Hierarchical Reinforcement Learning and Central Pattern Generators for Modeling the Development of Rhythmic Manipulation Skills

Anna Lisa Ciancio, Loredana Zollo, and Eugenio Guglielmelli
Biomedical Robotics and Biomicrosystem Lab, Università Campus Bio-Medico di Roma
Via Álvaro del Portillo 21, I-00128 Roma, Italy
{a.ciancio, l.zollo, e.guglielmelli}@unicampus.it

Daniele Caligiore and Gianluca Baldassarre
Laboratory of Computational Embodied Neuroscience, Istituto di Scienze e Tecnologie della Cognizione, Consiglio Nazionale delle Ricerche (LOCEN-ISTC-CNR)
Via San Martino della Battaglia 44, I-00185 Roma, Italy
{daniele.caligiore, gianluca.baldassarre}@iste.cnr.it

Abstract—The development of manipulation skills is a fundamental process for young primates as it leads them to acquire the capacity to finely modify the world to their advantage. As other motor skills, manipulation develops on the basis of motor babbling processes, which are initially heavily based on the production of rhythmic movements. We propose a computational bio-inspired model to investigate the development of functional rhythmic hand skills from initially unstructured movements. The model is based on a hierarchical reinforcement-learning actor-critic model that searches the parameters of a set of central pattern generators (CPGs) having different degrees of sophistication. The model is tested with a simulated robotic hand engaged in rotating bottle cap-like objects having different shape and size. The results show that the hierarchical architecture of the model allows it to develop skills based on different combinations of CPGs so to suitably manipulate the different objects. Overall, the model shows to be a valuable tool for the study of the development of rhythmic manipulation skills in primates.

Index Terms—Actor critic, humanoid simulated robotic hand, motor babbling, neural networks.

I. INTRODUCTION

The development of manipulation skills depends on the maturation of the nervous system and developmental mechanisms, however the precise contributions of each process are not fully understood yet. This is in part due to an incomplete understanding of the maturation dynamics of different parts of the motor system (e.g. the pyramidal tract and its direct projections to the motor neurons of the spinal cord, some of which are proposed to underlie skilled movements of hand and digits [1,2], and in part to the knowledge gap on the learning mechanisms that lead to the development of manipulation skills [3]. This research focuses on the latter issue.

During motor development infants perform a number of apparently unstructured movements (“motor babbling”, [4]) that since Piaget's theories on “circular reactions” [5] are thought to play a fundamental role in motor development. Although various models have been proposed to study motor babbling and its role in motor development, most of them (e.g., [6,7]) overlook the key fact that most movements produced during early motor babbling are rhythmic movements [8], not discrete movements (see [9] on the possible relations between the two in real brains). These movements involve for example repeated waving, scratching, petting, wiping, hitting (with or without another object), turning (e.g. to perceive objects from different perspectives), etc., that allow infants to discover the functioning of their own body, the structure of the world, and the potential effects of their actions on it [8,10,11,12]. Rhythmic movements are probably so important because allow infants to acquire multiple sample data needed to develop motor skills notwithstanding the high noise affecting early behaviour.

Importantly, the development of manipulation skills implies a gradual passage from random exploratory rhythmic movements to functional movements that produce interesting, relevant, or useful effects on the environment. These functional movements can be either rhythmic or discrete [10,13,14]. The literature on development of manipulation skills lacks of models and hypotheses to investigate and frame the mechanisms and processes that lead to the progressive development of rhythmic movements. This paper accomplishes a first step to build such a model, and shows how it can be used to generate interesting hypotheses and predictions on the development of rhythmic manipulation skills. In details, the paper presents a bio-inspired hierarchical softly-modular reinforcement-learning system based on central pattern generators (CPGs), suitable for studying the autonomous acquisition of manipulation skills by trial-and-error learning.

