(Accepted in Neural Networks, 2012)

Self-organizing high-order cognitive functions in artificial agents: implications for possible prefrontal cortex mechanisms

Michail Maniadakis[a,b], Panos Trahanias[a], Jun Tani[b,c]

[a]*Brain Science Institute, RIKEN, Wako-Shi, Saitama, Japan*
[b]*Institute of Computer Science, FORTH, Heraklion, Crete, Greece*
[c]*Electrical Engineering Department, Korea Advanced Institute of Science and Technology, Yuseong-gu, Daejeon, Korea*

Abstract

In our daily life, we often adapt plans and behaviors according to the dynamically changing world circumstances, selecting activities that make us feel more confident about the future. In this adaptation, Prefrontal Cortex (PFC) is believed to have an important role, applying executive control on other cognitive processes to achieve context switching and confidence monitoring; however, many questions remain open regarding the nature of neural processes supporting executive control. The current work explores possible mechanisms of this high-order cognitive function transferring executing control in the domain of artificial cognitive systems. In particular, we study the self-organization of artificial neural networks accomplishing a robotic rule-switching task analogous to the Wisconsin Card Sorting Test. The obtained results show that behavioral rules may be encoded in neuro-dynamic attractors, with their geometric arrangements in phase space affecting the shaping of confidence. Analysis of the emergent dynamical structures suggests possible explanations on the interactions of high and low level processes in the real brain.

Keywords: Neural Networks, Executive control, Meta-cognition, Prefrontal Cortex, Cognitive Robot, Attractor, Self-organization

1. Introduction

A well-known experiment investigating executive control functions and more specifically, rule switching, is the Wisconsin Card Sorting Test (WCST), Berg (1948); Milner (1963), where subjects are asked to discover and apply a card sorting rule based on reward and punishment feedback. In unpredictable times during the task, the rule is changed by the experimenter and must be re-discovered by subjects. The ordinary WCST can be further enriched with the option of betting on behavioural outcomes (i.e., success or failure of sorting). The WCST-with-Betting (WCSTB) tests the capacity of subjects to monitor and implement confidence about the currently adopted rule, Koren et al. (2005, 2006). This is a high-level cognitive task which requires coordinating a range of different processes, including the maintenance of working memory for the currently followed rule, the examination of conflicts between the adopted rule and the reward or punishment feedback, higher level executive control for rule adjustment, self-monitoring for confidence development, betting decisions and generation of physical actions on the basis of the selected rule.

Existing modelling studies on WCST employ discrete and algorithmic computational processes, based on the common assumption that, although the posterior cortices can be characterized as fundamentally analog systems, the prefrontal cortex (PFC) has a more discrete, digital character, O'Reilly (2006); Dayan (2007). Previous modelling studies, e.g. Dehaene and Changeux (1991); Stemme et al. (2007a), employ local and discrete neural network representations where currently adopted rules are represented by separately activated local units. Rougier and O'Reilly (2002) proposed an on-off type gating operation that acts on working memory for storing the currently adopted rules. The essential idea is that the neural activation patterns representing the current rules in the working memory can be preserved by closing the gate until the rules are in conflict with the new rule selected by the experimenter. Dayan (2007) generalized this model to deal with various executive control functions, assumed to be present in PFC, employing a computational scheme of conditional rule matching and action execution. Other relevant models interpreting computationally human assumptions about rule switching work also on the basis of discrete states for rule representation based on either petrinets, Narayanan (2003), or pools of excitatory and inhibitory neurons, Stemme et al. (2007b), or Hopfield neural network with a separate hypothesis generation module, Kaplan et al. (2006).

An alternative approach regards implementing cognitive capacities based on dynamic neural mechanisms. In this direction, a variety of computational models have interpreted computationally many of the well known PFC functionalities. For example, dynamic working memory models are investigated in Botvinick and Plaut (2006) showing that recurrent neural networks can adequately accomplish serial recall tasks considering also the effects of background knowledge. Additionally, Machens et al. (2005) have investigated interval discrimination tasks showing that attractor dynamics can effectively combine memory maintenance and decision making, while, Johnson et al.

(2009) have implemented an analogous model for visual working memory.

The discussion above highlights the main directions in the open debate regarding the discrete or dynamic nature of PFC processing (see also Brody et al. (2003)). The first type of models have been mainly inspired by experimental data showing active and relatively long lasting neural activity that may encode rules in working memory. However, an important drawback for the discrete models concerns how the static (usually bi-stable) representations can link executive control with the non fully predefined and inherently continuous behaviors of the agent. The dynamic approach has gained significant support from experimental works showing that PFC processing is based on time-dependent activation patterns, Singh and Eliasmith (2006); Romo and Salinas (2003), as well as dynamic interaction networks in the brain, Palva et al. (2010). Especially for rule switching, experimental electrophysiological data from monkeys trained to perform WCST showed that the DLPFC cells encode rules through dynamically changing neural activities, Mansouri et al. (2006). The observed dynamical patterns may be ascribed to the cognitive processes taking place when accomplishing a delayed response tasks such as external stimuli processing, decision making, response planning, motion execution monitoring etc. (see Jun et al. (2010) for multi-process coordination in PFC). The studies mentioned above indicate that PFC internal mechanisms are based on a dynamic rather than a stationary pattern of neural activity. In other words, active maintenance of neural activity, does not necessarily mean static representations. The present work aims to examine whether the executive control functions involved in WCSTB can be implemented on the basis of dynamic processing and whether such a possibility provides the basis for a new understanding of high-level cognitive functions. More specifically, the dynamic modelling approach suggests explanation for the confidence and preference people show for certain situations (i.e. we feel more confident when turning in a street cross of our hometown, than in a street cross of another town), as well as how our minds organize rules into classes using some type of similarity criteria.

The computational exploration of alternative mechanisms can be based on evolutionary robotic experiments similar to Borrett et al. (2005). This is because the real-time environmental interaction may provide more realistic and general explanations on executive control processes compared to the purely theoretically operating existing models. In particular, the current work employs a minimum constraint modelling approach to explore possible mechanisms of executive control functionality self-organized in simple neural network models achieving the WCSTB task. If the mechanisms for accomplishing the task consistently appear in statistically independent simulation runs, comparable principles may also operate in real brains, Ruppin (2002). In short, neural network models with recurrent connectivity are evolved to accomplish a robotic version of the WCSTB, using a standard genetic algorithm to search for optimal synaptic weights, Nolfi and Floreano (2000); Lipson (2005). Following this approach, neural dynamics are free to self-organize in any appropriate way, revealing new and potentially more natural mechanisms for explaining high-level cognition, Baev (2007). We study the successfully evolved neural network models identifying their common internal characteristics, in order to provide suggestions of possible working principles in the brain.

In contrast to previous studies that focus on WCST exploring pure rule-switching, the option of betting that is additionally investigated in the present work provides a means for the deeper exploration of executive control functions. In particular, our experiments investigate the self-awareness capacity of the artificial agent that regards monitoring (i) the current behavioural context (i.e. the agent being in either a rule exploration or a rule following mode) and (ii) the confidence that the agent feels for each behaviour and how the latter affects its betting strategy.

The rest of the paper is organized as follows. In the next section we present the methodology followed in our work. In particular, we present the Continuous Time Recurrent Neural Network (CTRNN) model used in our study, how it is connected to the sensors and actuators of the simulated robotic agent, the computational counterpart of the WCSTB problem, and the evolutionary procedure used to explore configurations of CTRNN robot controllers. Experimental results addressing robot switching and betting on the basis of three alternative behavioural rules are presented in the following section. Then, a detailed discussion highlights the main finding of our computational experiments, formulating suggestions for the organization of biological executive control processes. Finally, conclusions and suggestions for further work are presented in the last section.

