Preprint version; final version available at http://ieeexplore.ieee.org
IEEE International Conference on Robotics and Automation (2016)
Published by: IEEE

Hierarchical Action Learning by Instruction Through Interactive Grounding of Body Parts and Proto-actions

Maxime Petit and Yiannis Demiris

***Abstract*— Learning by instruction allows humans programming a robot to achieve a task using spoken language, without the requirement of being able to do the task themselves, which can be problematic for users with motor impairments. We provide a developmental framework to program the humanoid robot iCub without any hand-coded *a-priori* knowledge about any motor skills. Inspired by child development theories, the system involves hierarchical learning, starting with the human verbally labelling robot body parts. The robot can then focus its attention on a precise body part during robot motor babbling, and link the on-the-fly spoken descriptions of proto-actions to angle values of a specific joint. The direct grounding of proto-actions is possible through the use of a linear model which calculates the effects on the joint of the proto-action and the body part used, allowing a generalisation of the proto-action if the joint has never been used before. Eventually, transferring the grounding is allowed via learning by instructions where humans can combine the newly acquired proto-actions to build primitives and more complex actions by scaffolding them. The framework has been validated using a humanoid robot iCub, which is able to learn without any prior knowledge: 1) the name of its fingers and the corresponding joint number, 2) how to fold and unfold them and 3) how to close or open its hand and how to show numbers with its fingers.**

I. Introduction

Service robots are expected to assist humans for various tasks in a personalized manner, requiring the robot to adapt. Developmental robotics provides this capability through progressive and cumulative learning [1–3] using self-exploration or social interactions with humans [4], inspired by child development theories. Contrary to other primate species, caregivers interact actively with the children to help them organising information or teaching them new concepts [5]. In this context, human knowledge can be transmitted to the robot through verbal interactions. Several methods of learning have been investigated, *e.g.* learning by imitation [6–8] or learning by instructions [9–11]. The latter has the advantage of being an intuitive way for humans to teach the robot new actions. However, these methods always required some pre-coded primitive actions for the scaffolding of more complex actions. Learning by imitation or demonstration can be used to acquire them [12, 13], but this can be problematic for users with motor impairment. This is of crucial importance because these specific users are the ones which need robots to perform tasks for them which they cannot do themselves.

In this paper, by using human spoken description and feedback, a developmental framework is presented, allowing hierarchical learning in order for the robot to i) learn its body part name, ii) acquire proto-action semantic grounding, including generalisation capabilities for the used body part and iii) create primitives (based solely on proto-actions) and complex actions (based on proto-actions, primitives or other complex actions), as shown in Figure 1. One of the more important aspects is being able to develop these skills without any particular initial knowledge implemented *a-priori* inside the robot, removing the need for hand-coded or pre-programmed basic actions.



Fig. 1. Experimental setup of the framework: a human is interacting with the iCub through spoken language, allowing the robot to obtain the label of its body parts, acquire new proto-actions and learn more complex actions.

Section II presents the relevant work, especially focusing on grounding knowledge for robots and learning by instructions. Section III describes the architecture and framework and explains how body part names are grounded, proto-actions learned and complex actions built. Section IV illustrates the system with an experiment where the humanoid robot iCub learns the names of its fingers and acquires the proto-actions 'fold' and 'unfold' which are then used to teach the iCub more complex actions (how to close its hand, how to count with its fingers, *etc.*). Section V presents a conclusion and a discussion about future developments.

The authors are with the Personal Robotics Lab, Department of Electrical and Electronic Engineering, Imperial College London, UK. Email: *{m.petit, y.demiris}@imperial.ac.uk*

Preprint version; final version available at http://ieeexplore.ieee.org
IEEE International Conference on Robotics and Automation (2016)
Published by: IEEE

