Self-Organized Executive Control Functions

Michail Maniadakis, Panos Trahanias and Jun Tani

Abstract— **Executive control incorporates cognitive functions involved in the control and management of other cognitive processes. Such high-level skills are hard to be explored with brain imaging studies because they require complex and persistent experimental procedures. Alternatively, computational modeling may provide a new way to indirectly explore executive control mechanisms. The current work adopts this latter approach to explore possible characteristics of executive control, focusing particularly on behavioral rule switching and confidence neurodynamics in artificial agents. To this end, our study explores a robotic version of the classical Wisconsin Card Sorting Test, incorporating also the option of betting. Our ability to perform multiple and statistically independent computational experiments together with the in-depth study of the mechanisms created in the artificial cognitive systems, provides suggestions for the executive control aspects of the human brain.**

I. Introduction

Executive control functions refer to our ability to monitor and control our own thoughts and behaviors. This type of high level cognitive functions that involve working memory, planning and conflict monitoring are believed to be processed in prefrontal cortex. However, many aspects of this high level cognitive skill remain unknown.

A common way to investigate executive control functions is by using the well known Wisconsin Card Sorting Test (WCST) [1], [2], [3]. According to the WSCT experimental scenario, subjects are invited to repeatedly discover, apply and re-discover a given card sorting rule that is unpredictably changed by the experimenter, based on reward and punishment feedback. The ordinary WCST can be further enriched with the option of betting on behavioral outcomes (i.e., success or failure of sorting) testing the capacity of subjects to implement confidence on the currently adopted sorting rule [4]. Therefore, the WCST-with-betting (WSCTB) is appropriate for investigating complex cognitive processes that include self-monitoring.

The current work explores a robotic version of WCSTB investigating the development of high level cognition in artificial agents. Our task is based on the well known sample-response paradigm. The experimental procedure investigates robot responses for a sequence of trials in order to explore robot's ability to follow and switch along different sample-response rules, as well as to develop confidence about the correctness of the currently adopted rule.

More specifically, a Continuous Time Recurrent Neural Network (CTRNN) [5], [6] implements the artificial brain of a simulated mobile robot [7]. We use an evolutionary procedure to systematically explore CTRNN controllers with rule switching and betting capacity. The exploration of self-organized executive control mechanisms in artificial agents, is expected to provide possible explanations for the cortical neurodynamics supporting natural executive control functionality [8].

In short, our experiments revealed more than one mechanisms capable of executive control. Furthermore, these mechanisms are highly correlated to the different interpretations one may give on the investigated task. Therefore, the findings of the present study suggest that when different subjects understand a given problem in different ways, then it is likely to develop different cognitive mechanisms to solve that problem. This type of personalized cognitive mechanisms are more likely for high-level cognitive functions that rely on knowledge abstraction and prior experiences rather than the lower level skills involved in processing the sensory-motor details of behavior.

Our work clearly distinguishes from previous computational modeling studies addressing rule switching mechanisms, e.g. [9], [10], [11], [12], This is because earlier studies: (i) interpret computationally human hypothesis by hand coding the relevant mechanisms in the model (rather than letting these mechanisms self-organize) (ii) work in a pure theoretical level without being embodied in a robotic agent to interact with the environment. (iii) explore the simple version of the WCST task without considering the option of betting.

The rest of the paper is structured as follows. In section II we describe the CTRNN models used in the current study. In section III we describe the investigated task providing the details of our experimental setup. Then we present the evolutionary procedure used to explore the space of CTRNN solutions. In section V we present the results obtained by the independent evolutionary procedures, and the common characteristics self-organized in all successful solutions. Finally, in section VI we discuss how our findings may apply to biological cognitive processes formulating suggestions about executive control mechanisms in the cortex.

II. CTRNN-based Cognitive Model

We use Continuous Time Recurrent Neural Network (CTRNN) models [6] to investigate how rule switching and confidence mechanisms self-organize in neural dynamics.

Michail Maniadakis is a researcher in the Computational Vision and Robotics Laboratory, Institute of Computer Science, Foundation for Research and Technology – Hellas (FORTH), Crete, Greece. Additionally, he is a visiting researcher in the Laboratory for Behavior and Dynamic Cognition, RIKEN Brain Science Institute, Wako-shi, Saitama Japan. (email: mmaniada@ics.forth.gr).

Panos Trahanias is head of the Computational Vision and Robotics Laboratory, Institute of Computer Science, Foundation for Research and Technology – Hellas (FORTH), Crete, Greece. (email: trahania@ics.forth.gr).

Jun Tani is head of the Laboratory for Behavior and Dynamic Cognition, RIKEN Brain Science Institute, Wako-shi, Saitama Japan. (email: tani@brain.riken.jp).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

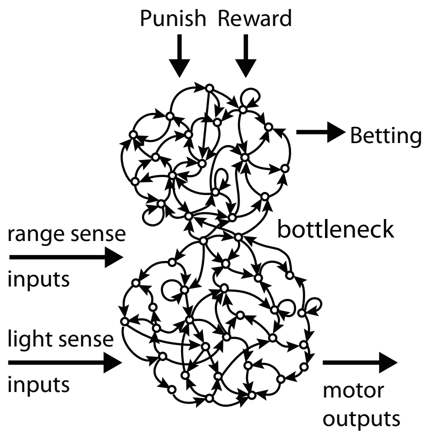


Fig. 1. An abstract schematic representation of the bottleneck CTRNN used in the current study.