2. Experimental Methodology

In order to investigate executive control dynamics, we have designed a robotic task that resembles Wisconsin Card Sorting test incorporating also a betting option for the artificial agent. The task investigates rule switching in a sample-response paradigm, similar to Joel et al. (1997). The agent has to learn three sample-response rules, selecting, applying and re-selecting each one of them, as indicated by reward and punishment signals provided by the experimenter. The three available rules named Same Side (SS), Opposite Side (OS) and No Response (NR), are shortly described in Fig 1. The robot starts always from the bottom of the T-maze environment, responding to the side of light presentation. According to the Same-Side (SS) rule, the agent must turn left if the light source appeared at its left side, and it must turn right if the light source appeared at its right side. According to the Opposite-Side (OS) rule, the robot has to turn to the opposite direction of the light side, i.e. right when light appears to the left, and left when light appears to the right. In the case of the No Response (NR) rule, the robot should ignore the side of light staying close to the starting position.

The task explored in the current study is separated in phases, each one consisting of several sample-response trials. While performing the trials of a given phase, the robotic agent has to discover and follow the sample-response rule that is assigned to the phase. Correct responses are rewarded, while incorrect are punished (see Fig 1). At the beginning of each trial, the agent bets for the success of the underlying response, having the opportunity to gain some profit.

Different phases correspond to different rules which requires the agent to switch the adopted response strategy. Changes from one phase to another are performed by the experimenter in a random manner. This results to unpredictable rule changes that make the agent give spontaneous incorrect responses. Therefore, the agent has to develop mechanisms that consider rule changes, switch the adopted response strategy and efficiently control the amounts of betting, in accordance with the dynamically changing circumstances of the experimental setup, in order to successfully accomplish the overall task.

In the current set of experiments, instead of hand designing the mechanisms (for accomplishing the above mentioned task) in the model, we evolved Continuous Time Recurrent Neural Network (CTRNN) robot controllers, which can freely self-organize their internal dynamics. This approach has the potential to reveal possible mechanisms accounting for executive control functions, Ruppin (2002). We have conducted multiple statistically independent runs using both fully connected and bottleneck (Paine and Tani (2005)) CTRNN topologies, in order to investigate (i) the appropriateness of the network structure and (ii) the self-organization of internal network dynamics encompassing rule-switching and self-monitoring capacity.

2.1. Behavioural Task

The overall task is structured into $P \in \{1...10\}$ phases, with each phase including 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. During a phase p, the agent has to follow the same response rule for all T_p trials. All trials have a predefined length of 170 simulation steps that is sufficiently large to ensure that the agent can move in any location of the T-maze. The external reinforcements (reward and punishment signals) are provided after the elapse of 100 simulation steps so that agent is not rewarded from the very beginning of the trial in the case of the NR rule.

Let's assume for example that the agent should follow the SS rule. Each trial tests 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 given trial. Then the agent is allowed to move freely in the T-maze, responding to the aforementioned light cue. 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 (i.e. the agent is normally punished or rewarded but this does not count in the evaluation procedure, see eq (5)). In the remaining $T_p - 10$ trials the performance of the robotic agent is evaluated in terms of following the desired response rule. If all the responses provided by the agent in these trials are correct, phase p is considered successful and the agent moves to the next phase. Otherwise, the agent is immediately interrupted and is evaluated for its performance so far.

When the agent enters into phase $p + 1$, the response rule is changed, let 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 feeling less confident about the forthcoming response, it will bet low 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 behaviour for a maximum of P phases.

2.2. CTRRN model and its Connectivity to the Robot

In order to investigate how rule switching and monitoring mechanisms self-organize in neuronal dynamics for the task described above, we use a Continuous Time Recurrent Neural Network (CTRNN), Doya and Yoshizawa (1989); Beer (1995), to control a simulated robotic agent. This type of artificial neural networks provide an adequate framework for investigating the temporal characteristics of cognitive functionality, Van Gelder (1998).

In the current study, we employ both bottleneck (Paine and Tani (2005)) and fully connected CTRNN topologies. As shown in Figure 2(a), a CTRNN is squeezed in the middle with the upper and lower parts of the network interacting only through bottleneck neurons, partially segregating information processing in different levels, maintaining minimum interactions between them. The lower part receives the sensory flow and outputs motor flow, while the higher part receives positive and negative reward stimuli and outputs betting rate. In contrast, in the fully connected case (see Figure 2(b)) information processing levels can hardly differentiate. All neurons are governed by the standard leaky integrator equations described in previous studies, Yamauchi and Beer (1996); Paine and Tani (2005):

$$\frac{d\gamma_i}{dt} = \frac{1}{\tau}\left(-\gamma_i + \sum_{k=1}^{R} w^s_{ik} I_k + \sum_{m=1}^{N} w^p_{im} A_m\right) \tag{1}$$

where γ_i is the state (cell potential) of the $i-th$ neuron. All neurons in a network share the same time constant τ in order to avoid explicit differentiation of CTRNN parts. The state of each neuron is updated according to external sensory input I weighted by w^s , and the activity of pre-synaptic neurons A weighted by w^p . The activation of the $i - th$ neuron is then estimated by the non-linear sigmoid function according to:

$$A_i = \frac{1}{1 + e^{-(\gamma_i - \theta_i)}} \tag{2}$$

where θ_i is the activation bias applied on the $i - th$ neuron.

One important characteristic of the CTRNN, is the contextual memory being implicitly represented by internal neurons dynamics. In our experimental setup, the neuronal state γ_i is initialized only once, and then neuronal dynamics continue for the rest of the steps of robot behaviour (i.e. without resetting γ_i). This characteristic is necessary in order to relate the internal dynamics of the model with the rate coding of neuron populations in the brain, and the temporal nature of cortical processing.

In order to investigate embodied rule switching and self-monitoring mechanisms, we employ a two wheeled simulated robotic agent equipped with 8 uniformly distributed distance, and light sensors and two front sensors receiving positive or negative reward stimuli. The robotic platform is based on YAKS environment, which simulates motion kinematics of the Khepera robot. The simulator has been slightly modified for the needs of the present study. In particular, the reward and punishment emitters which were not provided by YAKS have been implemented as circular areas where the amplitude of reward (in the range $[0, 1]$) is linearly increasing from the perimeter to centre.

For the case of the bottleneck architecture, the CTRNN affects the behaviour of the robot by means of one lower level motor neuron that specifies the speed of the robot, and one higher level betting neuron that specifies the risk that the agent takes for the success of a given trial. For the case of the fully connected architecture, the motor neuron and the betting neuron are randomly selected from the whole set of available neurons.

The activity of the motor neuron sets the speed of the left and right robot wheels. Let assume that at a given time step s, the activation of the motor neuron is A_m^s. Then, the instant left and right wheel speed of the simulated robot is given by:

$$speed_l = 0.4 + 0.6A_m^s \qquad speed_r = 0.4 + 0.6(1 - A_m^s) \tag{3}$$

Following this approach the agent moves with a constant total speed, while the activation of the motor neuron controls the direction of movement.