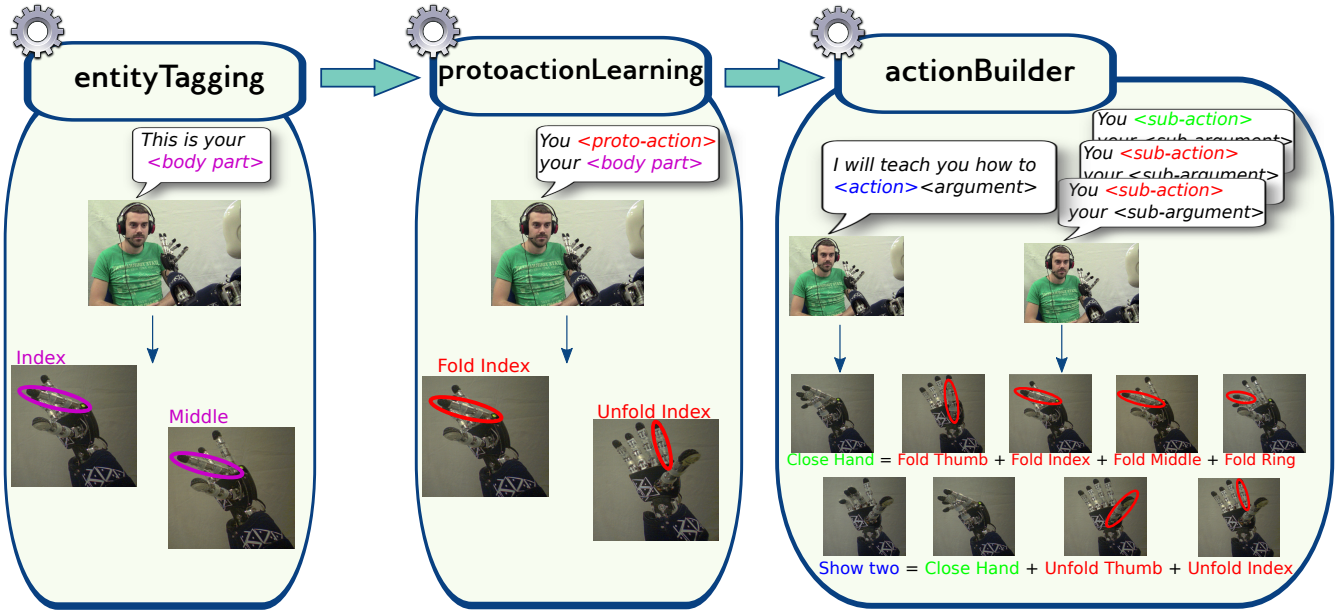


Fig. 2. Hierarchical Framework: entityTagging grounds body part name, protoactionLearning acquires basic motor skill using human description during motor babbling and actionBuilder allows the human to teach to the robot primitives and more complex actions with a learning by instructions method.

II. Related Work

Cangelosi and Riga have defined two mechanisms in order to ground language in autonomous cognitive systems [14]: direct grounding and transfer grounding (see Cangelosi's review [15] for more detail about the grounding problem). The first is defined as the capacity to link representations (perceptual and internal) to symbols with supervised feedback. Transfer grounding creates new symbols from basic ones, for instance using inference logic. Within their framework, direct grounding is managed with a learning by imitation to learn basic actions and their names, whereas linguistic descriptions from humans allow grounding transfer. They demonstrated this in simulated scenarios with the human communicating through written sentences. Higher order behaviors can also only be composed by two sub-actions.

Based on Bayesian Belief Networks, Demiris and Dearden developed a hierarchical learning by imitation system, using motor babbling in a developmental context [6]. They are able to imitate human actions using forward models, creating primitive actions that are then combined into more complex sequences. However, this knowledge is not yet semantically grounded, keeping the interaction mode with the human outside of the natural language speaking domain.

Felsberg *et al.* proposed an artificial cognitive system which is able to autonomously extend its capabilities [16] until being able to solve a shape-sorter puzzle. Using reinforcement learning for exploration in a perception-action approach, where the learning is driven from the motor side and not the sensing side, the system can learn to recognize objects by pushing them around and imitate grasping from a teacher and then generalize this action to unknown positions. But the language element is still missing from the grounding.

The semantically grounding problem has however been investigated by Mohan *et al.* while learning to ground adjective and nouns, spatial relations and verbs [9] or goal-oriented tasks [17] using interactive instructions within the Soar cognitive framework [18]. However the verbs and tasks are learned based on the composition of basic actions (point, pick up, put down) that are known by the robot *a-priori*.

The same problem of initial knowledge, which is problematic in particular for the framework's compatibility towards multiple robots, is still encountered in more recent works. The framework of Pointeau *et al.* is able to learn rules of board game rules (Tic-Tac-Toe), spatial knowledge (absolute locations or relative directions), temporal relations (before and after), actions (put, push), shared plans (cooperative musical activities), in successive levels of developmental mechanisms with the humanoid robot iCub [10, 19]. However the actions are based on a set of more basic actions (point-to, put, grasp, release) that are embedded in their motor control system.

Forbes *et al.* describe a system to allow a robot to understand spatial language (to the left of, below the, *etc.*), using inference to identify precise objects (rightmost, nearest, *etc.*) by programming with instructions [11]. However the initially needed knowledge is very large, including in particular 17 kinds of known motor commands.

III. Method

We first present the humanoid robot iCub used for the experiment and describe the different components of the architecture. Then we explain the semantic grounding of body parts using motor babbling from the robot and spoken descriptions from the human. We present the motor babbling procedure, and how a human describes proto-actions done by the iCub on-the-fly. Using the relation between a named bodypart and the joint, the robot can analyze the encoder values to extract the meaning of the proto-action (e.g. the

Preprint version; final version available at http://ieeexplore.ieee.org
IEEE International Conference on Robotics and Automation (2016)
Published by: IEEE

velocity to apply and/or the angle value to reach). These proto-actions (*e.g.* fold thumb, unfold index) are generalized to other body parts or combined to create more complex actions, like closing or opening the iCub's hand.

A. Platform and Architecture

We use the humanoid robotic platform iCub [20] with the middleware framework Yet Another Robot Platform (YARP) [21] to organise the communication between different modules. The Microsoft Speech Recognition software detects the spoken utterances, following a predefined grammar that allows semantic roles extraction for words in spoken utterances. An autobiographical-like memory system under the form of a PostgreSQL database [22] is used to store discrete events (*e.g.* recognized sentences, motor babbling actions) along continuous data (*e.g.* joint encoders). It captures the state of the world updated through all the sensor modules of the robot, and in particular the properties of entities of the world: objects have cartesian coordinates, dimensions, color and name whereas body parts have joint_number, name and limb part for example. The current work also involves a layer of 3 learning modules i) entityTagging to label objects and body parts, ii) protoactionLearning to extract proto-action commands of a body part, after a motor babbling events described by the human, and iii) actionBuilder, to compose such actions using the newly discovered proto-action, to compose primitives, or more complex actions when building using all the available type of action. This hierarchical architecture is shown in Figure 2.

B. Grounding Body Part

This developmental framework requires only a very little initial knowledge for the robot: it knows that it has several joints that can be moved, but their names are unknown. It has to be noted that not a single pre-defined primitive is known. Acquiring a symbolic label to link to a precise joint is crucial in order to be able to understand further descriptions or instructions during human-robot interaction. A typical user has little knowledge about the robot's internal structure, and does not know the joint number corresponding to each articulation. Instead, he can name the body part that corresponds to it: the robot should then acquire this knowledge quickly.