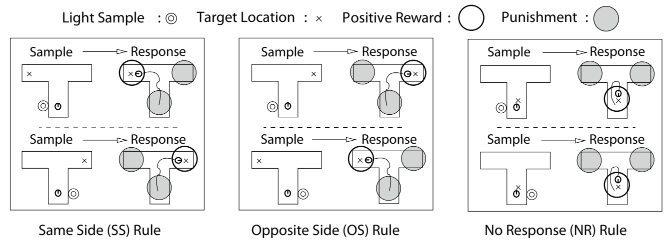


Fig. 2. A graphical interpretation of the three sample-response rules used in our experiments. Each box explains one sample-response rule. In each box, the first line shows correct robot response when light appears to the left side of the robot, while the second line shows correct response when light appears to the right.

In the current implementation, all neurons are governed by the standard leaky integrator equations described in previous studies [13], [7].

Interestingly this type of networks can adequately capture the continuous nature of biological cognition in the cortex. Therefore, in our experimental setup, the neuronal state is initialized only once in the beginning of the first trial, and then neuronal dynamics continue across trials and phases without resetting. In this manner, CTRNNs contextual memory is implicitly represented by internal neurodynamics. We speculate that dynamical states will emerge for representing the rule stored in working memory, while confidence mechanisms will also interact with these representations to decide the amount of betting.

Following our previous study [14] showing that bottleneck configurations [15] are more effective in rule switching tasks compared to fully connected CTRNNs, the current work focuses only on the bottleneck architecture. As shown in Fig 1, we use two bottleneck neurons to separate CTRNN in two levels. The bottleneck neurons loosely segregate information processing in two layers, maintaining minimum interactions between them.

In order to investigate embodied rule switching, we employ a two wheeled simulated robotic agent equipped with 8 uniformly distributed distance, light and reward sensors. The experiments discussed here have been carried out using YAKS[1] a simulated version of the real Khepera miniature mobile robot. The simulator has been slightly modified for the needs of the present study (e.g. by integrating a new sensor-type that supports feeling the special environmental signals simulating negative rewards). To comply with the basic anatomical characteristics of the brain, the lower layer of the CTRNN is linked to the sensors and motors accounting for environmental interaction (this is similar to primary sensory and motor cortices), while the higher layer of the network accepts reward information (that is similar to prefrontal cortex accepting reward from VTA), as it is shown in Fig 1.

[1]The simulator has been developed in the University of Skovde, Sweden, and can be downloaded at http://www.his.se/iki/yaks

III. The Robotic WCSTB Experimental setup

The current study is an extension of our previous work [14] that also addresses rule switching dynamics in a mobile-robot. In the current work, we have incorporated in the experimental setup the option of betting similar to [4], in order to explore the mechanisms involved in confidence development.

A. *Mobile Robot Rule Switching Task*

The task used in the current study is inspired by the rat version of WCST used to investigate the rule switching capacity of rodents based on the sample-response paradigm [16].

The overall task consists of a sequence of trials investigating the capacity of the agent to flexibly manipulate sample-response rules. In the onset of each trial, the robotic agent is located at the bottom of a T-maze environment where it observes a light source turning on, either on its left or right side (see Fig 2). The robot should navigate in the T-maze, responding to the side of the light sample as it is indicated by three sample-response rules. The first is the same-side (SS) rule, implying that the robotic agent should turn left if the light source appeared at its left side, and it should turn right if the light source appeared at its right side. The second rule is the opposite-side (OS), implying that the robot should turn to the side opposite to the light (i.e. right if the light appears to the left side, and left if the light appears to the right). The third is the no-response (NR) rule asking the agent to stay close to the starting position regardless of what the sample signal was.

At any given time, only one of the three available rules is correct. This is specified by the experimenter by properly positioning positive and negative reward signals. As a result, when the agent adopts the right rule giving a correct response in a given trial, it acquires a positive reward. However, in case that the response is not correct the agent receives a punishment. In order to evaluate the capacity of the agent to adopt and successfully follow a given rule, the overall task is split into several trials. The agent is required to find the correct rule (that is specified by the experimenter) and respond according to that rule, in order to be repeatedly rewarded in the sequence of trials.

Turning now to rule switching, the experimenter at a random time (unknown to the robotic agent) changes the rule that is considered correct. This means that the experimenter re-positions positive and negative rewards according to the new sample-response rule. The task for the agent now is to discover this rule change, and switch its response strategy adopting the new rule.

Moreover in the onset of each trial the agent bets for the success of its response in the given trial. Depending on the correctness of the response, the agent gains (or losses) the amount of reward (or punishment) received, multiplied by the amount of betting. Clearly the agent should reduce the betting amount during the rule switching period, and increase betting when the correct rule is successfully followed.

