25]. Ensemble methods have been shown effective in different learning frameworks to achieve high accuracy of predictions [26–28]. However, the use of combinations of different predictors has rarely been deployed to build internal models or to learn and produce motion behaviours on robots. In ensemble learning systems, multiple models, such as classifiers or experts, are strategically generated and combined to solve a particular problem. The key for the success of an ensemble system and its ability to correct the errors of some of its members is the diversity among the classifiers of the ensemble. The intuition is that if each model makes different errors, then a strategic combination can reduce the total error. Several approaches exist to solve offline regression problems [28]. Online ensemble learning algorithms for regression have received less attention, although recent works have presented advances in this field [29,30].

Inverse model learning: While forward models are uniquely determined, inverse models are generally not and do not always exist. Direct inverse modelling treats the problem of learning an inverse model as a classical supervised learning problem, however the main problem of these methods is in learning control of redundant robots because of the existence of a one-to-many

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

inverse kinematics problem [31]. Recent approaches have been proposed in which the inverse model is directly learned from data, by adopting machine learning techniques. For example, to learn the inverse kinematics of a humanoid robot, the Locally Weighted Projection Regression (LWPR) algorithm has been used in [32], while the Infinite Mixture of Linear Experts (IMLE) algorithm has been used in [33]. Both these approaches allow to learn the model while performing the task, similarly to the case in our study. However, these approaches are based on finding an approximation for the Jacobian, not considering different types of sensory modalities. A broadly used class of methods to learn inverse models is the Learning by Demonstration, or Programming by Demonstration framework. Seminal works on this class of approaches are for example [34–38]. One possibility to acquire a target trajectory and to learn inverse models from data is to collect multiple demonstrations of each task. This approach has been explored for example in [39–43], where statistical methods have been proposed to infer models from multiple demonstrated trajectory. However, this approach requires multiple demonstrations, which might be difficult to obtain. Other methods are based on a single demonstration instead, *e.g.* in [44–47]. In this case, however, it is assumed that all input features are observable. A diverse approach was proposed in [48,49], where context free grammars were used in order to learn sequences of demonstrated actions. A considerable number of works has recently addressed the problem of task learning and policy search [50–56], adopting reinforcement learning algorithms to learn complex movement tasks in robotics. However, these studies present a common limitation, that is they are usually task specific.

Multimodal Imitation learning: Whereas a variety of different sensors has become available on complex robots, most of the approaches to imitation learning are based on the use of data from a single modality, such as vision [46,57,58]. However, there exist studies that have taken into consideration multiple modalities for learning, *e.g.* [59,60]. A combination of sound and movements has been adopted in [59] to imitate human drumming behaviours, while motion and force data have been used to teach grasping gestures to a simulated manipulator in [60]. Other studies, *e.g.* [61–64], have presented solutions for merging different sensors' data to address classification-type problems, such as object/gesture recognition or speaker identification/spatial localisation, rather than imitating demonstrated behaviours. Different approaches have been used to solve imitation learning using different sensor informations, such as hierarchical architectures based on multiple internal models [46,57–59], and Gaussian Mixture Regression together with Hidden Markov Model [60]. With the goal of reproducing a human trajectory, motion capture systems, kinesthetic and teleoperation are often used [59,60], although manual design of the system and usually a certain number of demonstrations are required by these approaches. Motor babbling and self-exploration, as opposite, are bottom-up approaches [57,65], which require less prior design and leverage the advantages of a developmental approach to learning, such as more autonomy, incremental learning and adaptability to new conditions.

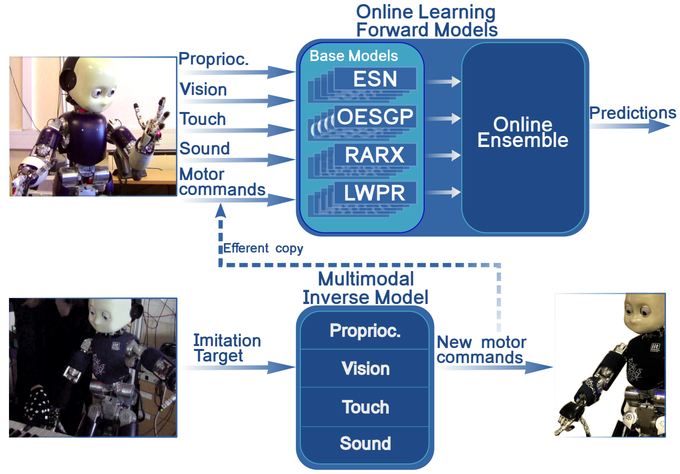


Figure 2: Architecture summary: forward models produce predictions of the sensory state given the current state and a motor command. This motor command can be the efferent copy of the signal calculated by the inverse model, which produces a new motor command given a target state. *(Upper part)* Online ensemble learning of sensorimotor representations from motor babbling. Data from multiple sensors are available from the robot, such as vision, proprioception, touch and sound data. The proposed approach allows to learn forward models online by building an ensemble of multiple different learning algorithms, namely Echo State Networks (ESNs), Online Echo State Gaussian Processes (OESGPs), Recursive ARX (RARX) and Locally Weighted Projection Regression (LWPR) models. *(Lower part)* Imitation targets are demonstrated by a human. The multimodal inverse model generates a new motor command to meet concurrent multimodal task constraints using multimodal sensorimotor representations.

3 Learning Sensorimotor Representation Models from Self-exploration

Studies in the area of child development and neuroscience on motor behaviour of infants have shown that mobility at early age is characterised by variations in movement trajectories, in temporal and quantitative aspects, that are not neatly tuned to environmental conditions [66]. In a first phase, infants perform movements characterised by this variability, also known as general movements, consisting of series of gross movements of variable speed and amplitude, which involve all parts of the body but lack distinctive sequencing. Only in a second phase, general movements are gradually replaced by goal-directed movements. Taking inspiration from these observations and from motor babbling approaches [5,6,57,65], the structure of the proposed learning architecture is based on a first self-exploration phase when sensorimotor representations are learned, and on a second phase where goal-directed movements are performed to imitate actions.

During the exploration phase, the robot performs pseudo-random movements in order to collect data from its sensorimotor system, so that relationships between motor commands and sensory effects can be learned. A forward model mapping is learned online, that is in parallel to the execution of the exploration movements. The forward model allows to predict the next sensory state given the current state and an action. The in-

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

formation accumulated during the self-exploration can be used to learn inverse models. This allows to generate on-the-fly a new desired motor command given a target state and the current state of the robot. This is necessary, for instance, to achieve imitation tasks where the robot needs to imitate a demonstrator behaviour.

A preview of the learning architecture underlying the learning and update of the internal models is shown in Fig. 2. The learning architecture proposed in this study retraces well-known learning architectures proposed for example in [2,3], in which coupled forward and inverse models are used to achieve learning and motion behaviour on robots. In particular, studies in the neuroscientific field have shown the existence of forward models in the human brain, that allow us to make predictions from sensorimotor stimula. A more comprehensive discussion on the biological plausibility of this general learning scheme has been presented in [67]. Although the learning architecture and the learning strategy, based on self-exploration and motor babbling, are biologically inspired, the implementation of the proposed methods are engineered solutions that resort machine learning approaches to build forward and inverse models.

3.1 Forward Model Learning

The formalisation of our method is general and not confined to specific modalities nor robotic platforms. Therefore we define the following general notation. We denote the sensory state vector containing different modalities (note that some modalities can be multidimensional) as $\mathsf{x} = [\mathsf{x}_1\ \mathsf{x}_2\ \ldots\ \mathsf{x}_\mathsf{N}]^T$ and the motor commands applied to the M motors of the body part as the vector $\mathsf{m} = [\mathsf{m}_1\ \mathsf{m}_2\ \ldots\ \mathsf{m}_\mathsf{M}]^T$.

Following seminal works such as [31,68] and [8], we formulate a sensorimotor representations model as a system that can be described by the equation

$$\mathsf{x}' = \mathsf{f}(\mathsf{x}, \mathsf{m}) \tag{1}$$

where x' is the sensory state after applying the motor command m from state x. The unknown map f represents the sensorimotor representations model; in our work, this map is learned incrementally throughout the experience accumulated by the robot itself. The sets where x and m take values are denoted as $\mathcal{X}$ and $\mathcal{M}$, respectively.

Exploration steps are performed to acquire a set of data points used to learn online an estimate of the sensorimotor representation model f. This is used as forward model to predict the next sensory data x', given the current state x and a motor command m. To learn the forward mapping, we propose a new method based on an online ensemble of different online regression algorithms. This method allows to obtain accurate predictions by leveraging the properties of the ensemble strategy adopted.

The ensemble learning strategy: Main ingredients in ensemble learning methods are ensemble generation and ensemble integration. The generation of base models is referred to as *ensemble generation*. The objective is to build a set of N_{bm} base models, also called *pool of models* $\mathcal{F}_{N_{bm}} = \{\hat{f}_h,\ h = 1, \ldots, N_{bm}\}$, to approximate a true function f. If the models in $\mathcal{F}_{N_{bm}}$ are all generated using the same induction algorithm, the ensemble is called *homogeneous*, while if more than one algorithm is used to build $\mathcal{F}_{N_{bm}}$, the ensemble is *heterogeneous*. Less work exists on heterogeneous ensembles than in homogeneous ones [28]; however, combining different algorithms is a promising strategy to obtain diversity, which has been shown of great importance in enhancing prediction accuracy [69].