In order to acquire the label, the robot triggers a movement for each joint and asks the human to name the moving part. The body part name is extracted from the human utterance ("This is your <body-part_name>") and is used to label the unknown body part. From now on, both the human and the robot can indicate the joints using their name.

C. Learning Proto-actions

Now that the robot knows the name of its body parts, it investigates what kind of actions can be done with them. Knowing the name of the body part is needed to lead the attention of the human to the corresponding limb. It is also important for the robot in order to reduce the complexity of the following reasoning method: instead of extracting all the available encoder values, the iCub is able to create a subset of only the named joints to check for the effects of the proto-action. Thus the complexity is reduced from 53 DOF (all the degrees of freedom of the iCub) to potentially 1 DOF.

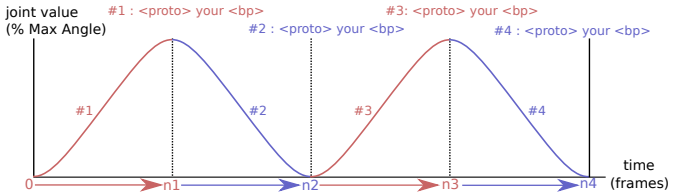


Fig. 3. Schema of the joint value according to time (frame number) divided into instances corresponding to the description. Internally, frame number are reset to 0 at the start of each description.

To proceed, the robot informs the human to pay attention to one of its body parts, asks him to describe the actions it will do (*i.e.* "you <proto-action_name> your <joint-name>") and executes motor babbling of the body part as developed by Zambelli and Demiris [23]: velocities commands are sent to the corresponding joint, leading to an increase or decrease of its angle. The encoder values are stored in the memory along with the recognized description of the human with their exact timing. This allows the decomposition of the motor babbling into different proto-actions (using <proto-action_name>), such as "fold" or "unfold", and among all the available joints, to extract the one that is of interest (using <joint_name> to find the corresponding joint number).

After the babbling is done, the iCub proceeds to a statistical analysis to generate the proto-action consequences. It retrieves the streaming of the joint values of the body-part during the babbling action from the memory: the annotations given on-the-fly by the human separate the stream into different instances corresponding to different proto-actions. The joint values are then normalised to become $\hat{\alpha}$ as percentages of the angle range $[\alpha_{j,m}{:}\alpha_{j,M}]$ of the current joint j: this is needed in order to compare different body part effects that may have different angle ranges. Figure 3 shows such decomposition of the motor babbling.

A Pearson Product-Moment correlation test is calculated for each proto-action described, in order to check if a linear correlation between the normalized angle $\hat{\alpha}$ and the frame number f is relevant. If the Pearson correlation coefficient r^2 is above a threshold τ, a linear regression is generated to estimate $\hat{\alpha}$ according to the frame number f for the current proto-action and body-part used, as shown in Eq. 1:

$$\hat{\alpha} = \beta_i + \delta_i \times f. \quad (1)$$

Where $\hat{\alpha}$ is the normalized angle, β_i is the starting value of the proto-action number i and δ_i is the velocity of the joint, with the evolution of the angle per frame f. However, in order to execute the proto-action, we are interested in the final value of the joint angle, not necessarily the starting value. As shown in Figure 4, because of the continuous motor babbling and the subsequent annotation, the final normalized angle value $\hat{\theta}_i$ of the proto-action i is the intercept β_{i+1} of the next proto-action $i+1$:

$$\hat{\theta}_i = \beta_{i+1}. \quad (2)$$

Preprint version; final version available at http://ieeexplore.ieee.org
IEEE International Conference on Robotics and Automation (2016)
Published by: IEEE

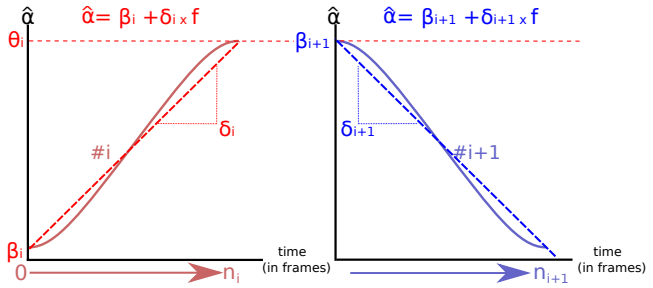


Fig. 4. Schema of the first linear model generated for proto-action i "fold" and $i+1$ "unfold". Note the intercept value β_{i+1} from the linear model of the proto-action $i+1$ is similar to the end value $\hat{\theta}_d$ of the proto-action i.

The robot now has a dataset of the final values of $\hat{\theta}_i$ for the proto-action number i, composed by the proto-action name and the bodypart used. A second linear model is then calculated to estimate the final value of the desired $\hat{\theta}_d$, with the effects of these two factors, $\gamma_{\text{pn}} \in \mathbf{\Gamma}$ for the proto-action name and $\psi_{\text{body}} \in \mathbf{\Psi}$ for the body part, along with the intercept angle value ϕ of the linear model:

$$\hat{\theta}_d \sim \phi + \gamma_{\text{pn}} + \psi_{\text{body}}. \tag{3}$$

A two-tailed t-test is performed for each parameter to test their significance, with an alpha risk $R_\alpha = 0.05$: if p-value > R_α, the null hypothesis can not be rejected and the coefficient is set to 0. This method to learn proto-actions is summarized in Algorithm 1. The generated ϕ, all the γ_{pn} and all the ψ_{body} are then saved.

