Online Ensemble Learning of Sensorimotor Contingencies

Martina Zambelli* and Yiannis Demiris*

***Abstract*— Forward models play a key role in cognitive agents by providing predictions of the sensory consequences of motor commands, also known as sensorimotor contingencies (SMCs). In continuously evolving environments, the ability to anticipate is fundamental in distinguishing cognitive from reactive agents, and it is particularly relevant for autonomous robots, that must be able to adapt their models in an online manner. Online learning skills, high accuracy of the forward models and multiple-step-ahead predictions are needed to enhance the robots' anticipation capabilities. We propose an online heterogeneous ensemble learning method for building accurate forward models of SMCs relating motor commands to effects in robots' sensorimotor system, in particular considering proprioception and vision. Our method achieves up to 98% higher accuracy both in short and long term predictions, compared to single predictors and other online and offline homogeneous ensembles. This method is validated on two different humanoid robots, namely the iCub and the Baxter.**

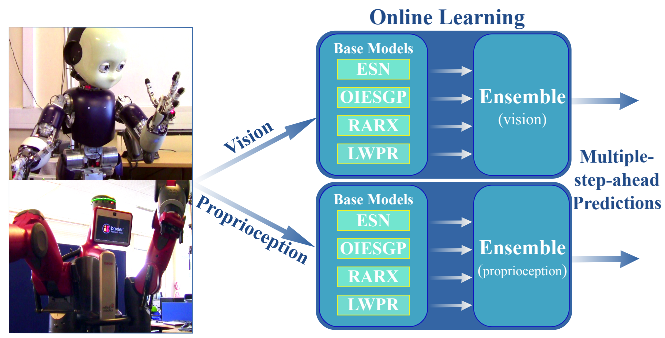


Fig. 1: Online Ensemble Learning of SMCs from Motor Babbling on the iCub (top) and the Baxter (bottom) for Multiple-Step-Ahead Predictions on proprioception and vision data.

I. Introduction

Complex robots rely on internal models describing the kinematics and dynamics for controlling and planning actions; however, constructing these models by hand can be costly. This motivates the interest in empowering robots with learning capabilities, in order to enable them to build their internal models through a learning process in which relations between actions and associated changes in sensory input, also known as sensorimotor contingencies [1], are involved. Robots benefit from being aware of their motion capabilities: self awareness allows the development of autonomous behaviours and the formulation of decisions, while providing a better understanding of the environment as well as other agents. This autonomous learning ability also simplifies the programming work, enabling automatic recovery from failures or morphological changes, and eliminating the need for explicit model formation while dealing with model drifts.

The approach we follow for learning internal models takes inspiration from neuroscientific studies arguing that infants use self-exploration and self-stimulation to "calibrate" their sensorimotor and body representations [2]. Analogously, a robot can explore its sensorimotor capabilities through self-exploration, or *motor babbling* [3]. The role of actions is constitutive in this learning process. No internal representations of the world are hand-crafted to generate sensory awareness. Instead, robots can learn internal models to predict SMCs, given the current sensory states and the motor commands; these are also known as *forward models* and have been employed to build biologically inspired control architectures [4–6]. The scientific challenge is devising algorithms that will extend the horizon of these predictions to more than a single time step. Enabling robots to perform long-term (multiple-step-ahead) highly accurate predictions is needed to enhance anticipatory capabilities.

Motivated by these observations, we focus on the one hand on multiple-step-ahead online learning and on the other hand on improving the accuracy of the predictions. In this paper, the focus is directed to learning forward models of SMCs which consist of mappings that relate motor commands to effects on two different sensory modalities, namely proprioception and vision. Our main contribution is an effective online ensemble learning method to learning forward models of sensorimotor contingencies. Our method, based on an ensemble of parametric and non-parametric predictors, is effective and accurate in building forward models of sensorimotor contingencies that relate motor commands issued to the robot arm joints during babbling and the effect on the position of the limb both in proprioceptive and vision space. The proposed model achieves 20-98% better performance compared to state-of-the-art alternatives like offline and online homogeneous ensembles. Accurate predictions are also achieved by the proposed method in generalising on gestures, such as waving and pointing.

Two different robots were involved in the experiments: the iCub and the Baxter (in Fig. 1), thus showing that our method is not designed *ad-hoc* for a particular robot, but can instead be employed with different robots and kinematic structures.