The ensemble integration step can be realised in a number of different ways. A common solution is to take the weighted average of the base models: $\hat{f}_\mathcal{F} = \sum_{h=1}^{N_{bm}} w_h \hat{f}_h$, where $w_h \in [0, 1]$, $\sum_{h=1}^{N_{bm}} w_h$=1, are the weights assigned to each base model $\hat{f}_h$. The weights state the importance of the single base models in building the ensemble, according to some application-dependent or optimality criterion. The weights can be constant or dynamically calculated according to each data sample. Popular algorithms to obtain ensemble weights are stacked regression [70] and dynamic weighting [71]. Given a learning set $\mathcal{L}$ with K data samples, the stacked regression approach calculates the weights by minimising $\sum_{k=1}^{K} \left[f(\mathbf{x}_k) - \sum_{h=1}^{N_{bm}} w_h \hat{f}_h(\mathbf{x}_k) \right]^2$, while the dynamic weighting method sets the weights according to performance measurement of the predictors.

The proposed online heterogeneous ensemble: We propose a heterogeneous online ensemble learning algorithm which combines predictors of different natures in an online manner.

Among the set of online learning methods, we consider four algorithms that have been shown effective in a number of diverse applications: the Echo State Networks (ESN) [72], which are a class of recurrent neural networks; the Online Echo State Gaussian Processes (OESGPs) [73], which combine ESN with sparse Gaussian Processes; the Locally Weighted Projection Regression (LWPR) [74], which exploits piecewise linear models to realise an incremental learning algorithm; and recursive ARX models (RARX) identified using the recursive least square method [75,76] (see Fig. 2).

These four algorithms differ from each other in several aspects: firstly, while the ESN, OESGP and LWPR are non-parametric approaches, the RARX is parametric and fits the data by finding polynomial coefficients. Also, the chosen algorithms rely on different structures, *i.e.* neural networks, Gaussian processes, piecewise linear models, polynomial transfer functions. Contrary to a large number of other learning algorithms, these are online algorithms, which are able to update as new data is available, iteratively or recursively. Moreover, an advantage of using these different state-of-the-art algorithms is that their dissimilarities guarantee the necessary diversity between the base models that constitute the ensemble; in particular different types of prediction errors (*e.g.* overshoot vs. undershoot, offsets) are given by the different algorithms.

The base models are trained separately and in parallel, and the update step is different for each of the diverse models. In the ESN model, only the output weights ($\mathbf{w}^{out}$) of the recurrent neural network are updated; the prediction is then obtained by $\tanh(\mathbf{w}^{out}\mathsf{x}(t))$, [72]. In the OESGP model, the prediction

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

is made through the Gaussian predictive distribution $\mathcal{N}(\mu, \sigma^2)$, where the mean μ and the variance σ^2 are estimated incrementally during training, [73]. The RARX model updates the parameter estimates $\hat{\theta}$ at each iteration, while the prediction is calculated as $\psi^T(t)\hat{\theta}(t)$ where ψ represents the gradient of the predicted model output [76]. The LWPR model updates local models parameters by minimising a predicted residual sums of squares function and then produces an estimated $\hat{\mathsf{f}}$ as a weighted combination of local models, [74].

We instantiate five models for each learning algorithm by using different initialisation parameters: for example different numbers of internal units in the ESNs, different length scales for the Gaussian distributions in the OESGPs, different orders for the polynomials of the RARX models and different weight activation thresholds for new local models to be generated for LWPR. Each base model provides an estimate $\hat{\mathsf{f}}_i$ of the true function f, that can be used to compute predicted values of $\hat{\mathsf{x}}'_i$. The ensemble prediction is obtained by combining the base models' estimates through the ensemble integration. The subscript E will be used to indicate the ensemble estimates in the following. The ensemble estimate $\hat{\mathsf{x}}_\mathsf{E}$ is computed online at each time step t as:

$$\hat{\mathsf{x}}_\mathsf{E}(t) = \sum\nolimits_i w_i(t)\hat{\mathsf{x}}_i(t) \tag{2}$$

where $w_i(t)$ are the normalised ensemble weights ($\sum_i w_i(t){=}1$); the aim is to calculate the ensemble weights so that the combination of models gives the closest estimate to the true value to be predicted.

We propose a new method to update the ensemble weights that combines the ideas of both dynamic weighting methods and stacked regression approaches. The proposed method also takes into account both the cumulative and current base models' performance, so that both the overall behaviour and the accuracy at the last data point available are used to evaluate the base models.

At each time step t, that is for each new data point available, each base model i produces an estimate $\hat{\mathsf{x}}'_i$. The performance scores obtained are evaluated through the following two values:

$$e_{c,i}(t) = \sum_{\tau=1}^{t} E\left(\mathsf{x}'(\tau) - \hat{\mathsf{x}}'_i(\tau)\right)^2,$$

$$e_{s,i}(t) = E\left(\mathsf{x}'(t) - \hat{\mathsf{x}}'_i(t)\right)^2$$

which are the cumulative mean squared error and the mean squared error at current time step, respectively, for all the base models $i = 1, ..., N_{bm}$. Note that since the algorithm calculates estimates online, the value x', which will be observed at $t{+}1$, is not yet available when the ensemble prediction $\hat{\mathsf{x}}'_\mathsf{E}$ is calculated.

A convex combination of the current and cumulative errors is then used to compute the score

$$\beta_{cs,i}(t) = \alpha e_{c,i}(t)^{-1} + (1-\alpha)e_{s,i}(t)^{-1},$$

where α is a parameter chosen so that more importance is given to the inverse of the cumulative error than to the inverse of the current error ($\alpha > 0.5$).

Then, meta-weights γ_i are calculated in order to minimise the prediction error that would be obtained by the ensemble of base models weighted with weights $\beta_{cs,i}(t)$, that is solving

$$\min_{\gamma_i} \sum_{\tau=1}^{t} \left[\mathsf{x} - \sum\nolimits_i \gamma_i(\tau)\beta_{cs,i}(\tau)\hat{\mathsf{x}}_i(\tau)\right]^2 \tag{3}$$

The final ensemble weights are then computed as

$$w_i(t) = \frac{\gamma_i(t)\beta_{cs,i}(t)}{\sum_i \gamma_i(t)\beta_{cs,i}(t)}. \tag{4}$$

Note that at each time step the quality of each base model is reassessed and the ensemble weights change dynamically.

3.2 Inverse Model Learning

Learning the motor commands required to achieve a target point x^* given the current state is also known as inverse model learning. We present here the generalisation of the work we first proposed in [14]. Consider multiple modalities, yielding a state space of dimension N. The data collected during n self-exploration movements can be juxtaposed to form a $\mathsf{N} \times n$ *multimodal sensory matrix*

$$\mathsf{S} = \begin{bmatrix} \mathsf{S}_1 \\ \vdots \\ \mathsf{S}_\mathsf{N} \end{bmatrix} = \begin{bmatrix} \Delta_{\mathsf{x}_1,1} & \cdots & \Delta_{\mathsf{x}_1,n} \\ \vdots & \ldots & \vdots \\ \Delta_{\mathsf{x}_\mathsf{N},1} & \cdots & \Delta_{\mathsf{x}_\mathsf{N},n} \end{bmatrix}, \tag{5}$$

where $\Delta_{\mathsf{x}_\nu,i}$ represents the change of the sensory state ν that has been observed during the execution of the exploration movement i; more specifically $\Delta_{\mathsf{x}_\nu,i}$ encodes the change of the sensory state relative to the starting position. Analogously the motor commands issued to the joints of the robot's arm used to perform the n self-exploration movements can be juxtaposed to form a $\mathsf{M} \times n$ *actuation primitives matrix*

$$\mathsf{A} = \begin{bmatrix} \mathsf{A}_1 \\ \vdots \\ \mathsf{A}_\mathsf{M} \end{bmatrix} = \begin{bmatrix} v_{1,1} & \cdots & v_{1,n} \\ \vdots & \vdots & \vdots \\ v_{\mathsf{M},1} & \cdots & v_{\mathsf{M},n} \end{bmatrix}. \tag{6}$$

Target trajectories defining an imitation task can be expressed as reference trajectories (functions of time) $\mathsf{r}_1(t), \mathsf{r}_2(t), ..., \mathsf{r}_\mathsf{N}(t)$. At each time, the imitation error ε, defined as the difference between the reference $\mathsf{r}_\nu(t)$ and the current state $\mathsf{x}_\nu(t)$, on each modality space, is defined as

$$\varepsilon(t) = \begin{bmatrix} \varepsilon_1(t) \\ \vdots \\ \varepsilon_\mathsf{N}(t) \end{bmatrix} = \begin{bmatrix} \mathsf{r}_1(t) - \mathsf{x}_1(t) \\ \vdots \\ \mathsf{r}_\mathsf{N}(t) - \mathsf{x}_\mathsf{N}(t) \end{bmatrix}. \tag{7}$$

At each step during the execution of the imitation task, the robot moves towards the next reference point, using a combination of the primitives explored. The velocity commands to apply to the motors in order to achieve the multimodal target, defined as the vector $\mathsf{v}^* = \left[v_1^*, v_2^*, \cdots, v_\mathsf{M}^*\right]^T$, is obtained as a combination of the those primitives that led to sensory results

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

which are close to the current target. The experience accumulated during the exploration and the learned sensorimotor representations can then be scanned to search for those states that are closest to the vector ε. This corresponds to a search on the multimodal space which includes at the same time all the modalities' constraints.