B. Experimental Details

The overall task is structured into $P \in \{1...10\}$ phases, each one consisting of T_p trials. The number of trials $T_p \in \{14, 16, 18, 20, 22, 24\}$ is randomly specified, so that the agent can not predict the end of a phase. The experimenter randomly assigns different correct rules in each phase, which means that during a phase p, the agent must follow the assigned response rule for all T_p trials. Let's assume for example that it should follow the SS rule. In a sequence of trials we test the response of the robot after light sample appearance at its left or right side (their order is randomly chosen). When a trial starts, the robot is sensing the light and stays at the initial position for five simulation steps formulating its response decision and betting for the success of the underlying trial. Then the agent is allowed to move freely in the T-maze, responding to the aforementioned light sample. According to the SS rule, the response is correct when the robot navigates to the end of the corridor and then turns towards the side of the light sample. If the robot makes the correct choice, it drives close to the target location where positive reward exists. In case that the robot turning is not correct, it will drive to a punishment area receiving negative reward indicating that the currently adopted rule is not correct and it should be switched. Depending on the success of the trial the agent gains (or losses) the amount of reward (or punishment) multiplied by the amount of betting. During phase p, the robot is given 10 free of charge exploratory trials to discover what is the correct rule. In the remaining $T_p - 10$ trials the performance of the robotic agent is evaluated in terms of following the desired response rule.

If phase p is completed successfully, the robot moves to phase $p+1$, where the response rule is changed, let's assume to OS. This means that the punishment and reward signals are moved and -for the sake of our example- they are now positioned according to the OS rule. However, the agent is not informed about the rule change and thus, in the first trials of the current phase it will continue responding according to the previous rule. In that case, the agent will drive to a punishment area indicating it is not following the correct rule. Ideally, the agent will realize that the rule has changed and being less confident about the forthcoming response, it will lower its bet in the next trial. In order to avoid punishments in the forthcoming trials, the robot must reconsider its rule choice, exploring alternative response rules, until switching to OS. After that, the agent should increase the amount of betting, in order to acquire more gains. In phase $p+1$, the robot is given again 10 free exploratory trials to discover the new correct rule. In the remaining $T_{p+1} - 10$ trials agent's responses are evaluated according to the currently correct rule.

If phase $p+1$ is completed successfully, the robot moves to phase $p+2$, where the response rule is changed again $-$let say to NR, for our example$-$ and a similar experimental procedure is repeated. Rules are changed in a random order, so that the agent cannot predict their sequence. Overall, the task evaluates agent's switching behavior for a maximum of P phases.

IV. Evolutionary Procedure

In order to explore the self-organization of executive control dynamics in CTRNNs, we use Genetic Algorithms [2]. We are interested in the broader set of mechanisms with the capacity to develop rule switching and self-monitoring, and thus, we do not explicitly specify any internal mechanisms in the model. The network is allowed to self-organize in any appropriate way, developing partial functionalities to accomplish the robotic WCSTB task.

Incremental Evolution. Due to the complexity of the investigated task, it is difficult for the evolutionary process to converge successfully when examining from the very beginning all the details of the problem. In order to support the success of the evolutionary procedure we follow an incremental approach similar to [7], investigating gradually more complex versions of the rule switching task. This is summarized in Table I. In the first generations, the evolutionary procedure aims at CTRNN controllers capable of adopting separately each one of the SS, OS and NR rules. In the forthcoming set of generations, we are interested in exploring all possible switching combinations and thus we explore 6 tasks in total (two tasks per rule). The accomplishment of all six tasks implies that the agent can successfully follow the three available rules, giving successful responses for a long sequence of trials. We note that the very same CTRNN model is evaluated six times (one for each task). At the beginning of each task, the states of all CTRNN neurons are set to zero (i.e. the robot is in a neutral state, without following any rule). The robot explores the environment in order to discover the rule that must be adopted for the successful completion of the single-phase task.

In the next set of generations, the tasks are getting more complex, searching for controllers capable of switching between rules. Specifically, during generations 201-700, we

[2]In the current study, the evolutionary procedure aims at exploring the domain of solutions of the underlying problem, and does not represent an artificial counterpart of biological evolution

TABLE I

THE INCREMENTALLY MORE COMPLEX TASKS EXPLORED IN DIFFERENT PARTS OF THE EVOLUTIONARY PROCEDURE.

Evolutionary Procedure for Rule Switching

Type	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Single Phase	SS	SS	OS	OS	NR	NR
Two Phase	SS ↓ OS	SS ↓ NR	OS ↓ SS	OS ↓ NR	NR ↓ SS	NR ↓ OS
Multi-Phase	SS ↓ OS ↓ NR ↓ ... SS ↓ OS	SS ↓ NR ↓ OS ↓ ... OS ↓ NR	OS ↓ SS ↓ NR ↓ ... NR ↓ SS	OS ↓ NR ↓ SS ↓ ... OS ↓ NR	NR ↓ SS ↓ OS ↓ ... SS ↓ OS	NR ↓ OS ↓ SS ↓ ... NR ↓ SS

explore tasks consisting of two phases, asking for controllers capable of making one rule-switching step, and additionally bet successfully for the given responses (i.e. reduce betting during the transition period, but increase betting when the rules are successfully followed). Note in Table 3, that each task examines a different switching combination among rules. For all six tasks, properly positioned reward and punishment signals indicate the response strategy that the agent should adopt in each trial. The state of CTRNN neurons is reset to zero only once, at the beginning of each task. For all the subsequent steps neural states are kept continuous. This means that special memory pathways have to develop in order to support rule switching.