II. Related Work

In this paper, we aim to achieve four key aspects in learning sensorimotor contingencies models: online learning, long-term anticipation, accuracy of predictions, and multi-modal learning, exploiting robots' proprioception and vision. These aspects are discussed in the rest of the section.

a) Online learning: Online learning has been employed in different studies to tackle forward model learning in

*M. Zambelli and Y. Demiris are with the Dept. of Electrical and Electronic Engineering, Imperial College London, UK. E-mail: {m.zambelli13, y.demiris}@imperial.ac.uk. This research was funded in part by the EU FP7 project WYSIWYD under Grant 612139.

robotics. In [7], an online learning scheme was developed using a mixture of recurrent neural networks. However, only one-step-ahead predictions were performed. Studies on mobile robots addressed online learning while adopting multiple modalities, in particular proprioception and vision, *e.g.* in [8–10]. Online learning has been employed also in humanoid robotics: in [11] online learning has been used to achieve reaching behaviour in a humanoid robot, while in [12] an online strategy has been implemented to learn the kinematic structure of a humanoid robot. However, these studies are either goal/task-directed or aim at identifying parameters of forward kinematics to retrieve rotation matrices. Our approach is task-independent, can generalise to different types of movement, and aims to learn accurate lower level mappings that can be integrated effectively within a full control architecture.

b) Long-term anticipation: Multiple-step-ahead predictions can be obtained by chaining multiple single-step-ahead predictors, that is by iterating one-step-ahead predictions. Several authors proposed this iterative strategy (*e.g.* [7–9]), showing that agents that are able to anticipate long-term sensory consequences of their movements or actions behave in a more effective and human-like manner compared with reactive agents.

c) Accuracy of predictions: To improve prediction accuracy, the ensemble learning method [13,14] has successfully been used in a wide variety of machine learning tasks. We decided therefore to apply this approach to the problem of learning accurate forward models of sensorimotor contingencies for humanoid robots. Several approaches exist to solve offline regression problems [15]. However, online ensemble learning algorithms for regression have received less attention.

d) Multimodal learning: While proprioceptive information is directly obtained from the joints encoders, different approaches can instead be considered to acquire visual information. Recent works have used optical flow to build kinematic model for robots, *e.g.* [16] and [17]. In contrast with those works, where a depth 3D image is acquired from external cameras, we exploit 2D images captured from the cameras placed in the eyes/head of the humanoid robots. Although this choice drastically limits the visual field, while increasing the possibility of occlusions, we rely on the robots' onboard cameras only, favouring the autonomy of the robots which do not have to rely on external data sources.

III. Methodology

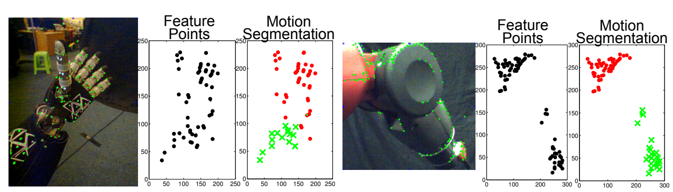
A. The Data: Proprioception and Visual Information

Using self-exploration, also known as *motor babbling* [3], the robots learn their sensorimotor contingencies, in particular the mappings between the commands issued to the joint motors and the effect in their sensorimotor perception. Ensemble predictor models are learnt in an online manner, as new information from the robots' sensorimotor systems are available. All the instances of the dataset are learnt separately, so that multiple models are built. To test the generalisation capabilities of the SMCs models learnt, we use two different datasets collected while robots perform waving and pointing gestures. These datasets might contain unexplored positions, present structured data for proprioceptive or visual trajectories (*e.g.* regular oscillations, flat trajectories), and have a longer duration to complete.

Motor babbling has been realised by issuing velocity commands to the robots arm joints. We recorded two sensory modalities to capture the positions reached by the limb, namely proprioception and vision. Positions are recorded in joint space by reading joints' encoders values in degrees ($[deg]$) and radians ($[rad]$) for the iCub and the Baxter, respectively; in the vision space, positions are measured in pixels ($[pxl]$), while OpenCV is used to analyse and extrapolate information from the acquired camera frames. In particular, for each frame we collect 2D feature points that can be tracked as part of the moving robot limb. A segmentation algorithm proposed in [18] is applied to identify parts of limbs in the visual frames. This algorithm is able to automatically cluster parts of the image according to motion, so that feature points moving together are clustered as one limb part. The frame size of the images acquired from the robot cameras are 320×240 $[pxl]$ and 480×300 $[pxl]$ for the iCub and Baxter, respectively. In our experiments, independent movements were identified for the forearm and hand of the iCub, and for the upper and lower arm of the Baxter, as shown in Figure 2. The position of the limb in the vision space is computed as the position in the 2D image space of the centre of the clusters identified in the visual frames. This approach is thus robust against differences in shape, dimension, specific morphological characteristics.