A range search strategy is implemented using a kd-tree [77] created from the multimodal sensory matrix S in order to optimize the search. The kd-tree algorithm partitions the multimodal sensory matrix S by recursively splitting points in k-dimensional space into a binary tree. The nearest neighbours of the query observation (ε) is then found by restricting the data space to the observations in the leaf node that the query observation belongs to. The kd-tree algorithm is particularly useful when k is relatively small; in our case k always remain limited, for example if one joint is constrained, and visual and tactile trajectories are defined, $k = 4$, while the number of samples in the exploration dataset can always satisfy $n \gg k$. The range search gives as result the column vectors containing the closest neighbours of the query point (ε) in the multimodal sensory matrix. The indices of those columns are then used to select the corresponding columns in the actuation primitives matrix. We denote the matrices obtained by selecting the indexed columns of S and A as $\tilde{\mathsf{S}}$ and $\tilde{\mathsf{A}}$, respectively. Sensory states contained in $\tilde{\mathsf{S}}$ can now be associated to the current state. A least square regression problem can be defined as follows:

$$\tilde{\mathsf{S}}\mathsf{w} = \varepsilon, \tag{8}$$

where w is a weighting vector. The solution of this equation gives the solution for the control problem in the task (sensory) space. The best approximate solution, also the minimum norm solution of equation (8), is given by $\mathsf{w} = \tilde{\mathsf{S}}^{\dagger}\varepsilon$, where $\tilde{\mathsf{S}}^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of the matrix $\tilde{\mathsf{S}}$. Since each column in $\tilde{\mathsf{S}}$ is directly related to a particular column in $\tilde{\mathsf{A}}$, the same vector w can be used to build new primitives as combinations of the primitives recorded during exploration:

$$\mathsf{v}^* = \tilde{\mathsf{A}}\mathsf{w}. \tag{9}$$

Equation (9) defines the desired motor command vector as a combination of the nearest motion primitives previously observed through the weight vector w. Note that the desired motor command vector v^* is found without requiring access to the Jacobian or any kinematic model of the robot.

3.3 Models Update

When a new control command m^* is applied, the new reached state x_r can be observed and used to calculate $\Delta\mathsf{x} = \mathsf{x}_\mathsf{r} - \mathsf{x}$, that is the state update caused by the new experienced motor commands. The new data m^* and $\Delta\mathsf{x}$ can then be added to the robot experience, that is to $\mathcal{M}$ and $\mathcal{X}$, respectively. If the error $\mathsf{e}_\mathsf{r} = \mathsf{x}^* - \mathsf{x}_\mathsf{r}$ obtained at the reached position exceeds a predefined tolerance threshold, then more exploration is required for the robot to refine its internal models.

In parallel to the execution of the calculated motor command m^*, an efferent copy of the same command is sent to the forward model, which performs an internal simulation of the action taken. The error $\mathsf{e}_\mathsf{p} = \mathsf{x}' - \mathsf{x}_\mathsf{r}$ between the prediction obtained by the forward model and the actual reached position is then evaluated. If the prediction error e_p exceeds a tolerance threshold, then the forward model needs to be refined.

Algorithm 1: Internal models update.

Initialise: $\kappa_\mathsf{r} = 0$, $\kappa_\mathsf{p} = 0$
Update :
if $\mathsf{C}_\mathsf{r} < \bar{\mathsf{c}}$ **then** $\kappa_\mathsf{r} = \kappa_\mathsf{r} + 1$
if $\mathsf{C}_\mathsf{p} < \bar{\mathsf{c}}$ **then** $\kappa_\mathsf{p} = \kappa_\mathsf{p} + 1$
Execute m^*
Update FM and IM
if $\kappa_\mathsf{r} \geq \bar{\kappa}$ *or* $\kappa_\mathsf{p} \geq \bar{\kappa}$ **then**
  Explore and update FM
  Set $\kappa_\mathsf{r} = 0$ and $\kappa_\mathsf{p} = 0$
  Update IM

In order to refine the internal models, more data points need to be acquired, either through some exploration steps or from other query points. The tracking error e_r and the prediction error e_p can be used to define confidence measures for the inverse and forward models, respectively. The confidence values, C_r and C_p for e_r and e_p respectively, can be calculated as a function of the corresponding errors. Note that, since the sensory data is normalised, then $\mathsf{x}, \mathsf{x}_\mathsf{r}, \mathsf{x}', \mathsf{x}^*$ take values in $[0, 1]$, and $\mathsf{e}_\mathsf{r}, \mathsf{e}_\mathsf{p} \in [-1, 1]$. We design the confidence as a normal distribution over the error, $\mathsf{C} \sim \mathcal{N}(\mu, \sigma^2)$, with $\mu = 0$ and $\sigma = 0.4$, so that when the error is close to zero, the confidence is approximately equal to 1 and when the error increases (symmetrically towards 1 or -1) the confidence values tend to zero. A threshold $\bar{\mathsf{c}}$ for the confidence is set to 0.6, corresponding to a deviation of approximately 1σ.

According to the confidence value, more exploration might be required. For every new query point, the inverse model produces a new motor command m^*, which can be passed as efferent copy to the forward model. After the execution of m^* and after the prediction x' is obtained, both the forward and the inverse model are updated with the new data point. Meanwhile, counters κ_r and κ_p are kept to check how many times the conditions $\mathsf{C}_\mathsf{r} > \bar{\mathsf{c}}$ and $\mathsf{C}_\mathsf{p} > \bar{\mathsf{c}}$ are violated. If κ_r or κ_p overstep a predetermined limit, $\bar{\kappa}$, then more data points are required to refine the internal models and exploration steps are triggered. The procedure is summarised in Algorithm 1.

The state predictions provided by the forward models can be used to support the robot behaviour during imitation by giving the robot an anticipation of the tracking performance. If both the forward and inverse models are accurate enough, then the forward model can represent a considerable support to the imitation behaviour. For this reason an important contribution to the overall behaviour performance is given by the prediction error and by the tracking error (see Fig. 3 for a schematic representation). In the preliminary implementation presented in this paper, these measurements are used to trigger model refinement, according to the procedure described in Algorithm 1.

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

4 Experimental Setup: Exploration and Data

As infants in their first months perform series of gross movements of variable speed and amplitude, which involve all parts of the body but lack distinctive sequencing [66], analogously a robot can explore its sensorimotor representations through self-generated movements. In this study, we have used an iCub humanoid robot and a MIDI keyboard as exploration environment. Our approach is based on a self-exploratory phase where forward models are learned. Pseudo-random control signals, also referred to as actuation primitives (see also [78]), are issued to the robot's arm joints to generate exploratory movements. In this work, the control signals are velocity commands $v(t)$, defined for each joint as

$$v_{j,i}(t) = \begin{cases} v_{j,i} & \text{if } t \in [t_{0_i}, t_{0_i} + {}^{D}/_{2}) \\ -v_{j,i} & \text{if } t \in [t_{0_i} + {}^{D}/_{2}, t_{0_i} + D] \end{cases} \tag{10}$$

where $v_{j,i}$ is the magnitude of the i-th primitive applied to joint j ($i \in \{1, 2, ..., n\}$, $j \in \{1, 2, ..., \mathsf{M}\}$), the value of which is sampled from a uniform distribution $v_{j,i} \sim \mathcal{U}(v_{\min,j}, v_{\max,j})$ in order to generate pseudo-random exploratory movements. The other parameters are the starting time t_{0_i} of the i-th primitive, and the duration D of the primitives. By adopting the same duration time D for all the primitives, and splitting in half each primitive movement, opposite movements are performed in the first and second part of the primitive execution (see Fig. 4).

Data from multiple modalities are acquired during the execution of the exploratory movements, including the joints positions from the motor encoders, the position of the hand in the vision field through one of the robot's eye cameras, the tactile information through the tactile sensors placed on the robot's skin, and the sound data from a MIDI keyboard.

Proprioception data: Proprioception information is acquired from the motor encoders. The positions $q_1, \ldots, q_{\mathsf{M}}$ of the M joints are acquired and normalised according to each joint's limits. A $\mathsf{M} \times n$ matrix

$$\mathsf{S_P} = \begin{bmatrix} \Delta_{q_1,1} & \cdots & \Delta_{q_1,n} \\ \vdots & \ldots & \vdots \\ \Delta_{q_\mathsf{M},1} & \cdots & \Delta_{q_\mathsf{M},n} \end{bmatrix}, \tag{11}$$

is then built, where $\Delta_{q_j,i} = q_j(t_{0_i} + {}^{D}/_{2}) - q_j(t_{0_i})$ denotes the relative position of joint j from the starting point of execution of primitive i.

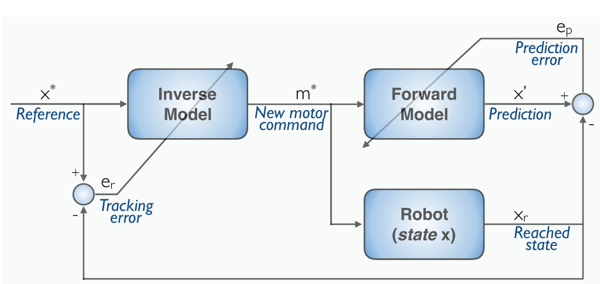


Figure 3: Learning architecture scheme: the inverse model is responsible for producing new motor commands m^* given a reference x^*. Motor commands are sent to the robot (characterised by the multi-modal state x), and, as an efferent copy, to the forward model, which produces a prediction x' of the next state. The prediction error ($\mathsf{e_p}$) and the tracking error ($\mathsf{e_r}$) are then computed and used to refine the internal models.