At this point, the system can now predict the desired normalised angle value $\hat{\theta}_d$ by the addition of the extracted ϕ (*i.e.* intercept value of the linear model) with γ_{pn} (*i.e.* effect on the proto-action *per se*) and ψ_{body} (*i.e.* correction depending on the body part used). A generalisation is then possible: if the body part is new, ψ_{body} will be 0, but an angle can still be calculated using only the effect γ_{pn} of the proto-action name and ϕ. Eventually, because $\hat{\theta}_d$ is a percentage, an adjustment is made in case $\hat{\theta}_d$ is out of these boundaries, as shown in the following:

$$\hat{\theta}_d = \begin{cases} 0 & \text{if } \hat{\theta}_d < 0 \\ 100 & \text{if } \hat{\theta}_d > 100 \\ \hat{\theta}_d & \text{otherwise} \end{cases}$$

D. Scaffolding action primitives

Even if the robot had no prior knowledge about any actions, it has now access to some proto-actions involving the movement of its joints, including proto-actions with some body parts that it did not try before. These proto-actions are designed to move only one joint at a time, so their use in instructions by the human can be tedious when trying to learn complex actions. A system to learn primitives is then available, allowing the human to teach them to the robot by scaffolding different proto-actions.

The human triggers the learning by naming the primitive that will be teached, along with some arguments ("I will teach you to <primitive_name> your/the <argument>"). If the robot does not know it, iCub will ask the teacher to describe the primitive, using the proto-action one by one ("<proto-action_name your <body-part_name>"), until the human stops the teaching ("Teaching finished"). For each proto-action, the robot i) computes $\hat{\theta}_d$, ii) extracts the joint number j and the angle range of the corresponding body part $[\alpha_{j,m}, \alpha_{j,M}]$, iii) calculates the real desired angle $A_{j,d}$ using Eq. 4 and iv) executes the command (e.g. set pos j $A_{j,d}$).

$$A_{j,d} = \left(\frac{|\alpha_{j,m} - \alpha_{j,M}| \times \hat{\theta}_d}{100} \right) - \alpha_{j,m} \tag{4}$$

The list of proto-actions is then stored for each primitive: in the future, when called, the primitive or complex actions will be uncompressed until reaching the proto-actions to be executed.

Algorithm 1: Algorithm for learning proto-actions by calculating linear model (lm) coefficients $(\phi, \gamma_{\text{pn}}, \psi_{\text{body}})$ to estimate angle $\hat{\theta}_d$

Input : $\hat{\alpha}$, normalized angle of a joint, for each annotated proto-action, during time
Output: ϕ, intercept angle value
γ_{pn}, effects of each proto-action pn
ψ_{body}, effects of each body part $body$

```
for i ∈ protoaction − 1 do
    r² ← Pearson.regression (i+1)
    if r² > τ then
        lm (protoaction_{i+1}, α̂ ∼ β_{i+1} + δ_{i+1}Frame)
        θ̂_i ← β_{i+1}
        endValueData.add (θ̂_i, pn[i], body[i])
lmAngle ← lm (endValueData, θ̂_d ∼ φ + γ_pn + ψ_body)
for i ∈ pn do
    if t.test(γ_i) > R_α then
        γ_i = 0
for i ∈ body do
    if t.test(ψi) > R_α then
        ψ_i = 0
```

IV. Experiments

In this section we show how this developmental framework allows the iCub i) to discover the name of its left hand fingers, ii) to acquire two proto-actions: "fold" and "unfold", applied to the fingers, iii) to build primitives such as opening and closing its hand and iv) to learn more complex actions: showing numbers with its fingers (see the attached video).

A. Grounding Fingers Names

The iCub's initial knowledge is only composed by five unknown body parts, with joint number 9, 11, 13, 15 and 15 (iCub's hand is designed to have only one joint for the ring and pinky fingers). EntityTagging is then called, and the iCub moves each joint at a time and asks for the human to name them. The spoken utterance is detected ("This is your <body-part_name>" [finger]) and the finger's label is attributed to

Preprint version; final version available at http://ieeexplore.ieee.org
IEEE International Conference on Robotics and Automation (2016)
Published by: IEEE

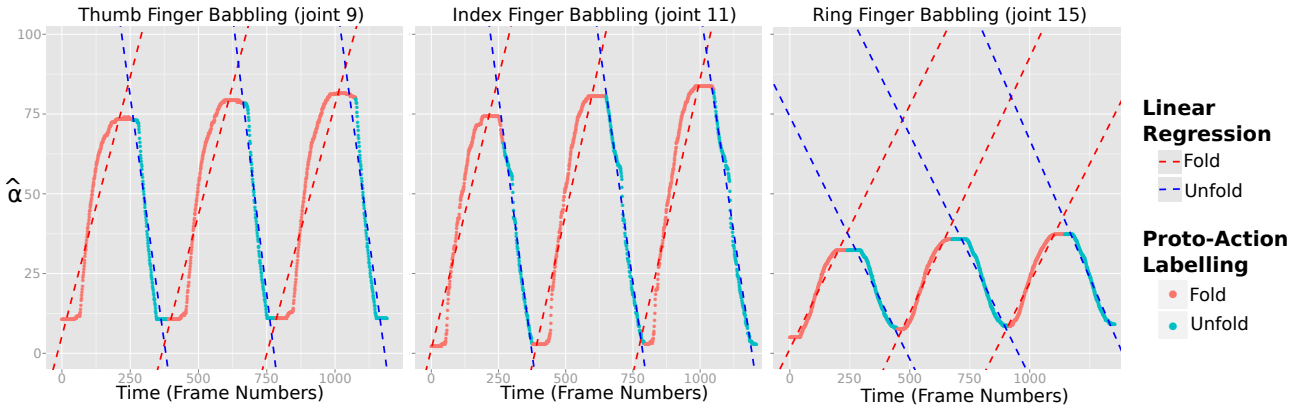


Fig. 5. Data obtained during the experiment: Angle value (normalized percentage) for the Thumb (Joint 9), Index (Joint 11) and Ring (Joint 15) fingers during the babbling. On-the-Fly spoken descriptions from the human provide segmentation and annotation of each prot-action during the babbling episode. Linear regression is shown in dotted line to estimate the intercept β_i of each protoaction i.

the moving body part: "thumb", "index", "middle", "ring", "pinky" are then respectively linked to joint number 9, 11, 13 15, and 15.