The data sample at time t of the dataset is represented as $\langle(\mathbf{x}, y)\rangle(t)$. In joint space the input point $\mathbf{x}$ is the vector $[v_j(t), p_j(t)]^T$ of the current velocity command v_j applied to the joint j and the current position p_j of the joint $j = 1, \ldots, N_J$, where N_J is the number of joints, while $y(t) = p_j(t + \Delta T)$ is the output consisting in the future position at time $t + \Delta T$, with $\Delta T > 0$, in the joint space. Although the model to learn in the proprioceptive space, representing the relationships between velocities and positions both in joint space, might appear simple, the underlying learning process is general, makes no *a priori* assumptions and can be applied to learn SMCs involving different modalities, such as vision.

In vision space the input point $\mathbf{x}$ is the vector $[\mathbf{v}(t), \mathbf{p}_c(t)]^T$ of the current velocity command vector $\mathbf{v} = [v_1, ..., v_{N_J}]$ issued to the arm joints, and the current position



(a) iCub (b) Baxter

Fig. 2: Visual Information. OpenCV feature detection and segmentation from motion are performed on the frames recorded from the eye/head cameras of the iCub and of the Baxter (left and right side, respectively). Clusters identifying limb parts are represented.

$\mathbf{p}_c$ of the cluster $c = 1, \dots, N_C$, where N_C is the number of limb parts identified as clusters in the image frames, while $y(t) = \mathbf{p}_c(t + \Delta T)$ is the output consisting in the future position at time $t + \Delta T$, with $\Delta T > 0$, of the limb part in the vision space.

The models are trained for few minutes (1∼3 minutes). Then, the built models are tested on new data obtained by issuing motor commands and recording the position of the limb in joint and vision space, to evaluate the predictors performance; these new data constitute the test datasets for the models.

B. Review of Ensemble Learning

Ensemble learning methods are based on few steps: ensemble generation, pruning, ensemble integration.

The generation of base models is referred to as *ensemble generation*. The objective is to build a set of M base models, also called *pool of models* $\mathcal{F}_M = \{\hat{f}_m,\ m = 1, 2, \dots, M\}$, to approximate a true function f. If the models in $\mathcal{F}_M$ are all generated using the same induction algorithm, the ensemble is called *homogeneous*, while if more than one algorithm is used to build $\mathcal{F}_M$, the ensemble is *heterogeneous*. Less work exists on heterogeneous ensembles than in homogeneous ones [15]; however, combining different algorithms is a promising strategy to obtain diversity, which has been shown of great importance in enhancing prediction accuracy [19].

Pruning strategies are frequently applied to improve the ensemble performance in terms of accuracy, in addition to reducing computational costs. Pruning simply consists of selecting a subset $\mathcal{F}$ from the pool of models: $\mathcal{F} \subseteq \mathcal{F}_M$. Several pruning strategies exist that are based on the performance of the base models [15], so that bad predictors can be excluded from the ensemble.

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: $\hat{f}_{\mathcal{F}} = \sum_{m=1}^{M} w_m \hat{f}_m(\mathbf{x})$, where $\mathbf{x}$ is the data sample and $w_m \in [0, 1]$, $\sum_{m=1}^{M} w_m$=1, are the weights assigned to each base model $\hat{f}_m$. The weights state the importance of the single base models in building the ensemble, according to some criterion. The weights can be constant or dynamically calculated according to each data sample. Popular algorithms to obtain ensemble weights are stacked regression [20] and dynamic weighting [21]. Given a learning set $\mathcal{L}$ with K examples, the stacked regression approach calculate the weights by minimising $\sum_{k=1}^{K}\big[f(\mathbf{x}_k) - \sum_{m=1}^{M} w_m \hat{f}_m(\mathbf{x}_k)\big]^2$, while the dynamic weighting method sets the weights according to some performance measurement of the predictors.