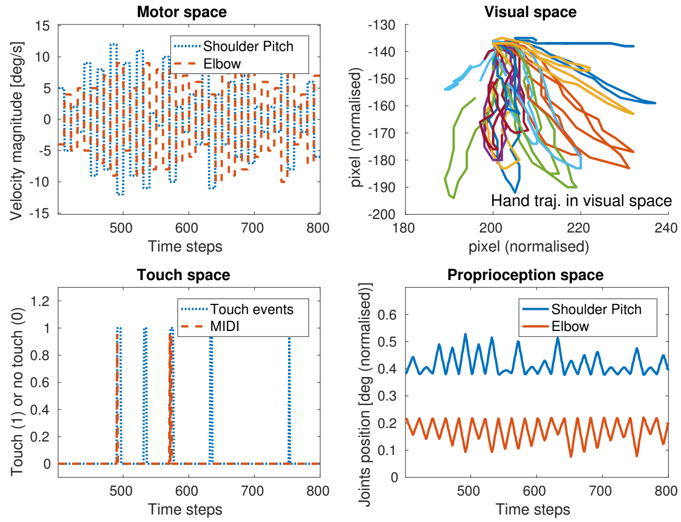


Figure 4: (Top-left) Velocity command signals for two of the actuators of the iCub arm (representative examples). Trajectories of the hand positions in the 2D image space (top-right - different colours correspond to different primitives), touch and MIDI feedback (bottom-left), and proprioception of the two degrees of freedom of the arm (bottom-right), during the execution of the generated primitives. This figure is best viewed in colour.

Vision data: One of the robot's eye cameras is used to acquire visual data. The position of the hand in the visual space is represented by the two-dimensional vector $[x, y]^T$ of the coordinates of the centre of the hand in the 2D image frames, computed as the average of the feature points detected by using the OpenCV optical flow algorithm [79], and then normalised according to the frame dimensions. A $2 \times n$ matrix

$$\mathsf{S_V} = \begin{bmatrix} \Delta_{x,1} & \cdots & \Delta_{x,n} \\ \Delta_{y,1} & \cdots & \Delta_{y,n} \end{bmatrix}, \tag{12}$$

is then built, where the relative displacements of the hand coordinates from the starting point of execution of primitive i is contained in $\Delta_{x,i} = x(t_{0_i} + {}^{D}/_{2}) - x(t_{0_i})$ and $\Delta_{y,i} = y(t_{0_i} + {}^{D}/_{2}) - y(t_{0_i})$.

Touch data: The iCub robot's skin consists of a network of tactile sensors (*taxels*), from which tactile information is recorded. In our experiments, we mainly focus on the hand skin, which contains 60 taxels, including the fingertips. For each taxel l of the hand ($l = \{1, 2, ..., 60\}$), a binarised pressure

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

output can be read. We then normalise each signal and calculate the average pressure sensed on the hand as $p = \frac{1}{60}\sum_{l=1}^{60} p_l$, A $1 \times n$ vector

$$\mathsf{S}_\mathsf{T} = \begin{bmatrix}\Delta_{p,1} & \cdots & \Delta_{p,n}\end{bmatrix} \tag{13}$$

is then built, where $\Delta_{p,i} = p_i(t_{0_i} + {}^{D}\!/_{2}) - p_i(t_{0_i})$ contains the tactile feedback (on/off) during the execution of primitive i.

Sound data: Sound information is acquired using a MIDI keyboard. MIDI is a symbolic representation of musical information incorporating both timing and velocity for each note played. In this work, we have used the information encoding the note played only, so that each key pressed is associated to a specific integer number. In Fig. 4 it is possible to note that only certain touch events are actually associated with a note. Similarly to the touch case, a single value is associated to each primitive execution, which is the code of the note if a note was played or zeros if not. A $1 \times n$ vector

$$\mathsf{S}_\mathsf{K} = \begin{bmatrix}s_1 & \cdots & s_n\end{bmatrix} \tag{14}$$

is then built, where $s_1, \ldots, s_n$ are normalised integer numbers coding the note played or zeros.

Example: Learning inverse models using vision, touch and proprioception: In the case vision, touch and proprioception are used, the multimodal sensory matrix and the imitation error vector are defined as

$$\mathsf{S} = \begin{bmatrix}\mathsf{S}_\mathsf{P}\\ \mathsf{S}_\mathsf{V}\\ \mathsf{S}_\mathsf{T}\end{bmatrix}^T, \quad \varepsilon = \begin{bmatrix}\varepsilon_\mathsf{P}\\ \varepsilon_\mathsf{V}\\ \varepsilon_\mathsf{T}\end{bmatrix}^T,$$

where $\varepsilon_\mathsf{P}, \varepsilon_\mathsf{V}, \varepsilon_\mathsf{T}$ are defined, respectively, as

$$\varepsilon_\mathsf{P} = \begin{bmatrix}q_n(t) - q_n^*(t)\end{bmatrix},\ \varepsilon_\mathsf{V} = \begin{bmatrix}x(t) - x^*(t)\\ y(t) - y^*(t)\end{bmatrix},\ \varepsilon_\mathsf{T} = |\tilde{p}(t) - \tilde{p}^*(t)|;$$

the couple $(x(t), y(t))$ represents the current hand position, and the couple $(x^*(t), y^*(t))$ the target position at time t; $p(t)$ denotes the pressure signal at time t, and $p^*(t)$ the target contact pressure at time t; η is the number of the joint (multiple joints can also be considered) on which a constraint (or reference) is given. Note that all the variables concerning the touch modality, that is both the state and the imitation error, are binary variables.

Experimental details: iCub and keyboard setup To demonstrate the proposed methodology we have used a humanoid iCub robot and a MIDI keyboard. Four of the robot's arm joints have been used for motor babbling and imitation, namely the shoulder pitch, roll and yaw, and the elbow.

The visual information has been extracted from feature points found by using the OpenCV optical flow algorithm on the image frames acquired from one of the robot's onboard 2D RGB cameras (with resolution 320×240 pixel). The position of the hand used to play the piano keyboard is computed as described in the previous paragraphs. A visual trajectory is similarly extracted when the demonstrator shows the notes execution. Representative trajectories are shown in Fig. 5. It is worth noting that the experimental setup, mostly concentrated on the piano keyboard and the moving hand (an example is shown in the top-left picture in Fig. 1), allows us to use the described strategy without being affected by problems caused for example by different amount of arm visible in the image, or moving background. Also, the point of view of the robot and of the teacher during the demonstrated execution is nearly the same (see Fig. 6), although perspective taking [80] could also be employed.

Proprioceptive references (joint angle data) could be acquired for example from motion capture systems, as in *e.g.* [59,60], or using more elaborate vision processing, *e.g.* [81], which are beyond the scope of this paper. For the purpose of demonstrating the effectiveness of our method, we let the application of these approaches as future work and input synthetic target trajectories to be imitated instead.

The tactile reference is also synthetically provided, that is it is not acquired from the human demonstrated but it is designed as a piece-wise constant reference. More specifically, the target tactile reference p^* is defined as $p^* = 1$ when a key should be hit, and $p^* = 0$ during transition movements. Regarding this modality, as a matter of fact, humans can not directly observe tactile sensations from others. On the other hand they are able to infer the tactile sensation from watching others. Although synthetic trajectories are used for the imitation experiments, using the multimodal sensory matrices of the inverse models, missing modalities can also be inferred by providing the available sensory information and searching for the nearest point in the multimodal space.

The sound information collected during the imitation task execution is compared with the demonstrator one, so to assess if the completion of the task was successful. We have used the integer numbers encoding each note, while the timing and the velocity of the musical information were ignored.

The exploration part consisted of 80 primitives execution, corresponding to 1600 data points, used to learn the forward model and build the actuation primitives matrix and multimodal sensory matrix. For the imitation part, 135 points were collected for the visual target trajectory demonstrated, corresponding to approximately 30 seconds demonstration.

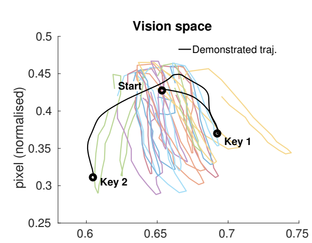


Figure 5: Visual trajectories: explored (coloured transparent lines), demonstrated (black).

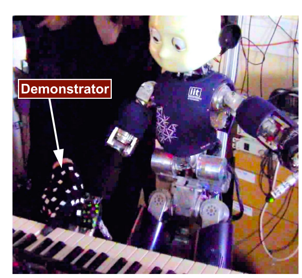


Figure 6: Demonstrating visual trajectories to the iCub robot.

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

5 Experiments and Discussion

We have demonstrated our approach to forward model learning and multimodal imitation learning on an iCub humanoid robot. The iCub first learns its sensorimotor representations models while interacting with a piano keyboard engaging vision, touch, proprioception and sound, while executing motor-babbling (see Fig. 1).