The model was developed following four biological constraints. First, the trial-and-error mechanisms guiding the learning of the system were implemented on the basis of a reinforcement-learning actor-critic model [15]. This model has been shown to have a structure and functioning that resembles those of basal ganglia, the brain structures at the basis of trial-and-error learning and decision making in organisms (the actor component of the model captures the functioning of the matrosomes of basal ganglia, the critic captures the functioning of the striosomes, and the TD-error learning algorithms used by the model resembles the dynamics of the phasic-dopamine learning signal [16]. Recently, actor-critic versions of reinforcement learning have also been shown to have desirable computational properties and to be applicable with success to robotic setups similar to those used here (e.g., based on the use of CPGs and other dynamic motor primitives, [17,18,19]). Note that here a simple neural
implementation of the actor-critic model was used as it was sufficient for the scope of this research.

Second, the actor-critic model generates the parameters of Central Pattern Generators (CPGs), systems capable of producing oscillatory signals when suitably activated [20, 21]. Neurophysiological evidence shows that in mammals rhythmic motor patterns are generated on the basis of the activation of neural circuits implementing CPGs located mainly in the spinal cord. When activated CPGs generate rhythmic movement patterns by alternating the activation of flexor/extensor muscles, thus supporting behaviours such as locomotion, respiration, and chewing [20,22]. Fingers can also be controlled by CPGs and exhibit rhythmic movement patterns [23,24]. The works presented in [25,26] have already proposed some models based on CPGs with the goal of reproducing the rhythmic contact patterns observed in humans engaged in rotating an object. However, those models use hardwired CPGs and this does not permit to study development of manipulation skills, the goal of this work.

Third, the model is based on a hierarchical soft-modular architecture in analogy with the hierarchical organization of basal ganglia and motor cortex [27,28]. In particular, the system relies upon hierarchy to decide which of a set of CPGs having different complexity can be used to tackle different tasks. These type of architecture is fundamental as it allows the system to autonomously decide the sophistication of the computational resources to invest to acquire and perform different manipulation behaviours, based on object features and task complexity (cf. [29,30]).

Lastly, the model was tested on a simulated robotic hand (the iCub robot hand) interacting with 3D simulated objects. This assured that the model acquired the capacity to face dynamical and noise complexities similar to those tackled by primates.

As we shall see, the results shows that the model can be used with success to study the transition, based on trial-and-error learning, from unstructured rhythmic movements to functional ones, and to investigate the specific processes involved in such passage. In this respect, the model shows how different object features (mainly size and shape) pose different challenges and require different computational resources to develop rhythmic movements to tackle them (in line with [10,11]). Moreover, they show how the hierarchical architecture of the system gives rise to an effective emergent utilization of combinations of different CPGs depending on the complexity of the manipulation task at hand.

The rest of the paper is organized as follows. Sec. 2 presents the architecture and functioning of the model. Sec. 3 presents the results of the tests of the model. Finally, Sec. 4 draws the conclusions.

II. METHODS

A. The manipulation task

The model was tested with a custom simulator of the hand of the humanoid robot iCub (www.icub.org). The task required that the simulated robot developed the movements of the thumb and index fingers from random exploratory rhythmic movements to functional rhythmic movements in order to rotate as fast as possible several objects (having different shape and size) around than axis perpendicular to the hand palm. The objects were anchored to the world as in the case of bottle caps to be unscrewed. Each version of the model was trained for 5000 trials, each time with a different object. At the end of every trial, each lasting 300 simulation cycles, the rotation angle of the object was normalized and used as reward signal (see Sect. C). Notice that the task was rather challenging for at least three reasons. First, learning was based exclusively on the rare scalar value of reinforcement. Second, different shapes and size of the objects required different movements. Third, the rotation of the object required well coordinated dynamical movements involving the controlled joints of the fingers.