Additionally, the activity of the betting neuron in the first five simulation steps of each trial determines the amount of risk that the agent undertakes for the success of the forthcoming response. In particular, let assume that the activity of the betting neuron in the $s - th$ simulation step of trial t, is A_b^s. Then the amount of betting for the underlying trial is given by:

$$B_t = \sum_{s=1}^{5} A_b^s \tag{4}$$

The agent will gain (loose) an amount in the underlying trial that is determined by B_t and the reward (punishment) received after the given response.

2.3. Evolutionary Procedure

In order to explore the self-organization of executive control dynamics in CTRNNs, we use Genetic Algorithms[1]. 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 executive control functions, 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 convergence of the procedure we have separated evolution in three stages similar to Maniadakis and Tani (2009),

[1] 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

investigating gradually more complex versions of the problem, starting with pure rule following, turning to one-step rule switching, and finally to multiple rule switching and betting. This is summarized in Table 1.

Additionally, we are interested in implementing CTRNNs that successfully deal with all the possible situations the agent may face in each evolutionary stage. This means we need to simultaneously examine all rules in the first stage, all switching combinations in the second stage, and all betting options in the third stage. To this end, we separate the overall experiment in six[2] different tasks each one examining a particular case of the problem. The evolutionary process will aim to design a single CTRNN accomplishing all tasks.

In the first generations, the evolutionary procedure aims at CTRNN controllers capable of adopting each one of the SS, OS and NR rules (two randomly initialized tasks per rule). Since we have six tasks to evaluate performance on three rules, in this stage we use two tasks per rule, which are randomly initialized. The joint accomplishment of all tasks implies the agent can follow the rules, giving successful responses for a long sequence of trials. 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 second evolutionary stage, the tasks are getting more complex,

[2] The number of tasks is specified by the six different switching combinations that may be defined among rules, that is (i) SS→OS, (ii) SS→NR, (iii) OS→SS, (iv) OS→NR, (v) NR→OS, (vi) NR→SS.

searching for controllers capable of switching between rules. Specifically, during generations 201-700, we 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 1, 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 follow 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 and the success of the betting strategy. 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-phases 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.

Measure of Successful Switching. To evaluate the accomplishment of a task, we consider two aspects of robot performance that regard (i) the success of rule following and (ii) the success of betting. In order to verify that rules are switched properly and the correct rule is followed at a given trial, we are based on target positions (see Fig 1). This approach is followed because it is necessary to have a continuous measure for the success of the given response. 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 cue, as it is described in Fig 1. 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:

$$SW_i = \sum_{q=1}^{p} \left(\sum_{t=11}^{T_q} \left(1 - \frac{d_{min}}{D} \right) \right) \tag{5}$$

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

Measure of Successful Betting. 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:

$$CB_t = \begin{cases} B_t \cdot (R_t - c \cdot P_t), & \text{if } B_t > 0.5 \\ -(1 - B_t) \cdot (R_t - P_t), & \text{if } B_t \leq 0.5 \end{cases} . \tag{6}$$

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 (6), 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 $c = 6.0$ encouraging 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:

$$BET_i = \sum_{q=1}^{p} \left(\sum_{t=1}^{T_q} CB_t \right) \tag{7}$$

Measure of Task Accomplishment. The 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:

$$E_{Task_i} = (SW_i) \cdot (BET_i)^d \tag{8}$$

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

Fitness Measure. 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 (8). Then, the "total fitness" of the individual is estimated by:

$$fit = \prod_{j=1}^{6} E_{Task_j} \tag{9}$$

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

2.4. Computational Details

The current study employs CTRNNs consisting of 6 neurons in the low level, and 8 neurons in the high level. This corresponds to a total number of 112 input synapses w_{ik} (96 directed to the lower part and 16 directed to the upper part). In the case of the bottleneck CTRNN scheme the interconnectivity of neurons is described by 128 synapses w_{im} (lower level: 36, upper level: 64, inter-level connectivity: 28). In the case of the fully connected CTRNN, we use 148 synapses w_{im} (lower level: 36, upper level: 64, inter-level connectivity: 48).

In order to evolve CTRNN configurations, we have used populations of 1000 individuals. Real-encoding is used to map synaptic weights $w_{ik}, w_{im} \in [-5, 5]$ and neural biases $\theta_i \in [-1, 1]$ to chromosomes. The time constant τ is not evolved, being set to 4 for all neurons. Based on the above, a complete network configuration is described in a chromosome of 254 parameters (i.e. 112+128+6+8) for the case of the bottleneck and 274 parameters (i.e. 112+148+6+8) for the case of the fully connected CTRNN.

Each candidate solution (i.e. CTRNN configuration) is separately tested on all six tasks described above, evaluating agent's rule-switching and betting capacity on all possible circumstances. At the beginning of each trial, the robot is located at a predefined starting position with its direction randomly specified in the range $[85^o, 95^o]$ degrees. The robot is kept in the same initial position for 5 simulation steps deciding the underlying response and the amount of betting. Then it is allowed to navigate freely in the environment for 165 more simulation steps (the total number of simulations steps in a trial is 170, which makes 170-5=165). Sensor noise has been set to 3%. After the completion of one trial the simulated robot is automatically transferred to the initial position having a new random direction, in order to start the next trial.

A standard genetic algorithm evolves populations, driven by the fitness measure described above. Due to the multiplicative nature of eq (9) being very sensitive in the failure of any of the six tasks, in the current implementation we have eliminated the use of crossover. This is because the drastic changes that crossover makes in the structure of CTRNNs impedes the progress of evolution in the early stages of the genetic algorithm. Therefore, the evolution have been based only on mutation that corresponds to the addition of a random number in $[-1.5, 1.5]$ for w_{ik}, w_{im} and a random number in $[-0.3, 0.3]$ for θ_i. During evolutionary steps, each parameter has probability 4% to be mutated.

3. Results

We have conducted experiments examining the possible neuronal mechanisms accounting for executive control function and particularly those involved in manipulating behavioural rules switching to the most appropriate ones as environmental conditions change, and additionally those involved in confidence development.

For each type of CTRNN architecture, we conducted 14 independent evolutionary runs. For the bottleneck CTRNNs, 6 runs converged successfully, producing robot controllers that can effectively switch rules and bet correctly. For the fully connected network architecture, none of the evolutionary runs produced a successful CTRNN model. Obviously, the bottleneck architecture outperformed the fully connected one in the WCSTB, implying that partial segregation of higher and lower level functionalities facilitates the performance of the overall system. In the following paragraphs we investigate further the results obtained with the bottleneck CTRNN.

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.

An interesting observation in Fig 3 is the consistency of trajectory patterns when the robot is heading to the same target under the same rule. For example, the trajectories generated at the 2nd, the 4th and the 30th trial, with the robot going to the right wing target under SS rule, they all follow a very similar pattern. However, if we look at the 18th, the 19th and the 25th trial, showing the trajectory of the robot when approaching the same target but now adopting rule OS, we observe a different motion pattern than before.

The same observation is also true if we examine the left directed turnings for these rules, as well as NR. This clear correlation between rules and robot trajectories, suggests that embodiment and sensory-motor interaction with the environment support the encoding of rules in the cognitive system.

3.1. CTRNN Internal Dynamics

In order to obtain insight into the cognitive mechanisms self-organized in the models, we have investigated the internal dynamics of the successfully evolved CTRNNs. We found that artificial evolution generated two broad categories of networks in which neural dynamics are qualitatively different. For the sake of clarity of our findings we will refer to these CTRNN categories as "Type-A" and "Type-B"[3]. In the totally 6 successful evolutionary runs, solutions of Type-A appeared 4 times, while solutions of Type-B appeared 2 times. In the following we discuss the particular characteristics of Type-A and Type-B solutions, examining one representative CTRNN from each category.