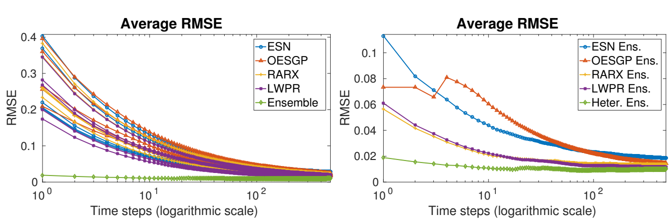
After exploration, a demonstrator shows the robot how to play a sequence of notes. The task assigned to the robot is to imitate the demonstrator execution based on the visual trajectory demonstrated. Touch and sound are fundamental in order to successfully play the piano keys. The task is multimodal, that is it forces constraints on different modalities, namely vision, touch and sound. In order to demonstrate the method including also the proprioception space, we add a constraint on proprioception by fixing one degree of freedom of the arm, so that the robot is forced to execute the imitation task without actually exploiting one of the arm's degrees of freedom. This constraint can also be seen as simulating a faulty joint: the robot is required to complete the task nonetheless, while its operational space is reduced.

We show that the robot is able first to make accurate predictions through the learned forward model, and second to leverage the multimodal data acquisition and the self-learned multimodal sensorimotor representations to complete the multimodal imitation task. The juxtaposition method used to build the multimodal matrices benefits multimodal imitation tasks by concurrently meeting requirements defined on different sensory data.

5.1 Forward Models Learning using Heterogeneous Online Ensemble

In this section we present the prediction performance of the proposed heterogeneous ensemble methods, showing that our method outperforms single base models and homogeneous ensembles in prediction accuracy.

In Fig. 7 the root mean squared error (RMSE) scores over the learning time steps are shown. The scores depicted in this figure are the average scores obtained over all the sensory modality dimensions, that is averaging the results obtained for the visual coordinates, the encoders and the touch data. The ensemble achieves the best accuracy, that is the lowest RMSE curve outperforming all the other single predictors as well as other homogeneous ensembles. Here, homogeneous ensembles have been obtained by applying the same proposed ensemble method on the 5 instances of models of the same type, thus obtaining 4 ensembles, one for each learning algorithm used.



(a) Comparison with all base models **(b)** Comparison with homogeneous ensembles

Figure 7: Average root mean squared error (RMSE) scores over iterations (logarithmic scale is used for the abscissal axis). The ensemble predictions are more accurate than predictions obtained using all other alternatives. The error also decreases monotonically as more data are acquired over time.

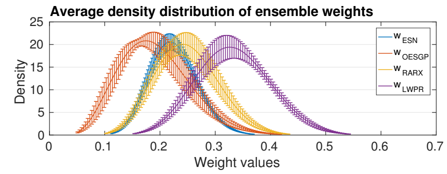
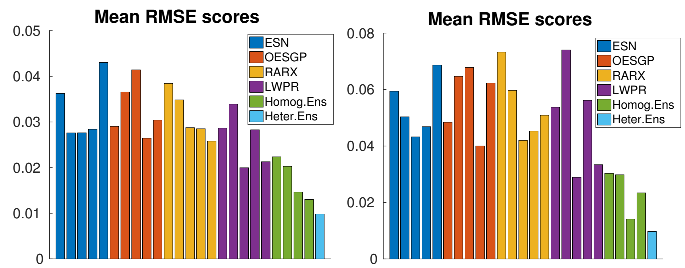


Figure 8: Density distribution of ensemble weights (representative examples). Since the weights change dynamically at every iteration, their temporal profile evolution is hardly readable. The distributions shown provide an overall view of the values of the ensemble weights over the whole learning period. Higher weights are assigned to the better performing base models.

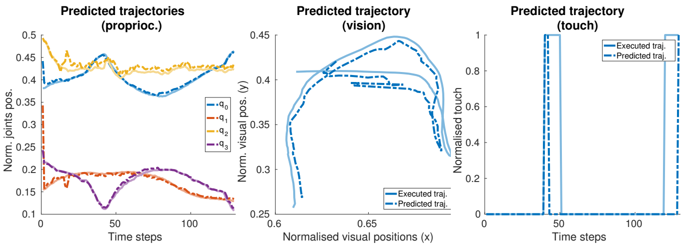


(a) Comparison with all base models on exploration (training) data **(b)** Comparison with all base models on imitation (test) data

Figure 9: (a) RMSE scores (average over different multimodal dimensions) obtained using single base modes (5 instances for each learning model), homogeneous ensembles (using the ensemble method on models of the same type: in the order ESN ensemble, OESGP ensemble, RARX ensemble and LWPR ensemble), and the proposed heterogeneous ensemble (light blue). The proposed heterogeneous ensemble scores the best accuracy compared to the alternatives. (b) RMSE scores on test data: the data for this test experiment consists of the trajectories executed during the imitation task on the piano keyboard (presented in section 5.2).

Fig. 8 shows the density distributions of the ensemble weights. This figure represents how the ensemble weights are distributed over the learning iterations. Recall that the ensemble weights change their values at each iteration, since the update takes into account not only the cumulative confidence of each base model, but also the actual performance at each time step. Since the weights change dynamically at every iteration, their temporal profile evolution is hardly readable. The distributions shown in Fig. 8 provide an overall view on the values of the ensemble weights over the whole learning period shown. Higher weights are assigned to the better performing base models. For example, it can be note that all the four algorithms contribute to the ensemble, as none of the distributions is concentrated around 0. On the other hand, a difference can be observed among the predictors. For example, the LWPR weight distribution is centred

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705



(a) Proprioception space **(b)** Vision space **(c)** Touch space

Figure 10: Predicted (dashed) and executed (solid) trajectories during the imitation task. The ensemble predictions are accurate on unforeseen data; in the touch space, predictions capture the overall behaviour but are less accurate due to the binary nature of this data.

at a higher value (approximately 0.35) compared to the other distributions; this translates into the fact that LWPR predictions become more relevant in the composition of the ensemble. The results in Fig. 8 also emerge in Fig. 9, where it can be noted that LWPR achieves the lowest RMSE score.

We compared our heterogeneous online ensemble with other ensemble combinations, as well as with the single base models. Results are shown in Fig. 9. The proposed heterogeneous ensemble scores the best accuracy compared to the alternatives not only on training data, that is on the data collected during self-exploration, but also on test datasets. Fig. 9b shows the RMSE scores on a test dataset consisting of the trajectories executed during the imitation task on the piano keyboard (presented in section 5.2). These results not only confirm the trend observed on training data, but also show that the learned forward models are able to generalise on novel uncharted gesture executions. Fig. 10 shows the predicted trajectories (dashed) in the proprioception, vision and touch spaces. Accurate results are achieved on all the multimodal dimensions. In the touch space, predictions capture the overall behaviour but the accuracy is lower than the results achieved on the other modality spaces due to the binary nature of this data.

We have observed that using multiple base model instead of a single one does not affect our results in terms of computational timing. At each time step, all the base models produce a prediction in less than 0.2 ms, which is under the rate used to control the robot.

5.2 Multimodal Imitation on Piano Keyboard

In this section we show the results obtained by using the proposed multimodal approach to learn inverse models in an imitation learning task.

The task designed for this experiment is to follow a demonstrated execution of a sequence of notes on the piano keyboard, exploiting the trajectory demonstrated on the vision space, while pressing the piano keys and without using one of the shoulder joints (constrained to remain fixed in a certain position). Note that the task can present some difficulties related to the extent to which the robot explored its sensorimotor system in the first exploration phase. This reflects in the number of times the robot touched the keyboard (for example in our experiments the keyboard was hit on 32 out of 80 executed primitives), in the extent of the region of the visual field where positions of the hand were registered, and, related to this, in the possibility that the demonstrated trajectory covers parts of the visual field that were not explored. Furthermore, due to the constraint forced on the proprioceptive space, the operational space of the robot's arm is effectively reduced: the robot needs to solve on-the-fly the imitation task using less degrees of freedom than the degrees available during the exploration.

Experimental results show that these issues can be effectively handled by our method. The search on the multimodal space, rather than on single modalities, plays a fundamental role. We have run a total of 50 repetitions of the experiment, where the iCub robot is required to imitate a demonstrator playing consequentially two notes. The success achieved, that is the successful execution of the two notes, was 45 over 50 (90%). In the failed attempts the pressure applied in order to play the piano keys was not sufficient, due to the fact that the tactile data acquired from the robot's fingertip were sometimes imprecise.

Fig. 11 shows the results of the multimodal imitation task for 10 repetitions of the task. It is possible to note that the robot aims at achieving a multimodal target: while following the demonstrated visual trajectory, it also moves in order to satisfy the touch modality requirement, that is actually touching the piano keys, and also trying to avoid moving the constraint joint.

Note also that the demonstration can include any number of keys, at different positions on the keyboard that is contained in the robot's visual field. Since no prior information is assumed, the information on the sensorimotor representations, the learned models, and the data used to learn them, have a notable impact on the imitation outcome; nonetheless, the experimental results show that the proposed method allows to effectively combine previous information to achieve points in the multimodal space.

Another important feature to take into consideration is the number of points defining the target trajectories (n_{target}). This reflects on the time taken to complete the imitation task and on the quality of the imitation: the more points are acquired, the more refined is the trajectory path, the more accurate is the tracking result, the slower the execution. The imitation obtained with increasing values of n_{target} results in more accurate outcomes.