B. Overview of the system architecture

The overall system architecture is shown in Fig. 1. The key elements of such architecture are: three neural “experts”; one neural selector; three CPGs having a different degree of complexity (one for each expert); Proportional Derivative (PD) controllers; simulated hand and objects. Each expert and the selector receives as input the activation of a neural map encoding the object size and shape (in this study each version of the model is trained separately for each object, so this input, added to study the generalization properties of the system in future work, acts as a bias and is not further discussed here). At the beginning of each trial (and only then), each expert generates the parameters of the corresponding CPG, and the selector generates the “weights” used to “gate” the commands decided by the experts' CPGs (gating is obtained through a weighted average, see below). At each step, the desired joint angles of the expert CPGs are weighted with the selector weights to produce one desired angle for each joint. The resulting desired joint angles are sent to Proportional Derivative (PD) controllers to generate, on the basis of the current angles, the joint torques of the robotic simulated hand engaged in rotating the object. The experts and selector are in charge of discovering by trial-and-error the parameters to send to the three CPGs so that the hand maximizes the rotation of the object. Learning is supported by a “critic” component computing the TD-error (the learning of the system is based on the actor-critic reinforcement learning algorithms) and is performed, at the end of each trial, on the basis of a reinforcement signal proportional to the rotation of the object. In the follows all these components and processes are described in detail.

C. Architecture, functioning and learning of the neural network component

The core of the model is based on a hierarchical actor-critic reinforcement-learning component (Fig. 2) formed by three experts, three CPGs associated to the experts, a selector, and a critic (the gating used here is inspired by the seminal model [30]; see [31] for a hierarchical actor-critic reinforcement-learning model that does not use CPGs).

1) Actor experts functioning

The three experts get as input the object size and shape and return as output the parameters of the three CPGs (amplitude, phase, frequency, and offsets, see Sect. C) They are described below:
Fig. 1 The architecture of the hierarchical version of the model. The components of the model are: actor-critic reinforcement learning experts and selector, CPGs, PD Controllers, and simulated hand and object. Arrows represent information flows happening at each steps whereas dashed arrows represent information flows happening at the beginning of each trial.

First expert: 12 output units encoding the parameters of the CPG-C (i.e. “complex CPG”) illustrated in Fig. 3.a.
Second expert: 5 output units encoding the parameters of the CPG-M (i.e. “medium complexity CPG”) illustrated in Fig. 3.b.
Third expert: 2 output units encoding the parameters of the CPG-S (i.e. “simple CPG”) illustrated in Fig. 3.c.

Formally, the output unit $y_j$ of the experts (all encoded in a single vector) receive the signals from the input units $x_i$ of the input array (encoded in a vector) via connections weights $w_{ji}$ and form an activation potential $PA_j$:

$$PA_j = \sum_i (w_{ji}x_i)$$  \hspace{1cm} (1)

and an activation based on a sigmoidal function:

$$y_j = \frac{1}{1 + e^{-PA_j}}$$  \hspace{1cm} (2)

To foster exploration, a noise is added to $y_j$:

$$y_j^N = y_j + N$$  \hspace{1cm} (3)

where $N$ is a random number. During each training session, $N$ is drawn from a uniform distribution with a range gradually moving from [-0.5, 0.5] to [-0.05, 0.05] from the beginning to 60% of the training (afterwards the range is kept constant).

2) Selector functioning

The selector receives the same input of the experts and has three output sigmoidal units, each encoding the weight to assign to one of the three CPGs. The desired hand joint angles are computed on the basis of the selector output and the output of the three CPGs associated to the experts:

$$out = s_1O_{\text{CPG-C}} + s_2O_{\text{CPG-M}} + s_3O_{\text{CPG-S}}$$  \hspace{1cm} (4)

where $out$ is the vector of the desired joint angles, $O_{\text{CPG-C}}$, $O_{\text{CPG-M}}$, and $O_{\text{CPG-S}}$ are the desired joint angles of the CPGs, $y_{sj}$ is the activation of the selector output unit $j$, and $s_j$ is the normalized selector output unit $j$. $out$ is remapped onto the range of movement of the controlled joints.

3) Critic functioning

The critic receives the same input of the experts and has a linear output unit estimating the evaluation $E$ of the currently perceived state based on the connections weights $w_i$:

$$E = \sum_i (w_i x_i)$$  \hspace{1cm} (5)

4) Critic learning

The critic computes the TD-error that determines whether things have gone better or worse than expected, and uses this to update the weights as follows:

$$\Delta w_i = \eta_E S x_i$$  \hspace{1cm} (6)

where $\eta_E$ is a learning rate (set to 0.1) and $S = R - E$ is the TD-error ("or surprise") with $R$ equal to the reward. $R$ is proportional to the object rotation in each trial, suitably scaled to have values around 1.