Focusing on a period of eight trials, Fig 4 shows the time developments of neural activation after principal component analysis (PCA), the betting rate and the received reward. In both cases, the experimenter changes the rule from OS to SS in the 3rd trial, with the agent adopting SS at the 5th trial. We see that for the Type-A solution (Fig 4(a)), the betting rate develops a peak at the onset of trials only when a reward is given in the end of the previous trial. Furthermore, neural activity showed distinct but similar

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

activation patterns for the cases that either OS (trials 1,2), or SS (trials 5-8) rule is followed. A similar observation is also true for the betting activity. In Type-B solution, neural activity keeps rather stationary values for both rules (Fig 4(b)). The rules SS and OS are now clearly differentiated, with the PCA of neural activation following completely different patterns for the two cases. The same is also true for the betting rate activity.

In order to investigate how different rules are encoded in the dynamics of the CTRNNs, we take the phase plots of higher level neural activities after PCA, for both Type-A and Type-B (Fig 5). In the two plots, three trajectories shown by different colours represent quasi-attractors encoding rules SS, OS and NR. In the case of Type-A (Fig 5(a)), we see three invariant sets of dynamically changing trajectories. Interestingly, 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). The overlap of SS and OS attractors suggests that these rules are organized as subclusters of a larger cluster separating them from NR. This is a reasonable organization since SS and OS exhibit similarities when they are both contrasted to NR. In particular, both SS and OS ask the agent to travel to the end of the corridor and turn left or right, while NR asks the agent to ignore cue stimulus and stay close to the starting position (see Fig 1). 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 facilitates the accomplishment of the investigated problem. On the contrary, the plot corresponding to Type-B solution (Fig 5(b)), shows three attractors akin to three different fixed points. This corresponds to clearly distinct representations for the rules SS, OS, NR. We would like to emphasize that this organization of rule encoding is also reasonable, since the three rules are actually independent from one another. The completely separate representation of SS and OS highlights their distance, while at the same time in Fig 5(b), they both remain far from NR. In that way, Type-B solution highlights the unique identity of each rule. In summary, the rule representations self-organized in Type-A and Type-B solutions reflects the different interpretations one can give to the rule-switching problem investigated in the present work.

Additionally, our observations imply that continuous dynamic systems may accomplish two types of rule encoding. The one is represented by partially overlapping attractors with invariant sets of dynamically changing trajectories and the other by distinct attractors akin to fixed points. The former case might be analogous to the monkey experiment results by Mansouri et al. (2006). On the other hand, the latter case might be similar to the discrete-like encoding suggested in O'Reilly (2006) even though our model does not employ complex hand-coded mechanisms like gating of local stripes.

3.2. Betting Strategy

Moreover, we examine the success of betting strategy based on the currently adopted rule in each trial (we remind that the agent decides the amount of betting during the very first steps of each trial). A representative sample of 80 consecutive trials corresponding to Type-A and the Type-B results is explored in Fig 7, demonstrating the time history of the average neural activity principal components at the onset of trials, the betting rate, and the reward at the end of the underlying trials. The time of rule changes (specified by the experimenter) is indicated by colour changes in the plots. For both Type-A and Type-B, we observe that the neural activation pattern differentiates depending on the currently adopted rule. Furthermore we observe that neural activity is different when the correct rule is unknown (rule transition period in the first trials of each phase), or successfully identified (rule following period of a phase). In the same way, the amount of betting decreases during the rule exploration period, but rises up when rules are successfully followed. Interestingly, the amount of betting in successful trials differs depending on the rules adopted, keeping nearly constant values. In the case of Type-A, the betting rate for NR is the highest, for OS is the lowest, while for SS is in the middle. For Type-B, we see a similar ordering in the betting rates of the three rules (NR:high, SS:middle, OS:low), with the average betting rate however, being higher than Type-A.

We have studied further the obtained CTRNNs in order to explain how rules are linked with the betting amounts. Statistical analysis of agent's behaviour under "no-rule-change" conditions revealed that the amount of betting is related with rule stability. In the no-rule-change experiment the agent has to follow a single rule for a long sequence of trials, without switching. Making statistics of the obtained CTRNN solutions on this test, we observed that there is a small chance for the agent to unreasonably switch the adopted rule, even if it is normally rewarded. For the case of Type-A solution, Table 2(a) shows the probability of unreasonable switching for the three rules, together with the corresponding average rates of betting for each case. We observe that rules OS and NR having the lowest and highest betting rates respectively, inversely show the highest and lowest unreasonable-switching probabilities. The same relationship is also valid for the case of

Type-B solution, as it is shown in Table 2(b). Note that in the latter case, the probabilities of unreasonable switching are generally lower compared to the Type-A solution, which explains the higher betting rates for Type-B.

This finding implies that the agent has developed a two-fold context-dependent monitoring system that affects betting. First, the agent employs some kind of objective criteria that are based on the probability of success or failure of forthcoming trials (i.e. when reward or punishment is more likely) in order to increase or lower the amount of betting. Second, the agent uses subjective criteria, considering its own capacity to stably follow a given rule. On this basis, an internal confidence mechanism makes the agent bet higher for more stable rules.

3.3. Relevance to Biological Data

The investigation of the implemented models revealed the emergence of brain-like characteristics in the CTRNNs. More specifically, the bottleneck connectivity used in the model enforces the modular organization of the overall system separating the processing of high and low level cognitive issues, in a way that resembles the organization of the primate brain. The upper part of the network is mainly involved in rule manipulation and betting, which clearly indicates PFC-like functionality, Ko et al. (2008); Fecteau et al. (2007); Bechara et al. (1997). Moreover, the lower part of the network considers mainly environment interaction issues, similar to the role of the posterior sensory-motor areas in the primates' brain. Note that the mechanisms of low-level control are shaped in the early generations of the tuning procedure, while the high-level skills need much more time to shape and stabilize. This is in agreement with the primates developmental procedure that starts with acquiring the capacity to interact efficiently with the environment, and then proceeds with high-level cognition and executive control functioning.

Rule preservation in WCST is a known deficit for adult pathologic subjects (e.g schizophrenia), Everett et al. (2001). The Type-A and Type-B models discussed in the previous sections do not show any type of preservation characteristics. However, the different rule switching mechanisms they have implemented differentiate the average number of trials needed to successfully identify the new correct rule (4.7 for Type-A and 1.8 for Type-B solutions). In an attempt to simulate pathologic brain functioning, we have investigated the effect of artificially reduced neural activity in the higher part of the CTRNNs. In all successful models, we have found pairs of high level neurons that after a 30% - 50% reduction on their activity significantly eliminate the rule switching capacity of the models. Note that the artificially imposed disorder reduced flexibility on rule switching does not actually affect their navigation skills which remain intact.