Experimental results show that the range search is a robust solution in our scenario. The width of the range used for the search, denoted by a parameter r, effectively defines the number of column vectors used to build the matrices $\tilde{\mathsf{S}}$ and $\tilde{\mathsf{A}}$. It can be noted that increasing values of r might potentially cause higher computational complexity, since $\tilde{\mathsf{S}}$ must be inverted to find the weight vector w. However, in practice, the inversion of $\tilde{\mathsf{S}}$ is always easily computable, as the number of neighbours found remains limited. Unlike a k-nearest-neighbour search, with the range search it is possible to choose the maximum distance allowed from the query points, without the need of a prior designed specification of a certain number of neighbours, which in turn could include vectors that are actually far from the query point. Using the range search, the number of neigh-

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

bours varies for every query point. Notably, if no neighbours are found, this corresponds to a situation in which the robot has never experienced anything sufficiently similar to the query. A void search would then correspond to the robot staying still, as no motor commands could be chosen among the column of the matrix A. This behaviour should not be seen as a limitation, instead it reflects directly the influence of the previous experience on the imitation learning. Therefore, either a larger r is allowed, or more exploration should be performed. The first case would push the robot to try and combine the learned sensorimotor representations to reach for the target anyway. The second case would instead lead the robot to collect more data from self-exploration and thus refine the internal models. Also, because the number of neighbours in fact depends on the number of samples n collected in the exploration phase and the portion of the multimodal space actually explored, experimental results show that if r is chosen so that there exist neighbours almost at all times, the imitation performance does not improve sensibly by increasing r (see Fig. 12a).

The parameter n determines the duration of the exploration phase and the amount of sensorimotor samples gathered. In Fig. 12b the performance obtained by using different amount of exploration data, with $n_{target} = 135$, for 10 repetitions, is reported. It can be noted that a considerable improvement is achieved by increasing the number of primitives used from $n = 20$ to $n = 50$ especially in the vision and proprioception space, while the touch space seams instead to be less influenced. In particular, the movement performed using $n = 50$ is more precise (see Fig. 11). Also, it is possible to note that the most critical moment in the imitation can be identified around time step 50, when the first key is reached and then the movement to reach the second key starts. The temporal profile of the vision error in Fig. 11 presents a maximum at this point, also due to the constraint forced on the proprioception space: note the pick of the proprioception error in correspondence of the pick in the vision space.

Experiments have been carried out using different constraint setups in order to evaluate the behaviour of the robot on the imitation task when different constraints are forced on the proprioception space. By forcing a constant reference on the robot joints we aimed to simulate the case of a faulty joint mentioned in the Introduction. Qualitative results are reported in Fig. 13, where four cases are represented: the first is obtained when no constraints are forced, the second and third when forcing constraints to one joint at time (the shoulder pitch q_0 and the shoulder yaw q_2, respectively), and the fourth when two joints (both q_0 and q_2) are forced to remain fixed. These results show that despite the task becomes more difficult as more joints are constrained (the tactile space presents increasing shifts in time), the imitation task is successfully completed in all cases.

We have also compared our method for inverse modelling against two well-known algorithms that have been broadly used in the framework of model learning in robotics, namely artificial neural networks (ANN) and locally weighted projection regression (LWPR). We have compared the methods on the behaviours obtained in the imitation task consisting in imitating the visual trajectory to play two keys of the keyboard while satisfying the tactile constraint (no restrictions are imposed to the arm joints). We report in Table 1 the average mean squared error (MSE) and the standard deviation over 10 repetitions of the task. For these experiments, we have trained a network with 10 hidden layers and a LWPR model to learn the mapping between the motor commands and the three dimensional target space (consisting of the two visual coordinates and touch). The results obtained show that the proposed method achieves the smallest imitation errors on the target space.

One issue that can rise at different levels in the proposed architecture is how to deal with lack of information or incom-

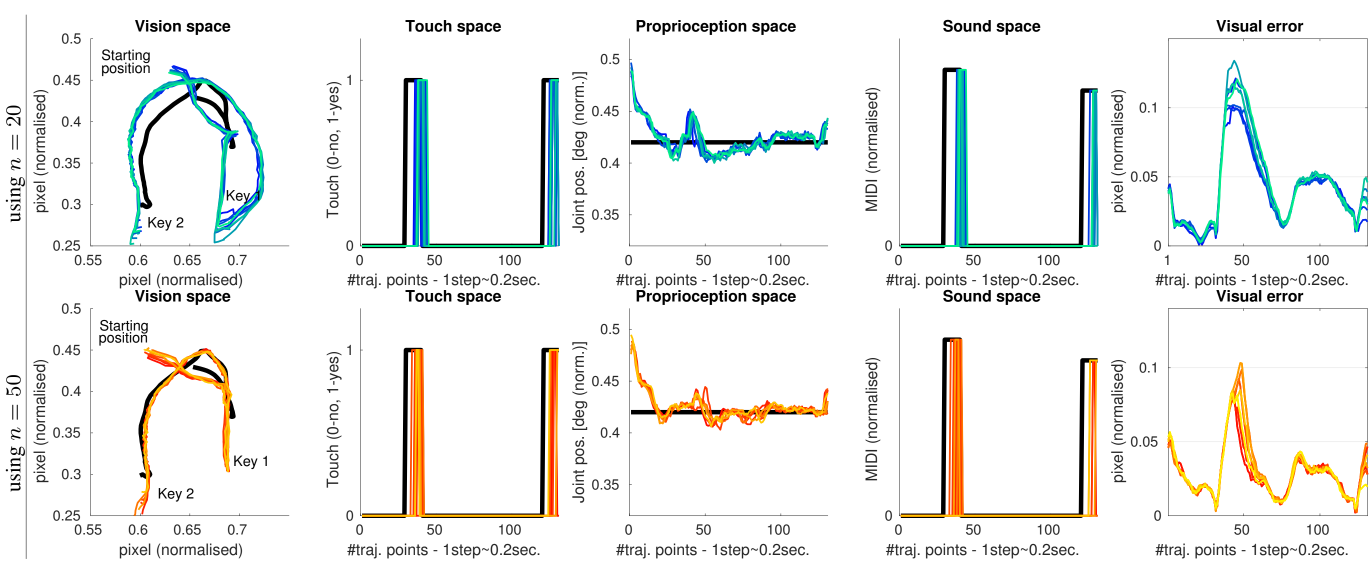


Figure 11: Multimodal imitation: qualitative results on the vision, touch, proprioception and sound spaces, using $n = 20$ (top row) and $n = 50$ (bottom row). A constraint on the shoulder yaw is forced, so that the robot should complete the imitation while keeping it fixed to a certain position. The temporal profile of the error, evaluated as the Euclidean distance of each point to the corresponding point in the target trajectory, is depicted in the right most pictures. The improvement in the performance achieved using more exploration steps can be noted especially in the vision and proprioception cases. All figures show the results of 10 repetitions. (Best viewed in colour).

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

Table 1: Comparison with ANN and LWPR: MSE scores and standard deviation obtained on the imitation task. Best results are shown in bold.

	Vision - x coord.	Vision - y coord.	Touch
ANN	0.0053 ± 0.0047	0.0116 ± 0.0064	0.55 ± 0.24
LWPR	0.0023 ± 0.0001	0.0087 ± 0.0022	0.48 ± 0.33
Proposed	$\mathbf{0.00037 \pm 0.00004}$	$\mathbf{0.00050 \pm 0.00017}$	$\mathbf{0.121 \pm 0.054}$

plete data. The hypothesis considered in our implementation is that the sampling times for all the different sensors are synchronised, that is all the information is gathered at a predefined rate (*i.e.* approximately 5 Hz), which allows us to align the data collected. Concerning the implementation of our inverse model used to achieve imitation, the juxtaposition of unimodal data allows us to include only the desired modality references while letting others free. This type of solution thus directly includes the possibility of dealing with incomplete information in terms of number of modalities considered for imitation. Finally, the problem of making predictions with missing data points is a well-known problem in many machine learning applications, that can be resolved, for example, by data completion methods. Solutions of this type could be included as future work in our framework.

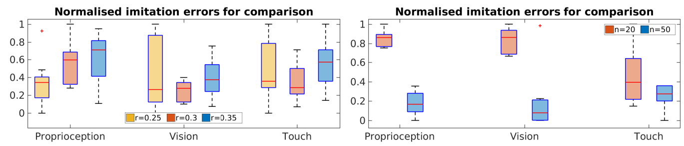
6 Conclusion and Future Work

In this study, we focused on the problem of learning accurate sensorimotor representations models in an online manner, following a developmental approach.

We have presented a learning architecture where forward and inverse models are coupled and updated as new data are available. No prior information on the robot kinematic structure is needed, nor an explicit formulation of the robot's kinematic and dynamic models. Taking inspiration from biological and neuroscientific studies, the proposed architecture is based on self-learned sensorimotor representations models, where the multimodal nature of the sensory system of the robot interacting with the environment is also taken into account.

We have proposed a heterogeneous online ensemble learning method which combines diverse parametric and non-parametric online algorithms. The sensorimotor representations models built through the ensemble learning process consist of predictors relating motor commands with effects on the robot sensory system. We have demonstrated that the proposed heterogeneous ensemble outperforms alternatives that include homogeneous ensembles and single base models, and achieves the best performance in generalising on novel uncharted datasets. The heterogeneous ensemble guarantees the necessary difference between the base models that allows improving the model prediction performance.

We have also presented a method to endow robots with multimodal learning skills enabling imitation learning. Our method, based on multimodal sensorimotor representations which are learned during exploratory actions, has been shown effective in performing on-the-fly multimodal imitation by combining the knowledge acquired during the multimodal learning steps.