5) Actor network learning

The weights of actor experts are updated as follows:

$$\Delta w_{ji} = \eta_A S (y_{ji}^N - y_j) (y_j (1 - y_j) x_i)$$  \hspace{1cm} (7)

where $\eta_A$ is a learning rate, set to 0.1. If $S > 0$ the weights change leads $y_j$ to approach $y_j^N$, if $S < 0$ the weights change leads $y_j$ away from $y_j^N$.

D. The CPGs models used in the system

CPGs were used to produce rhythmic trajectories and were modeled as coupled oscillators each controlling a different joint DOF. A very stable CPG was proposed by Ijspeert [33]. A single oscillator of this CPG can be expressed as follows:

$$\dot{\phi}_j = 2\pi v_i + \sum_j r_{ij} w_{ij} \sin(\theta_j - \phi_i - \phi_j)$$  \hspace{1cm} (8)
\[ \ddot{r}_i = a_i \left( \frac{a_i}{4} (R_i - r_i) - \dot{r}_i \right) \]  
(10)

\[ x_i = r_i (1 + \cos(\theta_i)) \]  
(11)

where \( \theta_i \) is the phase of the CPG oscillator \( i \), \( r_i \) is the amplitude, \( v_i \) is the intrinsic frequency, \( R_i \) is the desired amplitude, \( a_i \) is a positive constant determining how quickly \( r_i \) converges to \( R_i \). \( \Phi_{ij} \) is the desired phase difference (coordinated delay) between oscillator \( i \) and oscillator \( j \) of the CPG, \( w_{ij} \) is establishes the strength of the coupling of \( i \) with \( j \).

The evolution of the phase \( \theta_i \) depends on the intrinsic frequency \( v_i \), on the coupling \( w_{ij} \) and on the phase lag \( \Phi_{ij} \) of the coupled oscillators. Amplitude variable \( r_i \) smoothly follows \( R_i \) with a damped second order differential law.

Ijspeert's CPG was modified as follows to have the possibility of regulating the centre of oscillation of each oscillator:

\[ \dot{\vartheta}_i = 2\pi v_i + \sum_j r_j w_{ij} \sin(\vartheta_j - \vartheta_i - \varphi_{ij}) \]  
(12)

\[ \dot{r}_i = a_i \left( \frac{a_i}{4} (R_i - r_i) - \dot{r}_i \right) \]  
(13)

\[ \ddot{c}_i = \frac{b_i}{4} (C_i - c_i) - \ddot{c}_i \]  
(14)

\[ x_i = c_i + r_i \cos(\vartheta_i) \]  
(15)

where \( C_i \) is the desired center of oscillation of the oscillator \( i \), and \( c_i \) is the actual centre. \( C_i \) allows oscillations around any position of the joint DOF.

As a result, each CPG needs three types of parameters to be controlled (\( R_i \): desired amplitude; \( C_i \): desired oscillation center; \( v_i \): desired oscillation frequency) and one parameter for each coupling (\( \Phi_{ij} \); for simplicity in this simulations \( w_{ij} = 1 \)).