Regarding the cognitive dynamics involved in betting, our model follows the basic rule of probabilistic cognition that explains the majority of human risky decisions. According to the probabilistic explanation of decision making, subjects decisions aim to maximize the utility expected from their behaviours, Luce and Raiffa (1989); Boyer (2006). This is also a major assumption of reinforcement learning methods, Sutton and Barto (1998). Similarly in our model the agent bets higher for the more stably represented rules that have larger probability to lead to a positive feedback, while bets lower for less stable rules. A number of modern studies demonstrated that human behaviour is not entirely consistent with the above mentioned probabilistic approach, especially in non-laboratory, real life conditions, Boyer (2006). More specifically, humans in real-life are often risk-aversive or risk-seeking with the latter showing a clear peak during adolescence. Besides genetic parameters, the operating environment is one important factor that shapes our mood for risk, Dodge and Pettit (2003). A similar observation has also appeared in our experiments. More specifically, the coefficient c that weights negative reward in eq (6), determines how important is punishment signal for the survival of an agent in the next evolutionary generation. We observed that c significantly biases the mood for risk developed by the agents. Low c values lead to the development of gradually more risk-seeking strategies, while high c values lead to gradually more risk-aversive strategies. This dual performance for our models is similar to Frank and Claus (2006), exploring the role of orbitofrontal cortex in risky decisions.

4. Discussion

In the current work we investigate rule-switching and confidence development mechanisms in robotic agents. We examine the neurodynamics self-organized in CTRNN models in order to reveal possible working principles for the human brain. Our study follows a minimum constraint approach that avoids assigning predefined roles at different parts of the artificial cognitive system. The fact that statistically independent evolutionary runs produced results with similar qualitative characteristics, suggests that cortical mechanisms analogous to the ones found in our models may also govern brain functionality. More specifically, our study suggests the following principles that may apply to the real brain:

- High level cognitive processes are possible to work on the basis of dynamical systems operating in a continuous time and space domain. This contrast to the traditional approaches explaining high level brain workings adopting a discrete symbol manipulation approach, O'Reilly (2006); Dayan (2007). Such a dynamical systems framework is consistently supported by our previous robotic studies, Paine and Tani (2005); Yamashita and Tani (2008); Maniadakis and Tani (2009), as well as other neuromodelling works (e.g. Schapiro and McClelland (2009)) arguing that many rule-based processes accounting for human behaviours could be better represented by means of a continuous perspective. Interestingly, the dynamical systems approach may develop both stationary-like encoding, as well as dynamic encoding in the form of invariant sets. The latter case matches very well with electrophysiological brain imaging data revealing a fluctuating (rather than stationary) cortical activity in the brain of primates as it is shown in Mansouri et al. (2006).

- The embodiment of the cognitive system and the interaction of the agent with the environment, affect the self-organization of high level cognitive mechanisms in the case of dynamic representations. The necessity for an actual behavioural response (i.e. a sequence of motor commands) is an important feature of our approach which clearly separates our study from previous works aiming at instant conceptual responses (for example the switching task presented in Rougier and O'Reilly (2002) concentrates on 'naming' the rule rather than giving any particular behavioural response, a theoretical problem interpretation that is also followed in Dehaene and Changeux (1991); Stemme et al. (2007a)). However, as it is shown from our results (i.e. Type-B solutions) high-level functions may adopt stationary-like representations which seem to significantly improve the stability of the system against sensory noise (see Table 2).

- Our findings showed that a loose segregation of the cognitive model using by design bottleneck architectures, enhances the functionality of the global system. Non-structured and fully connected neural systems have significantly lower chances to accomplish complex functionalities. The partial segregation of system components facilitates the emergence of different roles in each subsystem, and the self-organization of functional hierarchies. A possible correspondence between our CTRNN models and the real brain assigns the higher part of the CTRNN to Prefrontal cortex and the lower part to posterior sensory-motor cortices. This is because the higher CTRNN part considers rule decision, and response planning, while the lower part takes care of motion details and the actual interaction with the environment.

- The self-organization of different types of solutions in our robotic WC-STB experiments suggests that some high level cognitive functionalities may are not universally organized in primates' brains. In particular, the obtained results showed that Type-A solutions focus on the similarity of SS and OS rules (compared to NR), while Type-B solutions follow uncorrelated and independent representations for SS and OS adopting an equi-distance representation for all the three rules. This finding suggests that the different interpretations that may be given to a problem are directly affecting the mechanisms that will be used to face the problem. This observation addresses a rather obvious but rarely considered parameter of cognition, that if two subjects understand a given problem in different ways, then it is likely to use brain resources differentially when solving the problem. This suggestion is mainly relevant for high level skills because they are not directly linked with the phylogenetically strict characteristics of the low-level sensory-motor system.

- Finally, our results showed that artificial confidence mechanisms involved in betting, rely on objective criteria (i.e. what is the probability of receiving a reward in the next trial) as well as subjective criteria (i.e. how efficient the agent is in following a given rule). This mechanism that resembles the role of the critic module in reinforcement learning schemes, Sutton and Barto (1998), suggests a stronger self-monitoring functionality for the critic.

4.1. Generalisation

The strong coupling between cognitive dynamics and embodiment that has been observed in our results (see paths in fig 3) raises the issue of whether dynamic models can perform successfully when environment properties change. This is because, in the context of embodied cognition, environment seems to significantly facilitate keeping high-level information in working memory. To investigate further this issue we have recently explored the capacity of robotic agents to perform rule switching in multiple T-maze environments having different structural characteristics, Maniadakis et al. (2011a). The task considered in this study was very similar to the one described in the present work, with the additional difficulty for the robot to successfully apply the delayed response rules in three different environments. This means that there were not constant environment properties that could support encoding rules in working memory, i.e. the moment of turning to the left or right side greatly varies between the three environments. After the evolution of CTRNN controllers, the analysis of neural activity showed that in the higher level of the network, three different shapes of attractors are implemented each one encoding one of the T-maze types (see fig 8, comparing SS and OS activity for Maze1, Maze2, Maze3). This observation indicates that dynamical models abstract high-level representations of environment types in order to improve robustness against sensory variations. Based on this abstraction, the agent successfully adjusts how rules are applied in each maze. Note that the agent can also operate successfully in new (previously unexperienced) environments that combine the characteristics of the known T-mazes (however, it fails in new over-sized mazes e.g. when the length of the corridor significantly exceeds the length that the agent has already experienced). Overall, by monitoring and abstracting environment specific information, agents can sufficiently generalize the representation of rules adapting smoothly their behavior in multiple environments. Note additionally, that the three different mazes could have been represented in the higher level by fixed point's. However, the model prefers attractor encoding. Intuitively, this is because dynamic approaches (i) enable operating successfully in mazes with varied characteristics and (ii) seems to facilitate switching from one maze type to the other (i.e. the partial overlap of attractors representations alleviates moving, in mental and behavioral terms, from one maze to the other).

In order to investigate further the generalization capacity of dynamic approaches we have also explored the role of temporal constraints in shaping neurodynamics, Maniadakis et al. (2011b). Keeping the task of rule switching in T-maze environments, we have investigated CTRNNs that can successfully deal with trials having either static (predefined) or dynamic (free agent's choice) temporal durations. The obtained results revealed that different principal components of neural activity undertake the responsibility of handling the two different versions of the experimental setup. In other words, two subsystems are implemented which consider the two modes of operation of the cognitive system.

Summarizing, according to the aforementioned supplementary studies, Maniadakis et al. (2011a,b), dynamic cognition approaches have at least two different mechanisms to support generalization, which account for (i) the abstraction of high-level representations and (ii) the internal formulation of partially distinct subsystems. Both can deal with sensory varying versions of a given task, improving the robustness of the cognitive process. Of course there are important generalization aspects which have not been addressed in our experiments, such as the ability to incorporate new behavioral rules, to apply known rules in drastically different situations, to combine knowledge obtained from different tasks and others. These issues need systematic investigation to conclude whether dynamic approaches can sufficiently explain high level cognitive processes and in which terms they might be more or less efficient than static representations.