(a) Effect of r **(b)** Effect of n

Figure 12: (a) Effect of r on imitation performance ($n_{target} = 135$) and (b) effect of n on imitation performance ($n_{target} = 135$ and $r = 0.3$). Normalisation is applied to the error measures for better comparison. Each modality is normalised separately in order to take into account different scales (results obtained from 10 repetitions).

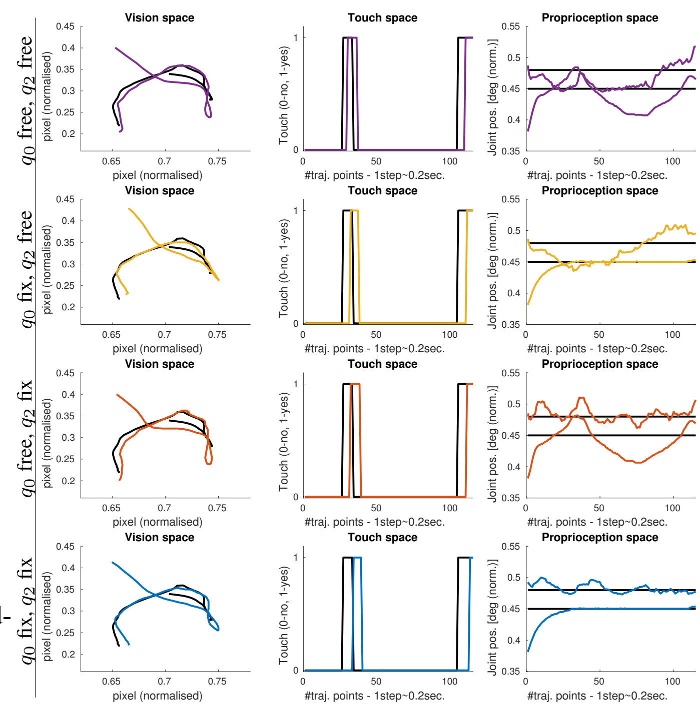


Figure 13: Comparison between behaviours obtained forcing different constraints on the proprioception space. These pictures show that the imitation task is achieved when no constraints are forced (first row, purple lines), as well as when forcing constraints to the shoulder pitch q_0 (second row, yellow lines), to the shoulder yaw q_2 (third row, orange lines), and to both (fourth row, blue lines). Proprioception trajectories present significant changes. Although the results on the tactile space present increasing delays when more joints are constrained, the imitation task is fulfilled in all cases.

Our approach to multimodal imitation benefits from learning sensorimotor representations using data from multiple sensors from self-exploration. The multimodal sensory matrices and the range search on a multimodal space are key aspects of the proposed method, that allow us to achieve successful imitation of multimodal tasks.

The formulation of the proposed method is general and allows to accommodate different modalities. We have demonstrated our method on an iCub humanoid robot, but since no *a priori* knowledge has been assumed on the kinematic and dynamic models of the robot, the proposed method can be applied to different robotic platforms.

Finally, it is interesting to note that the proposed framework presents some points of contact with the paradigm of predictive coding [82], such as the key roles played by the prediction error and the actions.

Future extensions of the presented work include demonstrating the method on different robots. Curiosity driven behaviours [83] could be implemented in order to enhance the exploration strategy to acquire new multimodal data for updating and refining the learned internal models.

Acknowledgements: This work was supported in part by the EU FP7 project WYSIWYD under Grant 612139.

References

[1] M. Kawato, "Internal models for motor control and trajectory planning," *Current opinion in neurobiology*, vol. 9, no. 6, pp. 718–727, 1999.

[2] M. Haruno, D. M. Wolpert, and M. Kawato, "Mosaic model for sensorimotor learning and control," *Neural computation*, vol. 13, no. 10, pp. 2201–2220, 2001.

[3] Y. Demiris and B. Khadhouri, "Hierarchical attentive multiple models for execution and recognition of actions," *Robotics and autonomous systems*, vol. 54, no. 5, pp. 361–369, 2006.

[4] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, pp. 503–507, 2015.

[5] A. N. Meltzoff and M. K. Moore, "Explaining facial imitation: A theoretical model," *Early Development & Parenting*, vol. 6, no. 3-4, p. 179, 1997.

[6] Y. Demiris and A. Meltzoff, "The robot in the crib: A developmental analysis of imitation skills in infants and robots," *Infant and Child Development*, vol. 17, no. 1, pp. 43–53, 2008.

[7] A. Maye and A. Engel, "Using sensorimotor contingencies for prediction and action planning," in *From Animals to Animats 12*. Springer Berlin Heidelberg, 2012, vol. 7426, pp. 106–116.

[8] A. Maye and A. K. Engel, "A discrete computational model of sensorimotor contingencies for object perception and control of behavior," in *IEEE International Conference on Robotics and Automation* . IEEE, 2011, pp. 3810–3815.

[9] R. Miall and D. M. Wolpert, "Forward models for physiological motor control," *Neural networks*, vol. 9, no. 8, pp. 1265–1279, 1996.

[10] A. Dearden and Y. Demiris, "From exploration to imitation: using learnt internal models to imitate others," in *4th International Symposium on Imitation in Animals and Artifacts*, 2007, pp. 5–13.

[11] Y. Yoshikawa, H. Kawanishi, M. Asada, and K. Hosoda, "Body scheme acquisition by cross modal map learning among tactile, visual, and proprioceptive spaces," vol. 94, pp. 181–184, 2002.

[12] G. Schillaci, "Sensorimotor learning and simulation of experience as a basis for the development of cognition in robotics," Ph.D. dissertation, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, 2014.

[13] G. Schillaci, B. Lara, and V. V. Hafner, "Internal simulations for behaviour selection and recognition," in *Human behavior understanding*. Springer, 2012, pp. 148–160.

[14] M. Zambelli and Y. Demiris, "Multimodal imitaiton using self-learned sensorimotor representations," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2016, p. to appear.

[15] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, "The iCub humanoid robot: an open-systems platform for research in cognitive development." *Neural networks : the official journal of the International Neural Network Society*, vol. 23, no. 8-9, pp. 1125–34, Jan. 2010.

[16] L. Jamone, L. Natale, F. Nori, G. Metta, and G. Sandini, "Autonomous online learning of reaching behavior in a humanoid robot," *International Journal of Humanoid Robotics*, vol. 9, no. 03, 2012.

[17] M. Hersch, E. Sauser, and A. Billard, "Online learning of the body schema," *International Journal of Humanoid Robotics*, vol. 5, no. 02, pp. 161–181, 2008.

[18] L. Jamone, B. Damas, J. Santos-Victor, and A. Takanishi, "Online learning of humanoid robot kinematics under switching tools contexts," in *IEEE International Conference on Robotics and Automation* . IEEE, 2013, pp. 4811–4817.

[19] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.

[20] G. Schillaci, V. V. Hafner, and B. Lara, "Online learning of visuo-motor coordination in a humanoid robot. a biologically inspired model," in *Joint IEEE International Conferences on Development and Learning and Epigenetic Robotics (ICDL-Epirob)*. IEEE, 2014, pp. 130–136.

[21] C. Von Hofsten, "An action perspective on motor development," *Trends in cognitive sciences*, vol. 8, no. 6, pp. 266–272, 2004.

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

[23] L. Jamone, L. Natale, K. Hashimoto, G. Sandini, and A. Takanishi, "Learning task space control through goal directed exploration," in *2011 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2011, pp. 702–708.

[24] D. Nyga, F. Balint-Benczedi, and M. Beetz, "PR2 looking at things - ensemble learning for unstructured information processing with markov logic networks," in *IEEE International Conference on Robotics and Automation* . IEEE, 2014, pp. 3916–3923.

[25] S. Heinrich, P. Folleher, P. Springstübe, E. Strahl, J. Twiefel, C. Weber, and S. Wermter, "Object learning with natural language in a distributed intelligent system: A case study of human-robot interaction," in *Foundations and Practical Applications of Cognitive Systems and Information Processing*. Springer, 2014, pp. 811–819.

[26] G. Brown, "Ensemble learning," in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 312–320.

[27] G. Valentini and F. Masulli, "Ensembles of learning machines," in *Neural Nets*. Springer, 2002, pp. 3–20.

[28] J. Mendes-Moreira, C. Soares, A. Jorge, and J. D. Sousa, "Ensemble approaches for regression: A survey," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, p. 10, 2012.

[29] E. Ikonomovska, J. Gama, and S. Džeroski, "Online tree-based ensembles and option trees for regression on evolving data streams," *Neurocomputing*, vol. 150, Part B, pp. 458 – 470, 2015.

[30] S. G. Soares and R. Araújo, "A dynamic and on-line ensemble regression for changing environments," *Expert Systems with Applications*, vol. 42, no. 6, pp. 2935 – 2948, 2015.

[31] M. I. Jordan and D. E. Rumelhart, "Forward models: Supervised learning with a distal teacher," *Cognitive science*, vol. 16, no. 3, pp. 307–354, 1992.

[32] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1. IEEE, 2001, pp. 298–303.

[33] B. Damas, L. Jamone, and J. Santos-Victor, "Open and closed-loop task space trajectory control of redundant robots using learned models," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 163–169.

[34] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer handbook of robotics*. Springer, 2008, pp. 1371–1394.

[35] A. Billard, "Learning motor skills by imitation: a biologically inspired robotic model," *Cybernetics & Systems*, vol. 32, no. 1-2, pp. 155–193, 2001.