C. Ensemble Learning and Multiple-Step-Ahead Predictions

1) Online 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: (1) the Echo State Networks (ESN) [22], which are a class of recurrent neural networks; (2) the Online Infinite Echo State Gaussian Processes (OIESGPs) [23], which combine ESN with sparse Gaussian Processes; (3) the Locally Weighted Projected Regression (LWPR) [24], which exploits piecewise linear models to realise an incremental learning algorithm; and (4) the Recursive Least Square algorithm which is underlying the recursive System Identification approach [25,26] to identify recursive ARX models (RARX). These four algorithms differ from each other in several aspects: firstly, while the ESN, OIESGP and LWPR are non-parametric approaches, the RARX is parametric and fits the data by finding polynomial parameters. Also, the algorithms rely on different structures, *i.e.* neural networks, Gaussian processes, piecewise linear models, polynomial transfer functions. The main advantage of using these 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, in an online manner. In the learning process, we build models for each degree of freedom and for each 2D coordinate of each cluster, only considering single-output systems, where the data consist of $\langle(\mathbf{x}_j, y_j)\rangle(t)$. The base models are trained iteratively 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 $\hat{y}(t) = \tanh(\mathbf{w}^{out}\mathbf{x}(t))$, [22]. In the OIESGP model, the prediction $\hat{y}(t)$ 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, [23]. The RARX model updates the parameter estimates $\hat{\theta}$ at each iteration, while the prediction is calculated as $\hat{y}(t) = \psi^T(t)\hat{\theta}(t)$ where ψ represents the gradient of the predicted model output [26]. The LWPR model updates local models parameters by minimising a predicted residual sums of squares function and then produces the prediction $\hat{y}(t)$ as a weighted combination of local models, [24].

The ensemble prediction is obtained by combining the base model predictions through the ensemble integration. The ensemble prediction $\hat{y}_E(t)$ of the true value $y(t)$ is computed online at each time step t as:

$$\hat{y}_E(t) = \sum\nolimits_m w_m(t)\hat{y}_m(t) \tag{1}$$

where $\hat{y}_m(t)$ are the base model predictions and $w_m(t)$ the corresponding normalised weights ($\sum_m w_m(t) = 1$).

The weights are calculated so that the combination of models gives the closest estimate to the true value to be predicted. The ensemble weights update is performed by following a Bayesian model combination approach [27]. First, at each time t, a pruning step is performed among the base models predictions in order to eliminate bad predictors. The corresponding ensemble weights are set to zero if a threshold on the prediction error is not fulfilled. The remaining ensemble weights are then randomly initialised according to a Dirichlet distribution to allow sampling from the space of possible ensembles, and renormalised (we denote these preliminary weights as v_m).

The predictive accuracy of the ensemble, weighted according to the random weights v_m, is evaluated through the cumulative Mean Squared Error (MSE) score, calculated up to time t, denoted as $\varepsilon(t) = \sum_{\tau=1}^{t} \mathbb{E}[(y(\tau) - \hat{y}_{ens}(\tau))^2]$. A loss function $\ell(t)$ is then defined, based on the ensemble performance: $\ell(t) = \varepsilon(t) \log \varepsilon(t) + (1 - \varepsilon(t)) \log(1 - \varepsilon(t))$.

Through an iterative refinement process, the weights are updated by combining the random initialisation values $v_m^{(i)}$ drawn in each iteration $i = \{1, 2, ..., c\}$:

$$w_m^{(i)}(t) = w_m^{(i-1)}(t) W_i + \tilde{w}^{(i)} v_m^{(i)} \tag{2}$$

where c is the number of iterations applied to refine the estimates at each time step (the higher c, the more refined $\hat{y}_{ens}(t)$, the slower the computation), $\tilde{w}^{(i)} = e^{\ell(i) - \ell(i-1)}$ measures the improvement of the refinement and is used to combine the new updated weights with the random initialisation values, and W_t accounts for the relative increment of the cumulative loss, $W_i = \sum_{j=1}^{i-1} \tilde{w}^{(j)} / \sum_{j=1}^{i} \tilde{w}^{(j)}$. Note that at each time step the quality of each base model is reassessed and the ensemble weights change dynamically.