5. Conclusions

In the present work we employ CTRNN models to explore cognitive processes involved in rule switching and confidence, in the domain of artificial agents. Our computational experiments suggest new ways for approaching high-level cognitive functions in the real brain.

In the future we will investigate whether the results of the current work can be used on humanoid robot control that provides a rather realistic framework for exploring complex cognitive phenomena, allowing also the direct comparison of artificial cognitive dynamics with primate's brain.

Baev, K., 2007. A new conceptual understanding of brain function: basic mechanisms of brain-initiated normal and pathological behaviors. Critical Reviews in Neurobiology 19 (2-3), 119–202.

Bechara, A., Damasio, H., Tranel, D., Damasio, A., 1997. Deciding advantageously before knowing the advantageous strategy. Science 275, 1293–1295.

Beer, R., 1995. A dynamical systems perspective on agent-environment interaction. Artifficial Inteligence 72, 173–215.

Berg, E., 1948. A simple objective technique for measuring flexibility in thinking. Journal of General Psychology 39, 15–22.

Borrett, D., Jin, F., Kwan, H., 2005. Evolutionary autonomous agents and the nature of apraxia. Biomedical Engineering Online 4 (1), doi:10.1186/1475–925X–4–1.

Botvinick, M., Plaut, D., 2006. Short-term memory for serial order: a recurrent neural network model. Psychological Review 113 (2), 201–233.

Boyer, T., 2006. The development of risk-taking: a multi-perspective review. Developmental Review 26, 291–345.

Brody, C., Romo, R., Kepecs, A., 2003. Basic mechanisms for graded persistent activity: discrete attractors, continuous attractors, and dynamic representations. Current Opinion in Neurobiology 13, 204–211.

Dayan, P., 2007. Bilinearity, rules, and prefrontal cortex. Frontiers in Computational Neuroscience 1, 1–14.

Dehaene, S., Changeux, J., 1991. The wisconsin card sorting test: theoretical analysis and modeling in a neuronal network. Cerebral Cortex 1 (1), 62–79.

Dodge, K., Pettit, G., 2003. A biopsychosocial model of the development of chronic conduct problems in adolescence. Developmental Psychology 39, 349–371.

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

Everett, J., Lavoie, K., Gagnon, J., Gosselin, N., 2001. Performance of patients with schizophrenia on the wisconsin card sorting test (wcst). J Psychiatry Neuroscience 26 (2), 123–130.

Fecteau, S., Knoch, D., Fregni, F., Sultani, N., Boggio, P., Pascual-Leone, A., 2007. Diminishing risk-taking behavior by modulating activity in the prefrontal cortex: A direct current stimulation study. The Journal of Neuroscience 27 (46), 12500–12505.

Frank, M., Claus, E., 2006. Anatomy of a decision: Striato-orbitofrontal interactions in reinforcement learning, decision making and reversal. Psychological Review 113 (2), 300–326.

Joel, D., Weiner, I., Feldon, J., 1997. 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 85, 187–201.

Johnson, J., Spencer, J., Luck, S., Schner, G., 2009. A dynamic neural field model of visual working memory and change detection. Psychological Science 20 (5), 568–577.

Jun, J., Miller, P., Hernandez, A., Zainos, A., Lemus, L., Brody, C., Romo, R., 2010. Heterogenous population coding of a short-term memory and decision task. The Joutnal of Neuroscience 30 (3), 916–929.

Kaplan, G., Sengor, N., Gurvit, H., Genc, I., Guzelis, C., 2006. A composite neural network model for perseveration and distractibility in the wisconsin card sorting test. Neural Networks 19, 375–387.

Ko, J., Monchi, O., Ptito, A., Petrides, M., Strafella, A., 2008. Repetitive transcranial magnetic stimulation of dorsolateral prefrontal cortex affects performance of the wisconsin card sorting task during provision of feedback. Int J Biomed Imaging 143238.

Koren, D., Poyurovsky, M., Seidman, L., Goldsmith, M., Wenger, S., Klein, E., 2005. The neuropsychological basis of competence to consent in first-episode schizophrenia: a pilot metacognitive study. Biol Psychiatry 57 (6), 609–616.

Koren, D., Seidman, L., Harvey, P., 2006. Real-world cognitive–and metacognitive–dysfunction in schizophrenia: a new approach for measuring (and remediating) more right stuff. Schizophrenia Bulletin 32 (2), 310–326.

Lipson, H., 2005. Evolutionary design and evolutionary robotics. Biomimetics, 129–155.

Luce, R., Raiffa, H., 1989. Games and Decisions: Introduction and Critical Survey. Dover Publications.

Machens, C., Romo, R., Brody, C., 2005. Flexible control of mutual inhibition: A neural model of two-interval discrimination. Science 307, 1121–1124.

Maniadakis, M., Tani, J., 2009. Acquiring rules for rules: Neuro-dynamical systems account for meta-cognition. Adaptive Behavior 17, 58–80.

Maniadakis, M., Tani, J., Trahanias, P., 2011a. Ego-centric and allo-centric abstraction in self-organized hierarchical neural networks. In: Proc. combined IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EPIROB).

Maniadakis, M., Wittmann, M., Trahanias, P., 2011b. Time Experiencing by Robotic Agents. In: Proc. European Symposium on Artificial Neural Networks (ESANN).

Mansouri, F., Matsumoto, K., Tanaka, K., 2006. 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 26, 2745–2756.

Milner, B., 1963. Effects of different brain lesion on card sorting. Archives of Neurology 9, 90–100.

Narayanan, S., 2003. The role of cortico-basal-thalamic loops in cognition: a computational model and preliminary results. Neurocomputing 52-54, 605–614.

Nolfi, S., Floreano, D., 2000. Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. MA: MIT Press/Bradford Books.

O'Reilly, R., 2006. Biologically based computational models of high-level cognition. Science 314, 91–94.

Paine, R., Tani, J., 2005. How hierarchical control self-organizes in artificial adaptive systems. Adaptive Behavior 13 (3), 211–225.

Palva, M., Monto, S., Kulashekhar, S., Palva, S., 2010. Neuronal synchrony reveals working memory networks and predicts individual memory capacity. Proc Natl Acad Sci U S A. 107 (16), 7580–7585.

Romo, R., Salinas, E., 2003. Flutter discrimination: neural codes, perception, memory and decision making. Nature Reviews Neuroscience 4, 203–218.

Rougier, N., O'Reilly, R., 2002. Learning representations in a gated prefrontal cortex model of dynamic task switching. Cognitive Science 26, 503–520.

Ruppin, E., 2002. Evolutionary autonomous agents: A neuroscience perspective. Nature Reviews Neuroscience 3 (2), 132–141.

Schapiro, A., McClelland, J., 2009. A connectionist model of a continuous developmental transition in the balance scale task. Cognition 110, 395–411.

Singh, R., Eliasmith, C., 2006. Higher-dimensional neurons explain the tuning and dynamics of working memory cells. Journal of Neuroscience 26 (14), 3667–3678.

Stemme, A., Deco, G., Busch, A., 2007a. The neuronal dynamics underlying cognitive flexibility in set shifting tasks. Journal of Computational Neuroscience 23, 313–331.

Stemme, A., Deco, G., Busch, A., 2007b. The neuronal dynamics underlying cognitive flexibility in set shifting tasks. J Comput Neurosci 23, 313–331.

Sutton, R., Barto, A., 1998. Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA.