B. Learning proto-actions 'fold' and 'unfold'

With this newly acquired knowledge of finger names, the iCub asks the human to explain the movements it will do during a motor babbling episode, using the human description on-the-fly to segment and annotate the streaming information coming from its proprioception sense. It will only proceed to such babbling with 3 of the 5 known fingers (thumb, index and ring fingers) to show the generalisation capability. The human describes the scene ("you <fold> your <finger_name>", "you <unfold> your <finger_name>") during the motor babbling, which correspond to 6 velocity changes of the joint over 18 seconds. The learning is very fast, and only a small amount of repetitions is needed: 18 proto-actions have been learned, as shown in Figure 5.

For each proto-action, a Pearson Product-Moment correlation test validates (with $\tau = 0.7$) the linear correlation of the angle value according to the frame number, and a linear model to estimate the begin angle value β_{i+1} (i.e. the intercept of the linear model) is calculated (dotted line in Figure 5), leading to the end angle value $\hat{\theta}_i$ of the previous proto-action. A new dataset is generated with this information, as shown in Table I. It has to be noted that, because of this transformation, the last proto-action of each babbling can not obtain an estimated end angle value, reducing the available dataset by 3 proto-actions (one per finger babbling).

The second linear model (Eq. 3) is calculated using the dataset generated to be able to estimate the end angle value $\hat{\theta}_d$ using the effect of the proto-action name ($\mathbf{\Gamma}$=$\{\gamma_{\text{fold}}, \gamma_{\text{unfold}}\}$) and the body-part used ($\mathbf{\Psi}$=$\{\psi_{\text{thumb}}, \psi_{\text{index}}, \psi_{\text{ring}}\}$). The respective default values are γ_{fold} and ψ_{thumb}.

A summary of the linear model is shown in Table II, and shows the different effects: ϕ = 69.183, γ_{fold} = 0, γ_{unfold} = -56.288, ψ_{thumb} = 0, ψ_{index} = 1.038 and ψ_{ring} = -19.639. Note that the p-value of the significance two-tailed t-test for ψ_{index} is superior to the risk R_α = 0.05, so we can not reject the null hypothesis and ψ_{index} is set to 0.

TABLE I

PROTO-ACTION DETAILS (NUMBER, NAME AND ARGUMENT) WITH THE r^2 OF PEARSON PRODUCT-MOMENT CORRELATION AND THE ANGLE END VALUE $\hat{\theta}_i$ BASED ON THE RESULT OF THE GENERATED PARAMETERS OF THE LINEAR MODEL $\hat{\alpha} = \beta_i + \delta_i \times f$

proto-action number (i)	proto-action name	proto-action argument	Pearson r^2	Angle $\hat{\theta}_i$	slope δ_i
1	fold	thumb	0.88	73.22	0.32
2	unfold	thumb	0.91	7.78	-0.62
3	fold	thumb	0.87	74.97	0.32
4	unfold	thumb	0.92	10.46	-0.70
5	fold	thumb	0.86	66.91	0.31
6	unfold	thumb	0.87	NA	-0.63
7	fold	index	0.87	68.91	0.31
8	unfold	index	0.98	-2.59	-0.65
9	fold	index	0.91	81.86	0.38
10	unfold	index	0.98	10.58	-0.60
11	fold	index	0.88	79.77	0.37
12	unfold	index	0.97	NA	-0.55
13	fold	ring	0.95	37.83	0.15
14	unfold	ring	0.94	4.28	-0.15
15	fold	ring	0.96	41.67	0.16
16	unfold	ring	0.95	9.67	-0.15
17	fold	ring	0.93	41.70	0.15
18	unfold	ring	0.97	NA	-0.17

TABLE II

SUMMARY OF THE LINEAR MODEL ESTIMATING THE EFFECT OF THE PROTO-ACTION AND BODY PART USED ON THE DESIRED $\hat{\theta}_d$

Linear Model : $\hat{\theta}_d \sim \phi + \gamma_{\text{pn}} + \psi_{\text{body}}$

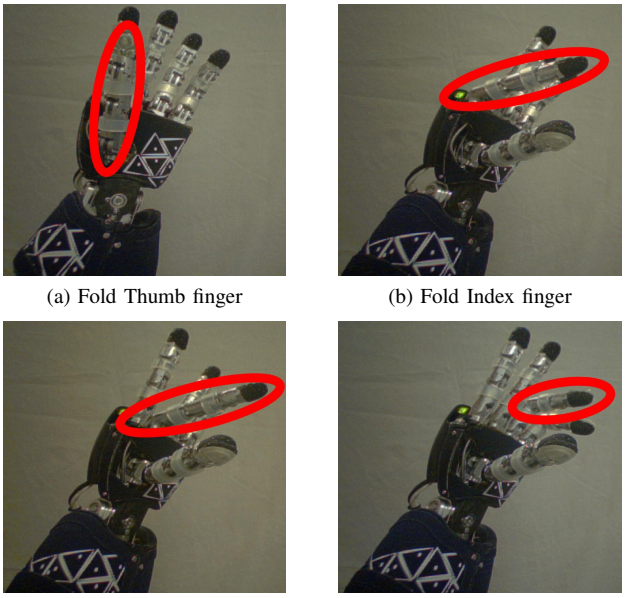
Coefficient name	Estimated Value	Standard Error	p-value	
ϕ	69.183	5.27	$4.62e^{-08}$	***
γ_{fold}	0	-	-	
γ_{unfold}	-56.288	5.62	$7.28e^{-07}$	***
ψ_{thumb}	0	-	-	
ψ_{index}	1.038	6.75	0.8805	
ψ_{ring}	-19.639	6.75	0.0141	*