2) Multiple-Step-Ahead Prediction Models: In order to give the robots a longer prediction horizon, k-step-ahead prediction models have been considered. To realise multiple-step-ahead predictors, some authors proposed chaining of single-step-ahead predictors [7–9,28–30]. This approach is based on feeding one-step-ahead predictions as input to the models that produced them. The simplest implementation of this method is also called "naive", due to the fact that uncertainties generated in each iteration step of prediction are not considered. We use the naive implementation, assuming that l previous outputs $y(i)$ and inputs $\mathbf{x}(i)$, for $i = \{(t - l), ..., t\}$, as well as k future velocity commands $v_f(i),\ i = \{t+1, \dots, t+k\}$, are known. This assumption is realistic, since the previous outputs and inputs are available from the past experience of the robot, while the future commands (up to a certain number of steps ahead) can be thought of as already planned. In this implementation, the one-step-ahead predictions, performed by the base learners, are fed back as input for the next prediction together with the next input vector $\mathbf{x}(t + 1)$. To achieve the k-step-ahead prediction, this iteration is repeated k times. Note that the input vector $\mathbf{x}(t + j)$ includes the future commands up to time $t+j,\ j = \{1, \dots, k\}$, which are assumed to be known. The ensemble learner is then built so that the k-step-ahead predictions obtained from the base models are combined to get the most accurate estimate $\hat{y}_E(t + k)$ of the true value $y(t + k)$.

IV. Experimental Results

Five joints are involved in the babbling, namely the shoulder pitch, roll and yaw, the elbow flexion, and the wrist deviation. Babbling is realised by issuing to these joints velocity commands which are realised as a sum of two sinusoids with different frequency and amplitude $\sum_i a_i \sin(2\pi f_i),\ i = 1, 2$. These motor commands produce a complex motion of the arm and hand/end-effector, while allowing independent movements of limb parts which can thus be identified through the motion segmentation algorithm in the vision frames.

The base models and the ensemble models are evaluated in terms of Root Mean Squared Error (RMSE)[1]. The ensemble predictors are compared against the single base models, against an offline implementation of a standard tree-based bagging ensemble for regression [31], and against the homogeneous online ensemble obtained by applying the same ensemble integration algorithm but adopting homogeneous structures as base models (that is ensembles of ESNs only, of OIESGPs only, of RARXs only, and of LWPRs only).

Quantitative results are reported in Table I. The predictors are evaluated both in short-term and long-term prediction performance. The one-step-ahead predictions obtained with the proposed online heterogeneous ensemble learner are in all the cases more accurate than those of all the other alternative solutions. The proposed heterogeneous ensemble method guarantees difference between the base models: each of the underlying predictors shows different error types, so that single predictors' faults can be compensated by other predictors that produce opposite types of errors (such as overshoot and undershoot). The accuracy obtained by the proposed heterogeneous online ensemble method is approximately 20 to 85% higher compared to single predictors and homogeneous ensembles, and 50 to 60% higher compared to the accuracy obtained with the tree-based model.

The proposed online ensemble achieves the best performance also in the multiple-step-ahead prediction task, outperforming single predictors, the tree-based offline ensemble and homogeneous ensembles. In this case, the proposed heterogeneous online ensemble method outperforms single models and homogeneous ensembles by approximately 30 to 98% in accuracy, and the offline tree-based ensemble by approximately 30 to 60%. In this task, we set the prediction horizon to $k = 30$, corresponding to roughly $\Delta T = 3$ seconds in the future. This time horizon is usually the time within small base actions take place. A contribution in improving the prediction accuracy is given by the pruning step: tightening the pruning threshold can enhance the ensemble performance by eliminating bad predictors.

The high estimation accuracy achieved by the heterogeneous ensemble is useful, for practical purposes, *e.g.* to improve a robot's performance in control tasks involving precise positioning of the arm/end-effector or localisation of the end-effector in the robot's vision space. The heterogeneous ensemble allows to achieve the highest performance both in short and long term predictions, providing an accurate model for sensorimotor contingencies involving both proprioception and vision.

To demonstrate the generalisation of the learnt predictors beyond the trained data, we have evaluated the performance also on gestures that have a longer duration to complete, for example waving and pointing. In Table II we show the ensembles behaviour in generalising on these gestures. The waving gesture is the simplest one, only involving a