Van Gelder, T., 1998. The dynamical hypothesis is cognitive science. Behavioral and Brain Sciences 21, 615–628.

Yamashita, Y., Tani, J., 2008. Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. PloS PLoS Computational Biology 4, e1000220.

Yamauchi, B. M., Beer, R. D., 1996. Spatial learning for navigation in dynamic environment. IEEE Trans. Syst. Man Cybern. 26 (3).

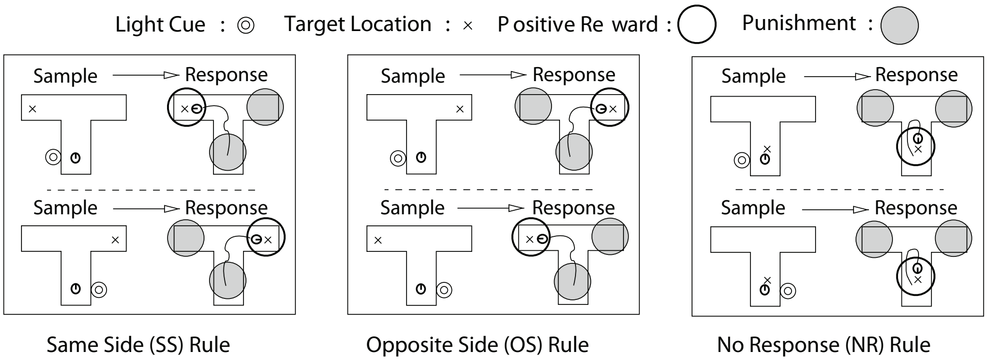


Figure 1: A graphical interpretation of the three sample-response rules used in our experiments. Each box explains one sample-response rule. In each box, the first line shows correct robot response when light appears to the left side of the robot, while the second line shows correct response when light appears to the right.

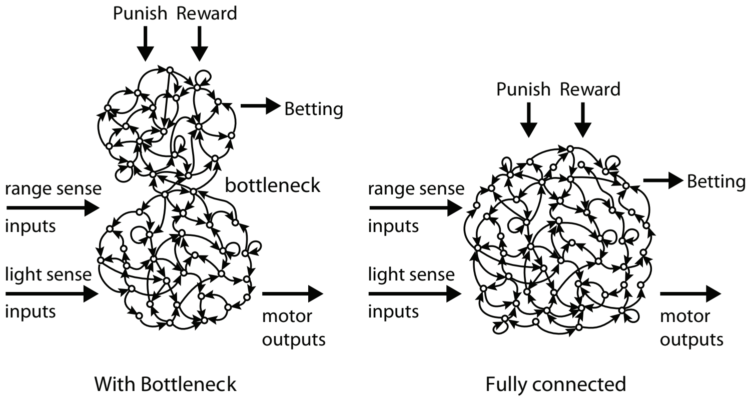


Figure 2: Schematic representation of the bottleneck and fully connected CTRNNs explored in the present study.

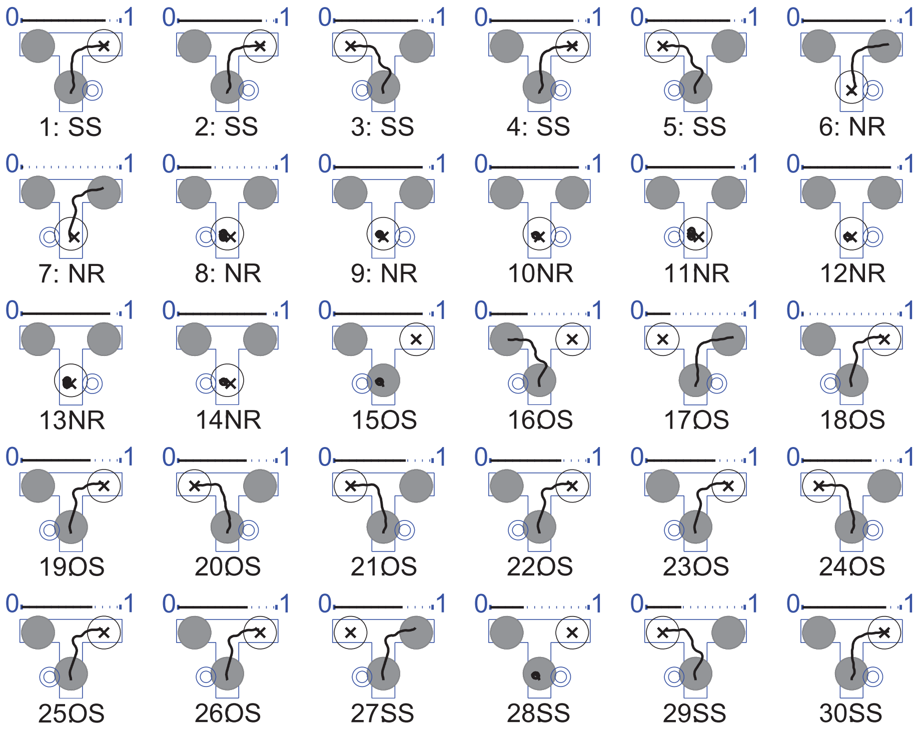
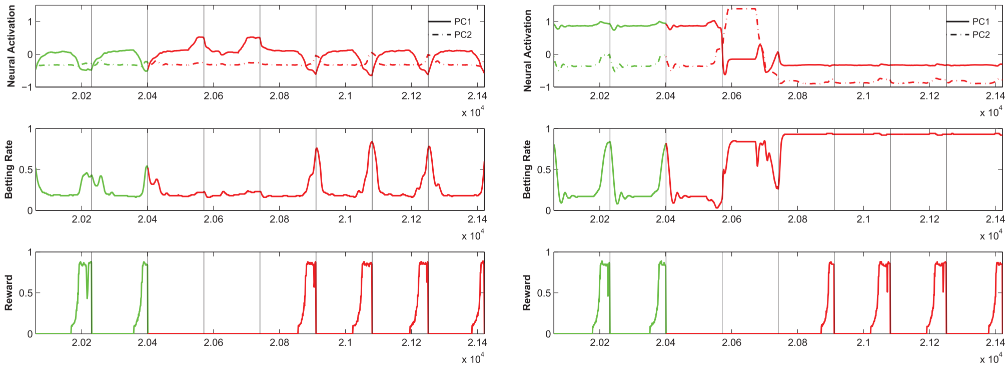


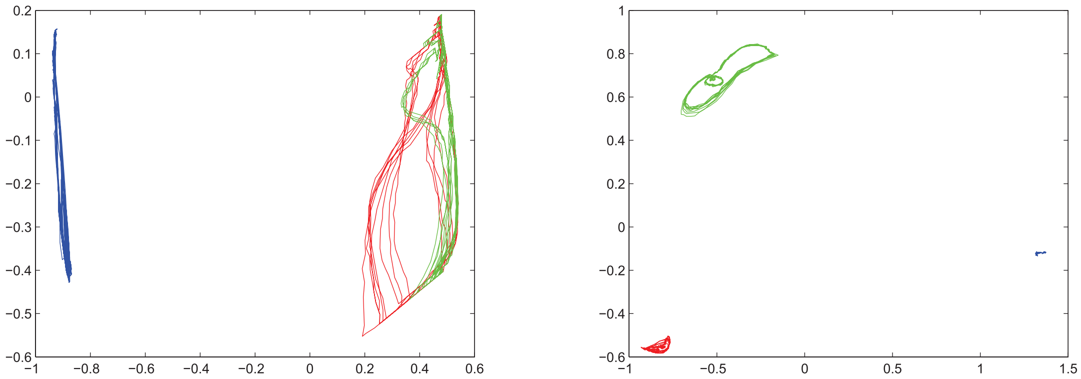
Figure 3: The behaviour of the agent in a sequence of trials. Light is depicted with a double circle, goal position is depicted with an ×, punishment area is depicted with a gray circle, while robot path is depicted with a black line starting from the bottom of the T-maze. In the present figure we follow a more compact representation of a sample-response trial than the one shown in Fig 1, in order to demonstrate an adequately large number of robot trials.