Finally, during generations 701-1200, we explore the stability of rule switching mechanism. In particular, we investigate the performance of CTRNN controllers under multiple and unpredictable changes of the correct rule as well as the capacity of the agent to reduce betting during rule transition periods, but increase it when rules are correctly followed. All tasks consist of a ten-phase sequence. Rules are randomly assigned to the phases, while the number of trials in each phase is also specified in a random manner. The performance of the agent is evaluated on phase p only if it has adopted the correct rule in phase p-1. Similarly to previous generations, CTRNN is reset to zero at the beginning of each task, and then keeps continuous neural state when passing from one phase to the other.

Fitness Measure. To evaluate the successful accomplishment of the task, we consider two main aspects of robot performance. The first aspect regards rule following and the second the success of betting strategy. In order to evaluate that rules are switched properly and the correct rule is followed at a given trial, target positions are appropriately exploited (see Fig 2). This approach is followed because it is necessary to have a continuous measure for the success of trials (either successful or not). Let's assume that D is the distance between the starting position of the robot and the target. Then, the minimum distance between the target and the robot route can be used for measuring the success of a given robot response. The target positions are specified according to (i) the current rule, and (ii) the side of the light sample, as it is described in Fig 2. Therefore, the changing of rules when we pass from one phase to the other will specify a varying set of target positions. Overall, the ability of the agent to switch (SW) between rules during the p phases of a task i, is measured by:

$$SW_i = \sum_{q=1}^{p} \left(\sum_{t=11}^{T_q} \left(1 - \frac{d_{min}}{D} \right) \right) \tag{1}$$

The evaluation starts from trial $t = 11$ because the first ten trials of each phase are exploratory and they are not considered in evaluation.

Furthermore, we evaluate agent's ability to bet correctly during a sequence of trials. Let us assume that in a given trial t, the agent bets the amount $B_t \in [0, 1]$, while after giving the underlying response the maximum punishment received was $P_t \in [0, 1]$, and the maximum reward received was $R_t \in [0, 1]$. Then the correctness of agent's betting choice (CB) in trial t is defined by:

$$CB_t = \begin{cases} B_t \cdot (R_t - c \cdot P_t), & \text{if } B_t > 0.5 \\ -(1 - B_t) \cdot (R_t - P_t), & \text{if } B_t \leq 0.5 \end{cases} . \tag{2}$$

We assume that the agent is willing to bet if B_t is larger than 0.5, while it avoids betting if B_t is less than 0.5. The first line of eq (2), examines the case that the agent bets (i.e. $B_t > 0.5$). If the agent is rewarded (i.e. R_t is high) it gains a profit, while if the agent is punished (i.e. P_t is high) it has a loss. High values of R_t imply low values of P_t and vice versa. The weighting coefficient for punishment is set to the relatively large value of $c = 6.0$ making the agent to reduce betting during the rule transition period. Low values of the weighting coefficient (e.g. $c = 1$) make the agent develop an "always-bet" strategy. In the second part of eq (6) we examine the case of avoiding betting (i.e. $B_t \leq 0.5$). When the response given by the agent is incorrect (P_t is high), the no-betting choice was right, and the agent makes profit. However, if the response given by the robot was correct (R_t is high), then "avoid-betting" choice was incorrect, and the agent has a loss of possible profit. Overall, for a task i described by a sequence of p phases, the capacity of the agent to bet efficiently (BET) is evaluated by the partial fitness measure:

$$BET_i = \sum_{q=1}^{p} \left(\sum_{t=1}^{T_q} CB_t \right) \tag{3}$$

The overall success of the agent on accomplishing the task $i \in \{1, 2, \ldots 6\}$, is obtained by the multiplication of SW_i and BET_i with a weighting coefficient d:

$$E_{Task_i} = (SW_i) \cdot (BET_i)^d \tag{4}$$

In the first stage of incremental evolution (i.e. generations 1-200) we use d=0, emphasizing the acquisition of rules. In the second stage of evolution (i.e. generations 201-700) d=0.5 making the agent to consider both rule switching and betting. In the last stage, (i.e. generations 701-1200) we use d=2.0, which makes evaluation focus on betting, considering

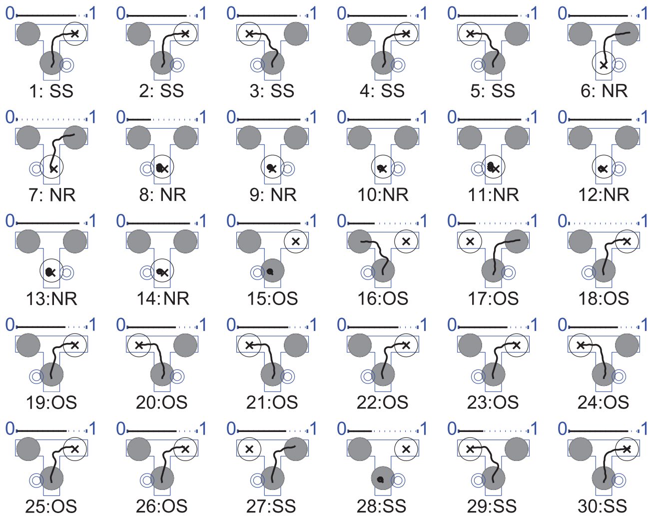


Fig. 3. The behavior of the agent in a sequence of trials. The line on top of each trial demostrates the current amount of betting. Light is depicted with a double circle, goal position is depicted with an $\times$, punishment area is depicted with a gray circle, while robot path is depicted with a black line starting from the bottom of the T-maze. In the present figure we follow a more compact representation of a sample-response trial than the one shown in Fig 2, in order to demonstrate an adequately large number of robot trials.

also that the rule switching capacity of the agent must be preserved.