The CPGs used here are shown in Fig. 3. Fig. 3.a shows the complex CPG (CPG-C) having 4 oscillators: N1 generates the desired angle of the flexion/extension of the index (FEI); N2 generates the desired angle of the flexion/extension of the thumb (FET) and N4 generates the desired angle of the opposition of the thumb (OT). In this case the expert produces 12 CPG parameters: \( \nu \) (used for all oscillators), \( R_{1,CPG1} \), \( R_{2,CPG1} \), \( R_{3,CPG1} \), \( R_{4,CPG1} \), \( C_{1,CPG1} \), \( C_{2,CPG1} \), \( C_{3,CPG1} \), \( C_{4,CPG1} \), \( \Phi_{12,CPG1} \), \( \Phi_{13,CPG1} \), \( \Phi_{23,CPG1} \). Fig. 3.b shows the medium complexity CPG (CPG-M) having 2 oscillators: N1 generates the desired angle for both FEI and FET; N2 generates the desired angle for both AAI and OT. The expert of this CPG produces 5 parameters: \( R_{1,CPG2} \), \( R_{2,CPG2} \), \( C_{1,CPG2} \), \( C_{2,CPG2} \), \( \Phi_{12,CPG2} \). Fig. 3.c shows the last simple CPG (CPG-S) having only 1 oscillator, N1, that generates the desired angles for all DOFs. The expert of this CPG produces only 2 parameters: \( R_{1,CPG3} \), \( C_{1,CPG3} \).

### III. RESULTS

The various versions of the model were tested with each of the nine objects separately. Each model was trained for 5000 simulation trials, each formed by 300 steps (3 s of real time). At the beginning of each trial the hand was set at an initial posture with all fingers in a straight position.

#### A. Effects of center of oscillations parameter (C)

The performance of the hierarchical CPG (CPG-H) model using the parameters C to set the center of oscillations was compared with the performance of the same model without such parameters. The resulting performances are reported in
The table shows that the model with C gets rather larger rewards. This is particularly evident for the small objects because without C the finger needs to oscillate in the whole range of amplitude indicated by the R parameter in order to touch the object, so lowering the performance of the system.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Shape</th>
<th>Sphere</th>
<th>Cylinder</th>
<th>Cube</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>C</td>
<td>3.596±0.088</td>
<td>3.621</td>
<td>3.621±0.088</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>3.975±0.085</td>
<td>3.975</td>
<td>3.975±0.085</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>3.781±0.035</td>
<td>3.781</td>
<td>3.781±0.035</td>
</tr>
<tr>
<td>Medium</td>
<td>C</td>
<td>4.768±0.385</td>
<td>4.768</td>
<td>4.768±0.385</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>4.103±0.297</td>
<td>4.103</td>
<td>4.103±0.297</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>4.017±0.071</td>
<td>4.017</td>
<td>4.017±0.071</td>
</tr>
<tr>
<td>Large</td>
<td>C</td>
<td>5.158±0.441</td>
<td>5.158</td>
<td>5.158±0.441</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>4.429±0.074</td>
<td>4.429</td>
<td>4.429±0.074</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>4.555±0.982</td>
<td>4.555</td>
<td>4.555±0.982</td>
</tr>
</tbody>
</table>

Table 2 shows the results of different CPG models with different objects. Each cell indicates the average value and standard deviation of the rewards obtained with four different simulations sequences with the same model. The bold font indicates the best reward value between all proposed CPG models, for every object used. Values in italics indicate the performance of the best runs of each model.

Tab. 2 also shows that the performance is different for the various shapes and dimensions of the object (see also the reward plots of Fig. 5). In particular, data indicate specific regularities that might be tested in experiments with real infants. It can be observed that the performance increases with the decrease of the object dimension, probably because the fingers have to cover a smaller distance to achieve a certain rotation of the objects. The systems achieve the best performance with the spheres, as the systems avoid getting stuck on edges, and achieve the worst performance with the cubes, as the distance that the fingers have to cover to touch them varies continuously.

Figure 5 shows the performance obtained during the simulations with different objects and different models. Each figure contains five curves. Four are related to the models using the single CPGs (CPG-S, CPG-M, CPG-C) or to the hierarchical model (CPG-H Avg): each one is the average of four runs; one is related to the best run with the CPG-H (CPG-H Max).