Preprint version; final version available at http://ieeexplore.ieee.org
IEEE International Conference on Robotics and Automation (2016)
Published by: IEEE

C. Scaffolding primitives and actions with iCub hand

The iCub has now learned 2 different proto-actions, "fold" and "unfold", including the precise body part effect when using 3 of its 5 fingers. We will first explain how commands can be generated from it to execute these proto-actions.

1) Executing the learned proto-actions: By combining the parameters discovered by the linear model of Eq. 3 and shown in Table II, the robot can produce the "fold" and "unfold" proto-action not only with the demonstrated thumb, index and ring finger, but also for instance with the middle finger. In such case, the effect of the body part is considered 0, so $\hat{\theta}_d = \phi + \gamma_{\text{pn}}$ (*i.e.* 69 for fold, 13 for unfold). Table III shows the desired percentage of the angle value $\hat{\theta}_d$ (rounded) with the computed actual angle $A_{j,d}$ (rounded) using the angle range of the corresponding joint j ([0:90] for thumb, index, middle fingers [0:250] for ring, pinky fingers). Figure 6 shows the hand configuration for each "fold" proto-action, including the generalisation for the middle finger.



(a) Fold Thumb finger (b) Fold Index finger

(c) Fold Middle finger (d) Fold Ring finger

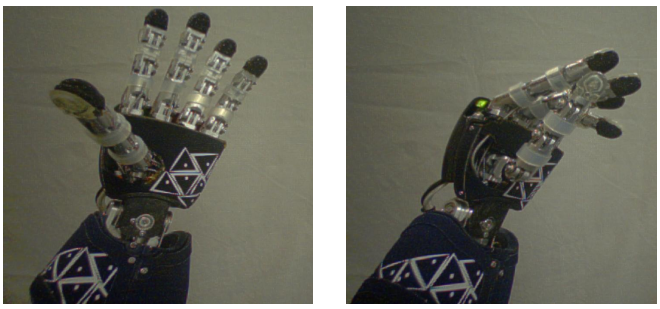
Fig. 6. Show fold each fingers (with the others unfolded). Note that the "fold middle" has been achieved through generalisation.

TABLE III

PROTO-ACTIONS COMMANDS EXECUTION. NOTE THE CORRECTION OF $\hat{\theta}_d$ FOR THE RING FINGER WHICH HAS TO BE IN [0:100], AND IN BOLD THE GENERALISATION OF THE MIDDLE FINGER.

Proto-actions			
proto-action name	proto-action argument	$\hat{\theta}_d$ ($[\alpha_{j,m}{:}\alpha_{j,M}]$)	command sent *set pos* j $A_{j,d}$
fold	thumb	69 ([0:90])	set pos 9 62
	index	69 ([0:90])	set pos 11 62
	ring	49 ([0:250])	set pos 15 123
	middle	**$\hat{\theta}_d$ = 69 ([0:90])**	**set pos 13 62**
unfold	thumb	13 ([0:90])	set pos 9 12
	index	13 ([0:90])	set pos 11 12
	ring	-7 $\longrightarrow$ 0 ([0:250])	set pos 15 0
	middle	**$\hat{\theta}_d$ = 13 ([0:90])**	**set pos 13 12**

2) Building primitives: The iCub has now acquired the capacity to produce several proto-actions. The user can then extend the motor skills of the robot by teaching it a primitive action: <close> the <hand>. The human can initiate a teaching interaction with the iCub ("I will teach you a <primitive>"), triggering a question from the iCub, which needs first to know the name of the primitive ("What kind of <primitive> do you want to teach me?"). The human provides the name and argument of the primitives ("<close> your <hand>") and the robot invites him to describe ("All right, can you describe how I can <close> my <hand>, please?"). The human details the proto-action sequence to do so, one by one, and the iCub is executing the command on-the-fly (as shown in Table IV): "<fold> your <thumb>", "<fold> your <index>", "<fold> your <middle> finger", "<fold> your <ring> finger"). As explained earlier, the joint moving the ring and pinky finger is the same for the iCub hand, so these 4 proto-action are sufficient to close the hand. This is shown in Figure 7, where the human asks the iCub to execute its newly acquired primitive action ("Can you <close> your <hand>"). In the same principle, Table IV shows the results of the learning of the primitive "open hand".



(a) Open Hand (b) Close Hand

Fig. 7. Learned Primitive actions with open and close hand.

TABLE IV

LEARNED PRIMITIVES, WITH THEIR PROTO-ACTIONS DECOMPOSITION

"Close Hand"		
Human Instructions	*proto-action decomposition*	*command sent*
1. "fold your thumb"	fold thumb	set pos 9 62
2. "fold your index"	fold index	set pos 11 62
3. "fold your middle finger"	fold middle	set pos 13 62
4. "fold your ring finger"	fold ring	set pos 15 123

"Open Hand"		
1. "unfold your thumb"	unfold thumb	set pos 9 62
2. "unfold your index"	unfold index	set pos 11 12
3. "unfold your middle finger"	unfold middle	set pos 13 12
4. "unfold your ring finger"	unfold ring	set pos 15 0

Preprint version; final version available at http://ieeexplore.ieee.org
IEEE International Conference on Robotics and Automation (2016)
Published by: IEEE



(a) Show one (b) Show two (c) Show three (d) Show four (e) Show five

Fig. 8. Illustrates the complex action "show", from left to right: one, two, three, four, five, with the real iCub.