All individuals encoding CTRNN controllers are tested on the incrementally more complex versions of Task1, Task2, Task3, Task4, Task5, and Task6 described above. The accomplishment of each task is evaluated separately according to eq (4). The total fitness of the individual is estimated by:

$$fit = \prod_{j=1}^{6} E_{Task_j} \tag{5}$$

The multiplication operator favors individuals that can accomplish (at least partly) all tasks, distinguishing them from those that fail in any one of them.

V. Results

In order to explore possible neuronal mechanisms accounting for executive control functions related to rule switching and confidence development, we have conducted 14 statistically independent runs of the evolutionary procedure described above. Six of these procedures converged successfully, producing robot controllers that can effectively switch rules and bet correctly accomplishing the WCSTB task.

An example sequence of robot trials together with the rule changes made by the experimenter is shown in Fig 3. In the first five trials the agent successfully follows the SS rule receiving rewards. The agent bets maximally with full confidence on its rule-choice. Then in the 6th trial, the experimenter changes the rule to NR. The robot that is not aware of this change responds according to the SS rule and is punished. Immediately after that, the amount of betting decreases, implying weakening of agent's confidence about the currently correct rule. After two explorative trials, the agent finds that NR is now the correct rule, receiving positive reward (in trial 8). Subsequently, its confidence to the currently adopted rule is strengthened, and thus the amount of betting increases. The agent follows the NR rule for some more trials giving successful responses. Then in trial 15, the rule is unexpectedly changed again, and the agent gives a wrong response which makes the amount of betting to fall down. The agent identifies the correct rule receiving a positive reward at trial 18. Then its confidence increases, and in the next trial it bets high. In subsequent trials, the agent responds following the OS rule, receiving rewards. The experimenter changes the rule again in trial 27. It takes two more trials to the agent to identify that now SS is the correct rule. In the following trials, the agent increased adequately the amount of betting, responding successfully according to SS rule. Overall, the figure shows that the robot successfully adapts the response strategy to the rules specified by the experimenter, after a short transition period of erroneous responses.

We have investigated the internal dynamics of the CTRNN solutions in order to obtain insight into the cognitive mechanisms self-organized in the successfully evolved models. We found that artificial evolution generated two broad categories of networks in which self-organized neural dynamics are qualitatively different (see below). For the sake of clarity of the current presentation, we will refer to these CTRNN categories as “Type-A” and “Type-B”[3].

A. *Layered functionality*

Initially we studied the functional differences of CTRNN layers in order to determine their functionality in the global network. We perform Principal Component Analysis (PCA) to highlight the main characteristics of neural activity in each layer. The first principal component of high and low layer neural activation for the two types of solutions is shown in Fig 4 where different rules are depicted in different colors (i.e. red:SS, green:OS, blue:NR). We observe that the activity of the higher layer is much more stationary compared to low layer activation. This difference suggests the specialized functionality of each layer. The rather fast fluctuation of the low layer suggests it is dealing with the sensory-motor issues arising from real-time environment interaction, while the higher part of the CTRNN is probably involved in encoding the currently adopted rule as well as in estimating the confidence of the agent in order to decide betting.

B. *Rule Encoding*

One more observation form Fig 4 is that neural activity in the higher layer encodes rule NR by using nearly constant and distinct values, while the representation of rules SS and OS are less differentiated. To explore further this issue, we have taken the phase plots of the first two principal components of high layer activity, shown in Fig 5. We see three trajectories of quasi-attractors to appear, each one encoding one of the available rules.

[3] We note that our findings do not exclude the possibility that more solution types may exist for the underlying problem.