The most striking result is the fact that the CPG-H is always quite faster to learn than the single-CPG models. This notwithstanding it has many more parameters than them. The reason of this might be that the model can use the CPG that has developed good parameters so far, or the fact that it is more robust to noise (the system can avoid using a CPG whose parameters have deteriorated due to noise). However, further investigations are needed to find the exact cause. It is also interesting to note that in one case (small sphere) the CPG-H succeeds to use both fingers whereas the other models fail to do so. This results in a remarkable higher performance of the hierarchical model. The reason of this is that when the system learns to use two fingers it also learns to alternate them so to impress more rotation to the objects. The fact that in this case the CPG-H succeeds to use two fingers might be the sign that it manages to find and to exploit the advantages of using two fingers in coordination, whereas the other models do not (another sign of the higher facility to learn of the system). The second result, anticipated by discussing Tab. 2, is that the CPG-H always achieves the best final performance with a single run (with the exception of the large sphere and cylinder for which it has a performance similar to the CPG-C).
graphs refer to the four controlled joint DOFs: FEI, AAI, FET, and OT. and real trajectories generated by the CPG-H model. The four couples of graphs shows the desired trajectory a joint generated by the three single-CPG models. Within each couple of graphs, the upper graph shows the desired trajectory, whereas the lower graph shows the real trajectory actually followed by the joints. Various interesting things are evident from the figure. First, all models have managed to find a similar oscillation frequency, probably the good/optimal one. Second, the models, for example the CPG-H, can exploit the possibility of regulating the oscillation centers to have a good contact with the objects (see for example how in the graph of the CPG-H FE T the desired joint angle leads the finger to hit the object and press it so to have a suitable grip on it). Third, recall from above that the higher flexibility of the CPG-H allows it to discover the use of both fingers whereas the other models fail to use the thumb: this is shown by the fact that the desired centers of oscillation of FE T of the single-CPG models are below 0.5 (hence far from the object), whereas that of the CPG-H is above 0.5. Fourth, Fig. 5 showed that by using both fingers the CPG-H could have a higher performance: Fig. 6 indeed shows that CPG-H moves AA T and O T with an opposite phase so as to alternate the fingers contact with the object and keep the object continuously rotating.

Fig. 7 reports the trajectories in the 3D space of the thumb and index tips during the same test of Fig. 6, for the CPG-C and CPG-H models. The figure shows how the hierarchical model manages to bring both fingers in contact of the object by also maintaining a suitable distance from it when regarding the contact. The CPG-C, instead, fails to learn to use the thumb and so brings it far away from the object to avoid interfering with its rotation.
IV. CONCLUSIONS

This research has proposed a hierarchical reinforcement learning model that can be used to study the development of rhythmic manipulation skills in infants. The model resorts to a bio-inspired hierarchical reinforcement-learning architecture. It is grounded on the key idea that rhythmic manipulation movements are generated on the basis of central pattern generators (e.g. located in the spinal cord), and that the parameters of these can be found by trial-and-error mechanisms. The system validation tests show that the model might indeed be a valuable tool to study the development of rhythmic manipulation skills. For example, they show how manipulation skills might develop in various ways, yielding different performances, depending on the different challenges posed by the different objects (alternatively, the results might predict different ages of learning of the manipulation of the different objects considered here).

The hierarchical version of the model was also able to learn rather faster than the simple versions, notwithstanding its higher number of parameters. Notably, it did so by suitably mixing the mixture in a stable fashion. Note that in both the learning runs the reward obtained was approximately the same. This demonstrates that the same task can be solved with different mixed combinations of the CPGs.

Future work will investigate how to make the hierarchical model more robust, in order to avoid occasionally incurring in local minima. Notwithstanding this current limit, the model seems a promising tool to investigate the emergence of rhythmic manipulation skills in primates due to its bio-inspired architecture and the behaviour exhibited so far. This study might be an important complement of the studies on the development of discrete hand movements [32].

ACKNOWLEDGMENT

This research was supported by the EU funded Projects ‘IM-CLeVeR – Intrinsically Motivated Cumulative Learning Versatile Robots’, contract no. FP7-IP-231722 and ‘ROSSI – Emergence of communication in RObots through Sensorimotor and Social Interaction’, contract no. FP7-STREP-216125 and by MIUR project 'OPENHAND - OPEN neuro-prosthetic HAND platform for clinical trials'.

REFERENCES