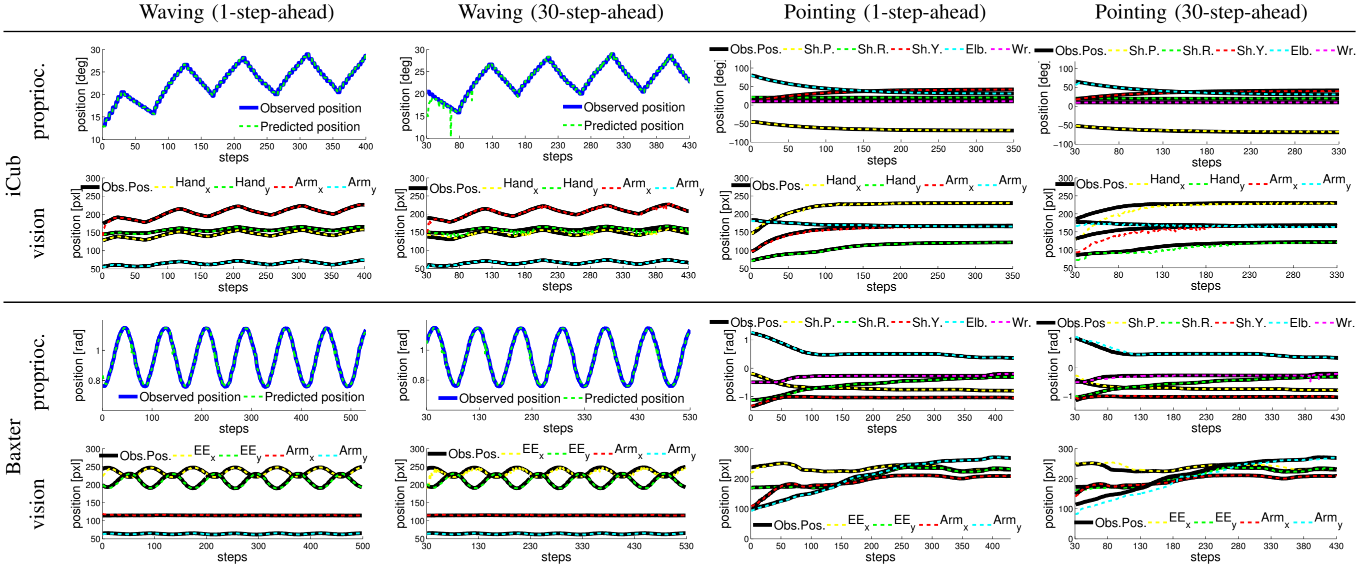
[1]The Root Mean Squared Error (RMSE) is defined as $\sqrt{\mathbb{E}[(y - \hat{y})^2]}$.

TABLE I: RMSE scores for one-step-ahead and 30-step-ahead predictions, on the iCub and the Baxter test data. The values for the proprioception data are measured and converted in $[deg]$ for the iCub and Baxter respectively, and measured in $[pxl]$ for the visual data. The proposed online heterogeneous ensemble achieves the best accuracy (scores in bold).