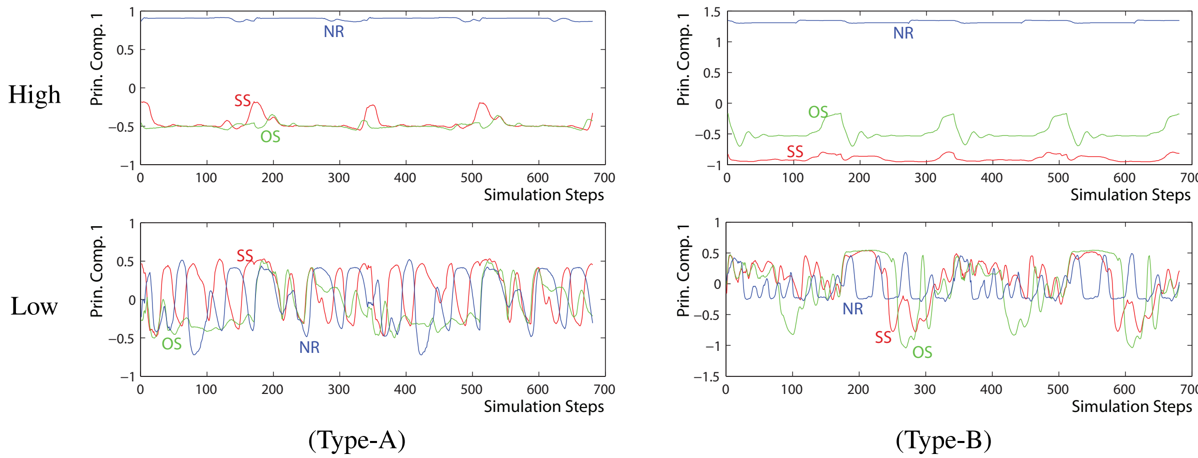


Fig. 4. The unfolding of the first Principal Component of neural activity in the high and low layer of the CTRNN for 4 consequtive trials (i.e. 680 simulation steps). Different colours correspond to neural activity when the agent is successfully following different rules. In particular, the principal component of neural activity for rule SS is shown in red, for rule OS is shown in green, while for rule NR is shown in blue.

Clearly, in the case of Type-A solution there is a partial overlap between the trajectories encoding SS and OS rules (i.e. trajectories shown in red and green) while NR is represented by a distinct attractor (i.e. blue trajectory). However, in the case of Type-B solution phase plot reveals attractors akin to three different fixed points with a clearly separate representation of each rule. We note that *the afore mentioned distinction characterizes the obtained CTRNN solutions as Type-A (i.e. with SS, OS overlap) or Type-B (i.e without overlap)*. In the totally 6 successful evolutionary runs, solutions of Type-A appeared 4 times, while solutions of Type-B appeared 2 times.

The overlap of SS and OS attractors in the case of Type-A solution (see Fig 5) suggests that these rules are organized as subclusters of a larger cluster separating them from NR. This organization is reasonable since SS and OS exhibit common characteristcs when they are both contrasted to NR. In particular, both SS and OS ask the agent to travel along the corridor and turn left or right, while NR asks the agent to ignore sample stimulus and stay close to the starting position (see Fig 2). As a result, the approach followed by the agent in the case of Type-A solution focusing on the differences of SS and OS to NR, is particularly appropriate for the investigated problem. On the contrary, the plot corresponding to Type-B solution (Fig 5), shows a clearly distinct representations for all three available rules. We would like to emphasize that this organization is also reasonable, since each of the three rules is actually standalone and may exists without the others. The completely separate representation of rules SS and OS highlights their independent nature, while at the same time they both remain separate from NR.

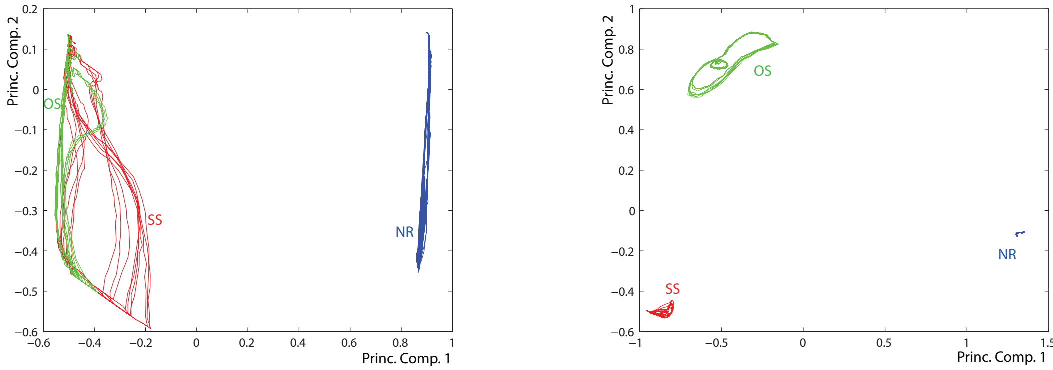
In summary, the representations of rules self-organized in Type-A and Type-B solutions reflect the different interpretations one can give to the rule-switching problem investigated in the present work by either focusing on the relation of SS and OS compared to NR, or the unique identity of each rule.

C. *Rule Switching*

Next we examine the rule transition mechanisms developed in CTRNNs. We note that the neurodynamic phenomena related to the switching of rules have been discussed in detail in [7], [14], therefore, here we concentrate on how rule representation differences between Type-A and Type-B solutions affect rule transitions. To this end, we consider neural activation in the early part of trials. This is a context-rich period, because at that time the agent decides its response according to the currently adopted rule and additionally decides the amount of betting.