3) Building more complex actions: The iCub has developed two proto-actions at this stage ("fold" and "unfold") with a precise knowledge when using thumb, index or ring finger, but is able to generalise to other body parts (*e.g.* its middle finger). He also learned two primitive actions: "close hand" and "open hand". These can be used to learn even more complex actions, using the newly learned primitives instead of the proto-actions, the human can reduce the number of instructions needed to keep the interaction less tedious.

We will now teach the iCub how to show numbers with its fingers, creating 5 actions, based on instructions using both acquired proto-actions and primitives. The procedure is the same than previously, except that building a complex actions allows the use of primitives in the sequence. Table V details the scaffolding of these actions and Figure 8 shows the results.

TABLE V
COMPLEX ACTIONS SCAFFOLDING: ICUB SHOWING NUMBERS

"Show one"		
Human Instructions	*proto-action decomposition*	*command sent*
1. "close your hand"	fold thumb fold index fold middle fold ring	set pos 9 62 set pos 11 62 set pos 13 62 set pos 15 123
2. "unfold your thumb"	unfold thumb	set pos 9 12
"Show two"		
1. "close your hand"	fold thumb fold index fold middle fold ring	set pos 9 62 set pos 11 62 set pos 13 62 set pos 15 123
2. "unfold your thumb"	unfold thumb	set pos 9 12
3. "unfold your index"	unfold index	set pos 11 12
"Show three"		
1. "open your hand"	unfold thumb unfold index unfold middle unfold ring	set pos 9 12 set pos 11 12 set pos 13 12 set pos 15 0
2. "fold your ring"	fold ring	set pos 15 123
"Show four"		
1. "open your hand"	unfold thumb unfold index unfold middle unfold ring	set pos 9 12 set pos 11 12 set pos 13 12 set pos 15 0
2. "fold your thumb"	fold thumb	set pos 9 69
"Show five"		
1. "open your hand"	unfold thumb unfold index unfold middle unfold ring	set pos 9 12 set pos 11 12 set pos 13 12 set pos 15 0

V. CONCLUSION AND FUTURE WORK

We have presented a developmental framework for humanoid robots and have shown that with no particular *a-priori* knowledge, the system is able to semantically ground some of its body parts (*e.g.* fingers), acquire proto-actions using these parts by social learning and eventually scaffolding these basic motor skills to create different primitives and complex actions. Only a small amount of data is needed to create 2 proto-actions (fold and unfold) with 3 different fingers (thumb, index, ring fingers): 3 motor babbling sequences of 18 seconds (with a velocity change each 3 seconds) is sufficient. The proto-actions are defined as moving the body part joint into a desired angle $A_{j,d}$ calculated by using linear model generation to estimate the intercept angle value ϕ, the effect of the proto-action *per se* (γ_{pn}) and the potential correction for the body part ϕ_{body}. This method allows a generalisation of the proto-actions, in particular when considering a proto-action known but never done with a body part: the body part correction effect ϕ_{body} is then set to 0.

The robot can then employ this knowledge to acquire new primitive capabilities or more complex actions using a learning by instructions method (transfer grounding) in a scaffolding context. The human can easily teach the robot how to perform different action primitives by combining the newly available proto-actions. In the same manner, more complex actions are detailed by the human, keeping the number of instructions at a low level by using both proto-actions and primitives.

The present works main assumption is that the human is able to describe a motor babbling action on-the-fly, in order to segment it into proto-actions. Future work will focus on reducing the user workload by allowing an unordered set of labels from the human, as introduced by Mangin and Oudeyer [24]. Using linguistic descriptions under the form of unordered primitive labels, they can learn to recognise parallel combinations of human actions (dance motions) by exploiting a non-negative Matrix Factorization.

Another improvement may also be to increment the kind of action (*e.g.* object oriented) that the system can learn, especially by introducing methods to learn manipulation of objects, from one place to another, and thus the need to develop spatial semantics as explored by Guadarrama *et al.* [25] and Zampogiannis *et al.* [26]. Shared plan learning, involving a human-robot cooperation to achieve a common goal [27], may also be a possible extension.

Preprint version; final version available at http://ieeexplore.ieee.org
IEEE International Conference on Robotics and Automation (2016)
Published by: IEEE

Such a developmental framework is also interesting for multi-robot teams: no prior knowledge about the robot body schema or specific module is needed, and the proto-actions are designed, using typical proprioceptive data, to work on low-level by directly controlling the angle of the robot's joints. Thus, even for an heterogenous team, the hierarchical knowledge learned might be transferred from one robot to another and can complement the multi-robot learning from demonstration [28] to go from homogenous to heterogeneous robot teams. We are planning to test this capability using additional humanoid robots (Nao and Baxter).

ACKNOWLEDGEMENT

This research was funded in part by the EU projects WYSIWYD (grant FP7-ICT-612139) and PAL (grant H2020-PHC-643783). The authors gratefully acknowledge the comments and continued support from the members of the Imperial College Personal Robotics Lab.

REFERENCES

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

[2] A. Cangelosi, G. Metta, G. Sagerer, S. Nolfi, C. Nehaniv, K. Fischer, J. Tani, T. Belpaeme, G. Sandini, F. Nori, L. Fadiga, B. Wrede, K. Rohlfing, E. Tuci, K. Dautenhahn, J. Saunders, and A. Zeschel, "Integration of Action and Language Knowledge: A Roadmap for Developmental Robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 167–195, sep 2010.