	Proprioceptive or Visual Element	Single base models				Offline	Online Homogeneous Ensembles				Proposed Online Heterog. Ensemble Method
		ESN	OIESGP	RARX	LWPR	Tree-based Bagging	Homog. ESN	Homog. OIESGP	Homog. RARX	Homog. LWPR	Heterog. Ensemble
iCub 1-step-ahead pred.	Shoulder Pitch	0.1489	0.2834	0.1516	0.1519	0.1740	0.1373	0.2746	0.1512	0.1419	**0.0830**
	Shoulder Roll	0.1140	0.2411	0.1240	0.1656	0.1501	0.1133	0.2328	0.1213	0.1501	**0.0698**
	Shoulder Yaw	0.1529	0.4972	0.1676	0.1879	0.1964	0.1521	0.2725	0.1607	0.1736	**0.0724**
	Elbow Flexion	0.1383	0.1753	0.1516	0.2016	0.1785	0.1371	0.1056	0.1468	0.2013	**0.0867**
	Wrist dev.	0.2584	0.3031	0.2850	0.2200	0.2748	0.2563	0.1758	0.2781	0.1811	**0.1066**
	Hand Cluster, x	0.3906	0.4939	0.3429	0.2678	0.3339	0.3823	0.4228	0.3418	0.2545	**0.2218**
	Hand Cluster, y	0.7740	0.4556	0.7297	0.5536	0.8163	0.7323	0.3961	0.7274	0.5268	**0.3004**
	Forearm Cl., x	0.2325	0.3904	0.2300	0.5910	0.2951	0.2312	0.3816	0.2284	0.5867	**0.1753**
	Forearm Cl., y	0.5640	0.6708	0.6008	3.1010	0.7144	0.5559	0.6402	0.5908	0.8121	**0.2808**
iCub 30-step-ahead pred.	Shoulder Pitch	0.3915	4.3151	3.8149	0.9188	3.7859	0.3530	0.7093	3.7393	0.3977	**0.1475**
	Shoulder Roll	0.3506	3.4874	3.1557	0.8847	3.1390	0.2905	0.2754	3.0993	0.4676	**0.1323**
	Shoulder Yaw	0.3782	4.1272	4.1806	1.1461	4.1523	0.3271	0.6241	4.1036	0.3621	**0.2238**
	Elbow Flexion	0.3969	3.6265	3.8086	0.9571	3.7832	0.3793	0.4393	3.7393	0.4460	**0.0684**
	Wrist dev.	0.7128	4.5872	3.0053	1.9300	2.9891	0.6797	0.4887	2.9913	0.6053	**0.1561**
	Hand Cluster, x	1.4875	8.9939	5.8513	1.6080	5.8028	1.4552	2.1733	5.7233	1.5515	**0.4340**
	Hand Cluster, y	2.7480	8.2960	6.1879	6.3595	16.0269	2.7398	4.2276	5.8135	1.7960	**1.1545**
	Forearm Cl., x	2.1960	1.9573	3.5112	7.0893	3.4258	2.1621	1.0487	3.4387	2.2106	**1.8361**
	Forearm Cl., y	2.5590	12.2303	4.2245	7.8542	13.9867	2.5555	1.5792	3.8746	7.2734	**0.9817**
Baxter 1-step-ahead pred.	Shoulder Pitch	0.1222	0.1157	0.0419	0.0447	0.0442	0.0502	1.7683	0.0414	0.0413	**0.0356**
	Shoulder Roll	0.0449	0.0733	0.0429	0.0505	0.0438	0.0650	1.7072	0.0424	0.0456	**0.0351**
	Shoulder Yaw	0.0508	0.0803	0.0440	0.0460	0.0452	0.0787	1.9564	0.0435	0.0438	**0.0365**
	Elbow Flexion	0.0439	0.1089	0.0421	0.0431	0.0437	0.0981	2.0010	0.0417	0.0421	**0.0341**
	Wrist dev.	0.1160	0.0937	0.0468	0.0502	0.0477	0.0839	1.8114	0.0457	0.0464	**0.0384**
	End-Eff. Cl., x	0.7773	0.8174	0.4862	1.2963	0.7996	0.4368	0.8178	0.4348	0.5812	**0.4340**
	End-Eff. Cl., y	0.7656	0.7735	0.6853	7.4233	1.1394	0.7496	0.7762	0.5997	0.8709	**0.5909**
	Arm Cluster, x	0.5028	0.5741	0.4809	2.1376	0.9796	0.5029	0.5586	0.4692	0.6012	**0.4566**
	Arm Cluster, y	0.2035	0.2134	0.2035	0.2095	0.2137	0.2036	0.2129	0.2024	0.2063	**0.2019**
Baxter 30-step-ahead pred.	Shoulder Pitch	0.2079	2.7314	0.5012	0.2315	0.5165	0.1865	1.0378	0.5011	0.2109	**0.0732**
	Shoulder Roll	0.1244	9.8553	0.4373	0.1419	0.4509	0.0949	0.8140	0.4372	0.1352	**0.0779**
	Shoulder Yaw	0.1626	2.9569	0.5053	0.2227	0.5217	0.1494	2.8191	0.5050	0.1995	**0.0932**
	Elbow Flexion	0.1941	2.3872	0.5130	0.2102	0.5290	0.1838	0.7002	0.5127	0.1899	**0.0867**
	Wrist dev.	0.1291	2.0303	0.4903	0.1874	0.5061	0.1204	1.7405	0.4894	0.1659	**0.0540**
	End-Eff. Cl., x	2.0561	4.3716	6.1392	6.4310	6.3767	1.9003	2.6667	6.1233	5.6672	**1.8755**
	End-Eff. Cl., y	2.7494	3.2962	10.0669	9.8536	10.4510	2.6049	3.1516	9.9824	9.6420	**2.4407**
	Arm Cluster, x	2.4029	4.8824	6.2943	6.8732	6.5788	2.4016	2.8459	6.2938	6.0330	**2.1155**
	Arm Cluster, y	1.3193	4.9293	1.3833	1.4951	1.3963	1.3187	4.4614	1.3829	1.2383	**1.0501**

TABLE II: Generalisation on longer duration gestures. Solid lines represent the observed joint positions, while dashed lines represent the predicted positions. Accurate one-step and 30-step predictions are achieved.