[36] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.

[37] K. Dautenhahn and C. L. Nehaniv, *Imitation in animals and artifacts*. MIT Press Cambridge, MA, 2002.

[38] G. M. Hayes and J. Demiris, *A robot controller using learning by imitation*. University of Edinburgh, Department of Artificial Intelligence, 1994.

[39] D. Korkinof and Y. Demiris, "Online quantum mixture regression for trajectory learning by demonstration," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3222–3229.

[40] S. P. Chatzis, D. Korkinof, and Y. Demiris, "A quantum-statistical approach toward robot learning by demonstration," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1371–1381, 2012.

[41] ——, "A nonparametric bayesian approach toward robot learning by demonstration," *Robotics and Autonomous Systems*, vol. 60, no. 6, pp. 789–802, 2012.

[42] H. Soh and Y. Demiris, "When and how to help: An iterative probabilistic model for learning assistance by demonstration," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3230–3236.

[43] ——, "Spatio-temporal learning with the online finite and infinite echo-state gaussian processes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 522–536, 2015.

[44] Y. Wu, Y. Su, and Y. Demiris, "A morphable template framework for robot learning by demonstration: Integrating one-shot and incremental learning approaches," *Robotics and Autonomous Systems*, vol. 62, no. 10, pp. 1517–1530, 2014.

[45] Y. Wu and Y. Demiris, "Towards one shot learning by imitation for humanoid robots," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2889–2894.

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

[46] ——, “Hierarchical learning approach for one-shot action imitation in humanoid robots,” in *2010 11th International Conference on Control Automation Robotics & Vision* . IEEE, 2010, pp. 453–458.

[47] ——, “Efficient template-based path imitation by invariant feature mapping,” in *2009 IEEE International Conference on Robotics and Biomimetics* . IEEE, 2009, pp. 913–918.

[48] K. Lee, Y. Su, T.-K. Kim, and Y. Demiris, “A syntactic approach to robot imitation learning using probabilistic activity grammars,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1323–1334, 2013.

[49] K. Lee, T.-K. Kim, and Y. Demiris, “Learning reusable task components using hierarchical activity grammars with uncertainties,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1994–1999.

[50] J. Kober, J. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey.” The International Journal of Robotics Research, 2013, vol. 32, pp. 1238–1274.

[51] F. Stulp and O. Sigaud, “Robot skill learning: From reinforcement learning to evolution strategies,” *Paladyn, Journal of Behavioral Robotics*, vol. 4, no. 1, pp. 49–61, 2013.

[52] P. Kormushev, S. Calinon, and D. G. Caldwell, “Reinforcement learning in robotics: Applications and real-world challenges,” *Robotics*, vol. 2, no. 3, pp. 122–148, 2013.

[53] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, “Gaussian processes for data-efficient learning in robotics and control,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 408–423, 2015.

[54] S. Levine and P. Abbeel, “Learning neural network policies with guided policy search under unknown dynamics,” in *Advances in Neural Information Processing Systems*, 2014, pp. 1071–1079.

[55] S. Levine, N. Wagener, and P. Abbeel, “Learning contact-rich manipulation skills with guided policy search,” in *2015 IEEE International Conference on Robotics and Automation*. IEEE, 2015, pp. 156–163.

[56] I. Mordatch, N. Mishra, C. Eppner, and P. Abbeel, “Combining model-based policy search with online model learning for control of physical humanoids,” in *IEEE International Conference on Robotics and Automation*, 2016.

[57] Y. Demiris and A. Dearden, “From motor babbling to hierarchical learning by imitation: a robot developmental pathway,” July 2005.

[58] M. Lopes and J. Santos-Victor, “Visual Learning by Imitation With Motor Representations,” *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 3, pp. 438–449, June 2005.

[59] A. Tidemann, P. Öztürk, and Y. Demiris, “A groovy virtual drumming agent,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5773 LNAI, pp. 104–117, 2009.

[60] A. M. Schmidts, D. Lee, and A. Peer, “Imitation learning of human grasping skills from motion and force data,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Sept. 2011, pp. 1002–1007.

[61] M. Fadlil, K. Ikeda, K. Abe, T. Nakamura, and T. Nagai, “Integrated concept of objects and human motions based on multi-layered multimodal LDA,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Nov. 2013, pp. 2256–2263.

[62] T. Araki, T. Nakamura, T. Nagai, K. Funakoshi, M. Nakano, and N. Iwahashi, “Autonomous acquisition of multimodal information for online object concept formation by robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Sept. 2011, pp. 1540–1547.

[63] A. M. Batula, M. Colacot, D. K. Grunberg, and Y. E. Kim, “Using Audio and Haptic Feedback to Detect Errors in Humanoid Musical Performances,” *Proceedings of the International Conference on New Interfaces for Musical Expression*, vol. 2, pp. 295–300, 2013.

[64] M. Johnsson and C. Balkenius, “Sense of Touch in Robots With Self-Organizing Maps,” *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 498–507, June 2011.

[65] K. Mochizuki, S. Nishide, H. G. Okuno, and T. Ogata, “Developmental human-robot imitation learning of drawing with a neuro dynamical system,” in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2013, pp. 2336–2341.

[66] M. Hadders-Algra, “The neuronal group selection theory: a framework to explain variation in normal motor development,” *Developmental Medicine & Child Neurology*, vol. 42, no. 8, pp. 566–572, 2000.

[67] Y. Demiris, L. Aziz-Zadeh, and J. Bonaiuto, “Information processing in the mirror neuron system in primates and machines,” *Neuroinformatics*, vol. 12, no. 1, pp. 63–91, 2014.

[68] D. M. Wolpert and M. Kawato, “Multiple paired forward and inverse models for motor control,” *Neural networks*, vol. 11, no. 7, pp. 1317–1329, 1998.

[69] L. I. Kuncheva and C. J. Whitaker, “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy,” *Machine learning*, vol. 51, no. 2, pp. 181–207, 2003.

Preprint version; final version available at http://ieeexplore.ieee.org/document/7731167/
IEEE Transactions on Cognitive and Developmental Systems (2016)
DOI: 10.1109/TCDS.2016.2624705

[70] L. Breiman, "Stacked regressions," *Machine learning*, vol. 24, no. 1, pp. 49–64, 1996.

[71] N. Rooney, D. Patterson, S. Anand, and A. Tsymbal, "Dynamic integration of regression models," in *Multiple Classifier Systems*. Springer, 2004, pp. 164–173.

[72] H. Jaeger, "Adaptive nonlinear system identification with echo state networks," in *Advances in neural information processing systems*, 2002, pp. 593–600.

[73] H. Soh and Y. Demiris, "Spatio-temporal learning with the Online Finite and Infinite Echo-state Gaussian processes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 522–536, 2015.

[74] S. Vijayakumar and S. Schaal, "Fast and efficient incremental learning for high-dimensional movement systems," in *2000 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 2000, pp. 1894–1899.

[75] L. Ljung, Ed., *System Identification: Theory for the User (2nd Ed.)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.

[76] L. Ljung and T. Söderström, *Theory and practice of recursive identification*. MIT press Cambridge, MA, 1983.

[77] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, "Orthogonal range searching," *Computational Geometry: Algorithms and Applications*, pp. 95–120, 2008.

[78] P. Kormushev, Y. Demiris, and D. G. Caldwell, "Encoderless position control of a two-link robot manipulator," in *IEEE International Conference on Robotics and Automation*. IEEE, May 2015, pp. 943–949.

[79] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[80] T. Fischer and Y. Demiris, "Markerless perspective taking for humanoid robots in unconstrained environments," in *IEEE International Conference on Robotics and Automation*. IEEE, 2016, pp. 3309–3316.

[81] H. J. Chang and Y. Demiris, "Unsupervised Learning of Complex Articulated Kinematic Structures combining Motion and Skeleton Information," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[82] A. Clark, "Embodied prediction," in *Open MIND*. Open MIND. Frankfurt am Main: MIND Group, 2015.

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

Martina Zambelli received her B.Sc. degree in Information Engineering in 2011 and her M.Sc. degree in Automation Engineering in 2013, both from the University of Padova, Italy, and developed her master thesis project in the Automatic Control department at KTH - Royal Institute of Technology, Stockholm, Sweden, in 2013. She is currently a Ph.D. student in the Personal Robotics Lab at the Electrical and Electronic Department, Imperial College London, since January 2014, under the supervision of Prof. Yiannis Demiris. Her research interests involve autonomous robotics, artificial cognitive systems, human-robot interaction, autonomous system, with particular focus on learning methods for humanoid robots.

Yiannis Demiris (SM03) received the B.Sc. (Hons.) and Ph.D. degrees from the Department of Artificial Intelligence, University of Edinburgh, Edinburgh, U.K. He is a Professor with the Department of Electrical and Electronic Engineering at Imperial College London, where he heads the Personal Robotics Laboratory. His current research interests include human robot interaction, machine learning, user modeling, and assistive robotics; he has published more than 150 journal and peer reviewed conference papers on these topics. Professor Demiris was the Chair of the IEEE International Conference on Development and Learning in 2007 and the Program Chair of the ACM/IEEE International Conference on Human Robot Interaction in 2008. He was a recipient of the Rectors Award for Teaching Excellence in 2012 and the FoE Award for Excellence in Engineering Education in 2012. He is a senior member of the IEEE, and a Fellow of the IET, BCS and the Royal Statistical Society.