[3] A. F. Morse, J. De Greeff, T. Belpeame, and A. Cangelosi, "Epigenetic robotics architecture (ERA)," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 4, pp. 325–339, 2010.

[4] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 143–166, 2003.

[5] B. J. King, "Social Information Transfer in Monkeys Apes and Hominids," *Yearbook of Physical Anthropology*, vol. 34, no. Supplement 13, pp. 97–116, 1991.

[6] Y. Demiris and A. Dearden, "From motor babbling to hierarchical learning by imitation: a robot developmental pathway," in *International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, 2005, pp. 31–37.

[7] Y. Demiris and A. Meltzoff, "The Robot in the Crib: A Developmental Analysis of Imitation Skills in Infants and Robots," *Infant Child Development*, vol. 17, no. 1, pp. 43–53, 2008.

[8] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *International Conference on Human-robot Interaction*, 2007, p. 255.

[9] S. Mohan, A. Mininger, J. Kirk, and J. E. Laird, "Learning grounded language through situated interactive instruction," in *AAAI Fall Symposium on Robots Learning Interactively from Human Teachers*, 2012, pp. 30–37.

[10] G. Pointeau, M. Petit, and P. F. Dominey, "Successive Developmental Levels of Autobiographical Memory for Learning Through Social Interaction," *IEEE Transactions on Autonomous Mental Development*, vol. 6, no. 3, pp. 200–212, sep 2014.

[11] M. Forbes, R. P. N. Rao, L. Zettlemoyer, and M. Cakmak, "Robot Programming by Demonstration with Situated Spatial Language Understanding," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 2014–2020.

[12] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in Cognitive Sciences*, vol. 6, no. 11, pp. 481–487, 2002.

[13] M. Petit, S. Lallee, J.-D. Boucher, G. Pointeau, P. Cheminade, D. Ognibene, E. Chinellato, U. Pattacini, I. Gori, U. Martinez-Hernandez, H. Barron-Gonzalez, M. Inderbitzin, A. Luvizotto, V. Vouloutsi, Y. Demiris, G. Metta, and P. F. Dominey, "The Coordinating Role of Language in Real-Time Multimodal Learning of Cooperative Tasks," *IEEE Transactions on Autonomous Mental Development*, vol. 5, no. 1, pp. 3–17, mar 2013.

[14] A. Cangelosi and T. Riga, "An embodied model for sensorimotor grounding and grounding transfer: experiments with epigenetic robots." *Cognitive science*, vol. 30, no. 4, pp. 673–689, 2006.

[15] A. Cangelosi, "Grounding language in action and perception: From cognitive agents to humanoid robots," *Physics of Life Reviews*, vol. 7, no. 2, pp. 139–151, 2010.

[16] M. Felsberg, J. Wiklund, and G. Granlund, "Exploratory learning structures in artificial cognitive systems," *Image and Vision Computing*, vol. 27, no. 11, pp. 1671–1687, 2009.

[17] S. Mohan and J. E. Laird, "Learning Goal-Oriented Hierarchical Tasks from Situated Interactive Instruction," in *AAAI Conference on Artificial Intelligence*, 2014, pp. 387–394.

[18] J. E. Laird, *The Soar Cognitive Architecture*. MIT Press, 2012.

[19] G. Pointeau, M. Petit, and P. F. Dominey, "Robot Learning Rules of Games by Extraction of Intrinsic Properties," in *International Conference on Advances in Computer-Human Interactions*, Nice, France, 2013, pp. 109–116.

[20] 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*, vol. 23, no. 8-9, pp. 1125–1134, 2010.

[21] P. Fitzpatrick, G. Metta, and L. Natale, "YARP: Yet Another Robot Platform," *International Journal of Advanced Robotic Systems*, vol. 3, no. 1, pp. 43–48, 2006.

[22] M. Petit, T. Fischer, and Y. Demiris, "Lifelong Augmentation of Multi-Modal Streaming Autobiographical Memories," *IEEE Transactions on Cognitive and Developmental System*, 2016.

[23] M. Zambelli and Y. Demiris, "Online Ensemble Learning of Sensorimotor Contingencies," in *IEEE/RJS International Conference on Intelligent Robots and Systems, Workshop on Sensorimotor Contingencies for Robotics*, Hamburg, 2015.

[24] O. Mangin and P. Y. Oudeyer, "Learning to recognize parallel combinations of human motion primitives with linguistic descriptions using non-negative matrix factorization," in *IEEE International Conference on Intelligent Robots and Systems*, 2012, pp. 3268–3275.

[25] S. Guadarrama, L. Riano, D. Golland, D. Gouhring, Y. Jia, D. Klein, P. Abbeel, and T. Darrell, "Grounding spatial relations for human-robot interaction," in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 1640–1647.

[26] K. Zampogiannis, Y. Yang, C. Ferm, and Y. Aloimonos, "Learning the Spatial Semantics of Manipulation Actions through Preposition Grounding," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 1389–1396.

[27] S. Lallee, K. Hamann, J. Steinwender, F. Warneken, U. Martienz, H. Barron-Gonzales, U. Pattacini, I. Gori, M. Petit, G. Metta, P. Verschure, and P. F. Dominey, "Cooperative human robot interaction systems: IV. Communication of shared plans with Naive humans using gaze and speech," in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 129–136.

[28] S. Chernova and M. Veloso, "Confidence-based multi-robot learning from demonstration," *International Journal of Social Robotics*, vol. 2, no. 2, pp. 195–215, 2010.