	Waving (1-step-ahead)	Waving (30-step-ahead)	Pointing (1-step-ahead)	Pointing (30-step-ahead)



single joint (the elbow) oscillating between two positions. Predictions on both joint and vision space are represented, for single-step and multiple-step predictions. Table II shows that the positions of the joints and of the clusters are very accurately predicted by the proposed ensemble method.

The experimental results demonstrate that the proposed heterogeneous ensemble benefits the learning process of forward models, guaranteeing diversity among the base learners, providing better performance than homogeneous solutions, and achieving highly accurate short- and long-term predictions in an online manner.

V. Conclusion and Future work

In this study, we focused on the problem of learning accurate sensorimotor contingencies models for humanoid robots in an online manner, following a developmental approach. We propose a heterogeneous online ensemble learning method which combines four diverse parametric and non-parametric online algorithms. The SMCs models built through the ensemble learning process consist of predictors relating motor commands with effects on proprioception and vision. Single- and multiple-step-ahead predictions have been addressed. The proposed online heterogeneous ensemble outperforms in accuracy individual predictors, offline tree-based ensembles and homogeneous ensembles. The effectiveness of the proposed method has been shown not only on test data but also in generalising on longer-duration gestures: the ensemble learners are able to accurately predict the movements of gestures such as pointing and waving. We demonstrate that the heterogeneous ensemble outperforms homogeneous methods. The heterogeneous ensemble learning method guarantees the necessary difference between the base models that allows improving the model prediction performance.

In this paper, each modality has been learnt separately; nonetheless, in future work multiple modalities can be integrated in the learning scheme. This will further enhance learning and anticipatory skills in autonomous robots, improving their model formation and adaptability.

References

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] J. Tani and S. Nolfi, "Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems," *Neural Networks*, vol. 12, no. 7, pp. 1131–1141, 1999.

[8] H.-M. Gross, A. Heinze, T. Seiler, and V. Stephan, "Generative character of perception: A neural architecture for sensorimotor anticipation," *Neural Networks*, vol. 12, no. 7, pp. 1101–1129, 1999.

[9] H. Hoffmann, "Perception through visuomotor anticipation in a mobile robot," *Neural Networks*, vol. 20, no. 1, pp. 22–33, 2007.

[10] A. Ribes, J. Cerquides, Y. Demiris, and R. L. de Mántaras, "Sensory anticipation of optical flow in mobile robotics," *arXiv preprint arXiv:1210.1104*, 2012.

[11] 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.

[12] 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.

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

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

[15] 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.

[16] A. Broun, C. Beck, T. Pipe, M. Mirmehdi, and C. Melhuish, "Bootstrapping a robot's kinematic model," *Robot. Auton. Syst.*, vol. 62, no. 3, pp. 330–339, Mar. 2014.

[17] T. Schmidt, R. Newcombe, and D. Fox, "Dart: Dense articulated real-time tracking," *Proceedings of Robotics: Science and Systems, Berkeley, USA*, 2014.

[18] H. Jung, J. Ju, and J. Kim, "Rigid motion segmentation using randomized voting," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 1210–1217.

[19] 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.

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

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

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

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

[24] S. Vijayakumar and S. Schaal, "Fast and efficient incremental learning for high-dimensional movement systems," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 1894–1899.

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

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

[27] K. Monteith, J. L. Carroll, K. Seppi, and T. Martinez, "Turning bayesian model averaging into bayesian model combination," in *Neural Networks (IJCNN)*. IEEE, 2011, pp. 2657–2663.

[28] R. Palm, "Multiple-step-ahead prediction in control systems with Gaussian process models and TS-fuzzy models," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 8, pp. 1023–1035, 2007.

[29] A. Girard, C. E. Rasmussen, J. Quinonero-Candela, R. Murray-Smith, O. Winther, and J. Larsen, "Multiple-step ahead prediction for non linear dynamic systems–a Gaussian process treatment with propagation of the uncertainty," *Advances in neural information processing systems*, vol. 15, pp. 529–536, 2002.

[30] J. Zhao, Y. Zhu, and R. Patwardhan, "Identification of k-step-ahead prediction error model and MPC control," *Journal of Process Control*, vol. 24, no. 1, pp. 48–56, 2014.

[31] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.