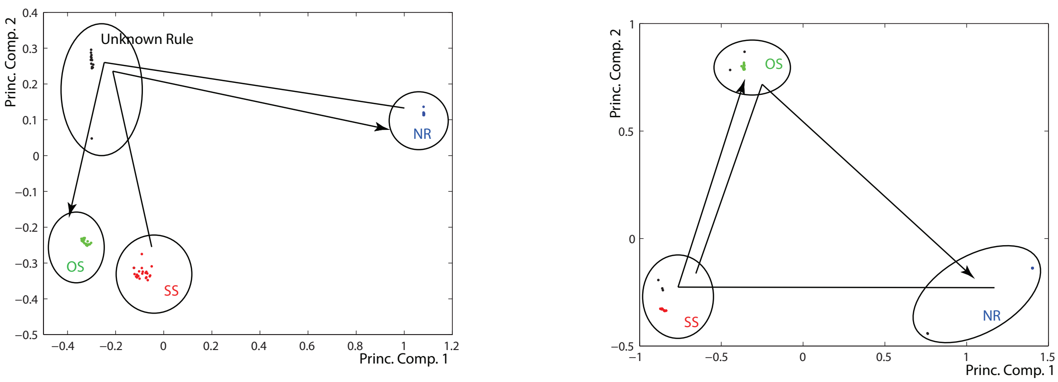
We have estimated the average, over the first 15 simulation steps of a trial, for the first two principal components of neural activity in the higher layer of CTRNNs. A 2-D plot of the estimated averages for 68 trials is shown in Fig 6, both for Type-A and Type-B solutions. In this sequence the agent starts by following rule SS, then adopts NR and finally OS. In the trials that the agent successfully follows a given rule, the 2-D plot illustrates points in red, green and blue colors (depending on the rule). During the transition trials where the agent gives erroneous responses (trying to identify the correct rule), the points arising from the principal component averages are illustrated in black. Interestingly, we observe that different transition mechanisms are developed for the Type-A and Type-B solution.

In the case of Type-A solution we see that a common rule transition area is formulated that corresponds to the 'unknown rule' state. When the experimenter unpredictably changes the (currently correct) rule, the agent that is not aware of this change gives erroneous responses and thus it is punished. The punishment received causes an instability in the high layer of the CTRNN which makes neural activity move in the 'unknown rule' area. From this state, the agent randomly selects a rule to be applied in the next trial (a rule might be selected more than one times even if it is not correct). If the rule choice proves to be successful, the agent adopts it for the forthcoming trials. However, if the selected rule is incorrect then the network remains in the



(Type-A) (Type-B)

Fig. 5. Phase plot of higher level neural activity when the agent follows (a) the SS rule and (b) the OS rule. Neural activities stabilize to attractors having distinct shapes for each case.



(Type-A) (Type-B)

Fig. 6. The averages of the first two principal components in the begining of trials. The plots associated to Type-A and Type-B solutions correspond to a sequence of trials investigating rule transitions from SS to NR and then to OS.

’unknown rule’ state, selecting a new rule in the next trial. As an example, Fig 6 shows SS→NR and NR→OS switchings. In addition, after testing every possible combinations of rule switching, we observed that all transitions always pass through the unknown rule area.

In the case of Type B solution, a different transition mechanism is self-organized in CTRNN as it is shown in Fig 6. For the given example investigating SS→NR→OS, when the agent is successfully following rule SS and the experimenter is unpredictably changing the rule, the instability caused by punishment signals makes the adopted rule jump to OS. However, it happens that rule OS is not correct. Then in the next trial the robot is punished again, which makes CTRNN rule state jump to NR, that is the correct rule. A similar procedure is also observed when the experimenter changes the rule to OS. The punishments provided to the agent make the rule state jump first to SS and then to OS. Overall, we observe that in the case of Type B solutions there are direct transitions from one rule to the other, following a circular organization. We note that additinal expriments revealed that circular transitions apply for all possible rule switching combinations).

D. Betting Mechanism

Finally, we have investigated confidence mechanisms providing agent the capacity to bet successfully while switching among sample-response rules. Our findings suggest different betting strategies for the two types of solutions. In particular, for Type-A solution, when the agent is in a rule exploration mode (i.e. unknown rule state in Fig 6) its betting choice is always the same without any correlation to the rule currently tested. This is summarized in the second column of Table II showing betting amounts during rule testing. When the agent receives a reward approving a rule, then betting strategy differentiates depending on rule as it is illustrated in the third column of Table II.

However, for the Type-B solution that is based on direct transitions among rules, the betting strategy shows different characteristics. In particular, when rules are assessed to determine their correctness the agent needs to minimize betting in order to avoid the loss of gains. For Type-B solution, during these testing trials, the agent differentiates betting depending on the rule assessed. This is summarized in the fourth column of Table II, listing the amounts of betting during rule testing. When the agent receives reward approving a rule, the betting strategy remains differentiated as it is illustrated in the fifth

TABLE II

THE RANGE OF BETING AMOUNTS WHEN THE AGENT TESTS OR FOLLOWS EACH RULE, FOR THE TWO TYPES OF OBTAINED SOLUTIONS.

	Type-A		Type-B	
Rule	**Test Betting**	**Rule Follow Betting**	**Test Betting**	**Rule Follow Betting**
SS	[0.18-0.23]	[0.58-0.76]	[0.41-0.47]	[0.89-0.93]
OS	[0.18-0.23]	[0.41-0.52]	[0.01-0.05]	[0.71-0.77]
NR	[0.18-0.23]	[0.96-0.99]	[0.34-0.38]	[0.97-0.99]

column of Table II.

Overal, our observations indicate that confidence interpretation in the CTRNN is directly correlated to the possible views that may be developed on a given problem, as well as the characteristics of the neural mechanisms supporting the solution of the problem.

VI. DISCUSSION

In the current work we investigate executive control functions putting them in the context of artificial agents. Our study follows a minimum constraint approach that avoids assigning predefined roles at different parts of the artificial cognitive system. Examining the internal neurodynamics of CTRNNs we found two different types of solutions self-organized in the models.

Interestingly, we observed that a loose segregation of system components by means of bottleneck architectures facilitates the emergence of different roles in each part of the system, and the self-organization of functional hierarchies. In the obtained results, higher layer is involved in the manipulation of sample-response rules, while the lower part takes care of environment interaction issues.

The evolutionary self-organization of CTRNNs revealed two possible mechanisms accounting for high level executive control in WCSTB. The relevant mechanisms arise from two different interpretations one may give to the problem investigated in the current study (i.e. according to the similarity of SS and OS when compared to NR, or, according to the standalone nature of the three rules). Therefore, our findings suggest for biological cognitive systems that the way a task is understood by a human subject is likely to affect the development of the relevant dynamics in his brain. In other words, when two subjects understand a given problem in different ways, then they may use cortical resources in different ways when solving the problem. This is a novel way to approach high level executive control functions in the cortex that is rarely considered in neuroscientific studies.

We note that we have also explored how the low layer of CTRNNs specialize for Type-A and Type-B solutions. However, we have been unable to identify clear specialized characteristics. Therefore, our experiments suggest that high level cognitive functions are more likely to differentiate among subjects than low level processes. Intuitively this view is supported by the fact that low level processes are less plastic because they are linked to the phylogeneticaly hard coded characteristics of the sensory-motor system, while high level cognition has enough freedom to flexibly self-organize in the cortex considering prior experiences and knowledge. Our suggestion is further supported by the argument that high level cortical areas far from the primary cortices show increased flexibility when adopting their functionality [17].

VII. CONCLUSIONS

We have adopted an evolutionary robotics approach to explore posible characteristics of executive control functions. Our findings suggest that the mechanisms involved in excutive control may depend on the interpretations that humans may give to a particular problem.

In the future we will investigate further the betting mechanisms self-organized in CTRNNs in order to obtain better insight on the possible slf-monitoring mechanisms of the human brain.

REFERENCES

[1] B. Milner, "Effects of different brain lesion on card sorting." *Archives of Neurology*, vol. 9, pp. 90–100, 1963.

[2] K. Greve, T. Stickle, J. Love, K. Bianchini, and M. Stanford, "Latent structure of the Wisconsin Card Sorting Test: a confirmatory factor analytic study," *Archives of Clinical Neuropsychology*, vol. 20, pp. 355–364, 2005.

[3] F. Mansouri, K. Matsumoto, and K. Tanaka, "Prefrontal cell activities related to monkeys's success and failure in adapting to rule changes in a wisconsin card sorting test analog." *The Journal of Neuroscience*, vol. 26, pp. 2745–2756, 2006.

[4] D. Koren, L. Seidman, and P. Harvey, "Real-world cognitive–and metacognitive–dysfunction in schizophrenia: a new approach for measuring (and remediating) more right stuff." *Schizophrenia Bulletin*, vol. 32, no. 2, pp. 310–326, 2006.

[5] K. Doya and S. Yoshizawa, "Memorizing oscillatory patterns in the analog neuron network." in *Proc. of 1989 Int. Joint Conf. on Neural Networks, (IJCNN-89)*, 1989, pp. 27–32.

[6] R. Beer, "A dynamical systems perspective on agent-environment interaction." *Artifficial Inteligence*, vol. 72, pp. 173–215, 1995.

[7] M. Maniadakis and J. Tani, "Acquiring rules for rules: Neuro-dynamical systems account for meta-cognition." *Adaptive Behavior*, vol. 17, pp. 58–80, 2009.

[8] E. Ruppin, "Evolutionary autonomous agents: A neuroscience perspective," *Nature Reviews Neuroscience*, vol. 3, no. 2, pp. 132–141, 2002.

[9] S. Dehaene and J. Changeux, "The wisconsin card sorting test: theoretical analysis and modeling in a neuronal network." *Cerebral Cortex*, vol. 1, no. 1, pp. 62–79, 1991.

[10] R. O'Reilly, "Biologically based computational models of high-level cognition." *Science*, vol. 314, pp. 91–94, 2006.

[11] P. Dayan, "Bilinearity, rules, and prefrontal cortex." *Frontiers in Computational Neuroscience*, vol. 1, pp. 1–14, 2007.

[12] A. Stemme, G. Deco, and A. Busch, "The neuronal dynamics underlying cognitive flexibility in set shifting tasks." *Journal of Computational Neuroscience*, vol. 23, pp. 313–331, 2007.

[13] B. M. Yamauchi and R. D. Beer, "Spatial learning for navigation in dynamic environment," *IEEE Trans. Syst. Man Cybern.*, vol. 26, no. 3, 1996.

[14] M. Maniadakis and J. Tani, "Dynamical systems account for meta-level cognition," in *10th Int. Conf. on the Simulation of Adaptive Behavior (SAB-08)*, 2008.

[15] R. Paine and J. Tani, "How hierarchical control self-organizes in artificial adaptive systems," *Adaptive Behavior*, vol. 13, no. 3, pp. 211–225, 2005.

[16] D. Joel, I. Weiner, and J. Feldon, "Electrolytic lesions of the medial prefrontal cortex in rats disrupt performance on an analog of the Wisconsin Card Sorting Test, but do not disrupt latent inhibition: implications for animal models of schizophrenia," *Behavioural Brain Research*, vol. 85, pp. 187–201, 1997.

[17] J. Fuster, *Cortex and mind: unifying cognition*. New York: Oxford University Press, 2005.