(a) Type-A

(b) Type-B

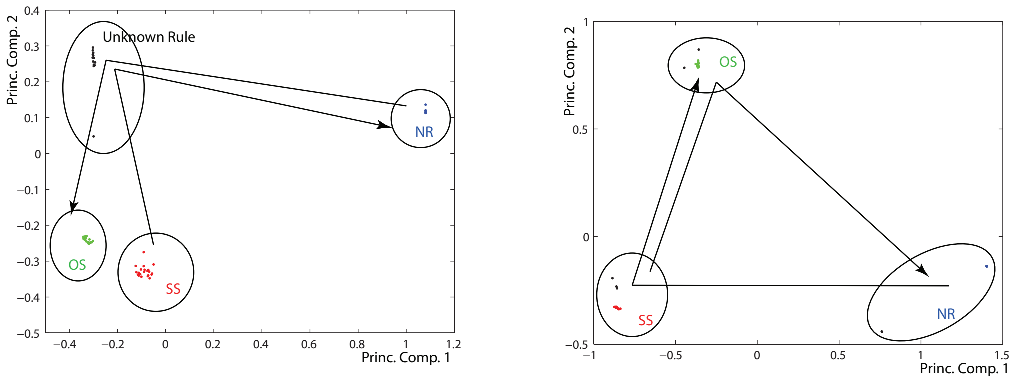
Figure 4: A comparison of the internal dynamics of Type-A and Type-B CTRNNs, for a sequence of eight trials (separated by vertical lines) during rule transition from OS to OS. The trials corresponding to OS are shown in green, while trials corresponding to SS are shown in red. In the top subplots we show the first two principal components of neural activity. The subplot in the middle depict the activity of the betting neuron. Finally, the subplots in the bottom show the reward received by the agent in each trial. We can easily see that missing rewards (i.e. the response of the agent is not correct) destabilize the dynamics of the CTRNNs facilitating the transition from one rule to the other.



(a) Type-A

(b) Type-B

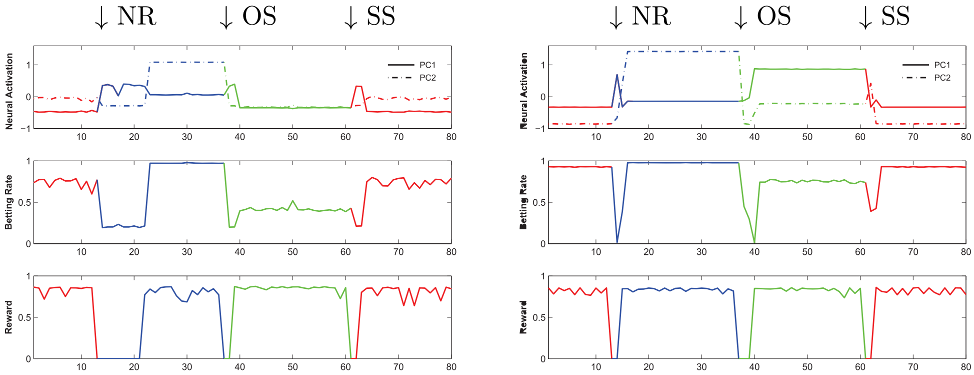
Figure 5: Phase plots of the principal components of neural activity for the Type-A and Type-B CTRNNs. The x-axis corresponds to the first principal component, and y-axis corresponds to the second principal component. The phase plot of each rule is shown in a different colour, with red corresponding to SS, green corresponding to OS, and blue corresponding to NR. We observe that in the case of Type-A solution there is a partial overlap in the representation of rules SS and OS, while in the case of Type-B solution all rules have a clearly distinct representation.



(a) Type-A

(b) Type-B

Figure 6: The averages of the first two principal components in the beginning 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.



(a) Type-A

(b) Type-B

Figure 7: A graphical illustration of the relationship between neural activity, betting rate and received reward, for the Type-A and Type-B CTRNNs. In both cases, a sequence of 80 consecutive trials is depicted. The transitions between rules are illustrated by colour changes. The trials corresponding to SS are shown in red, the trials corresponding to NR are shown in blue, and those corresponding to OS are shown in green. For both Type-A and Type-B, the subplot in the top shows the average of neural activity first two principal components at the onset of trials, the subplot in the middle shows the amount of agent betting in each trial, and the subplot in the bottom shows the amount of reward received by the agent in each trial (zero reward indicates non-successful responses).

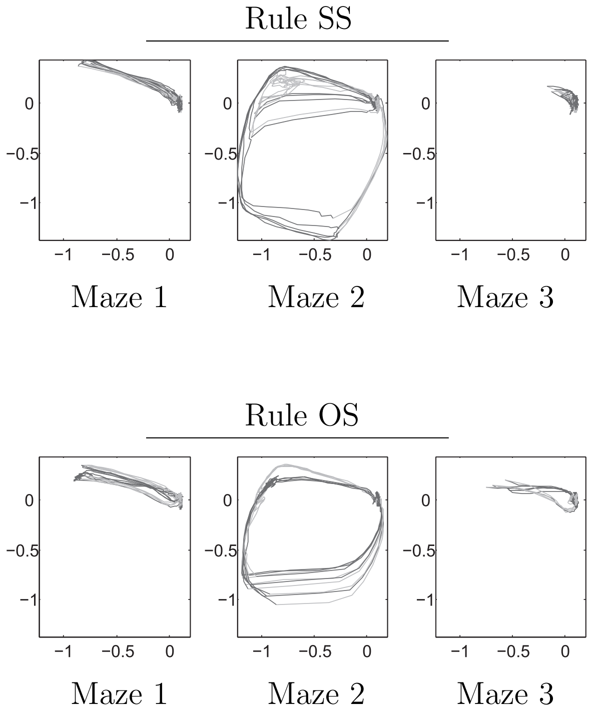


Figure 8: Phase plots of neural activity in the upper level of the CTRNN, when the agent follows the SS and OS rule in three different T-maze environments. For each plot, the x-axis and y-axis correspond to the first and second principal component respectively. For more details, see Maniadakis et al. (2011a).

Evolutionary Procedure for Rule Switching

Generations	Task Type	Description
1st Stage	*Task 1*: CTRNN reset- SS	
	Task 2: CTRNN reset- SS	
	Task 3: CTRNN reset- OS	Single Phase
1-200	*Task 4*: CTRNN reset- OS	
	Task 5: CTRNN reset- NR	
	Task 6: CTRNN reset- NR	
2nd Stage	*Task 1*: CTRNN reset- SS → OS	
	Task 2: CTRNN reset- SS → NR	
	Task 3: CTRNN reset- OS → SS	Two Phases
201-700	*Task 4*: CTRNN reset- OS → NR	
	Task 5: CTRNN reset- NR → OS	
	Task 6: CTRNN reset- NR → SS	
3rd Stage	*Task 1*: CTRNN reset- SS → OS → NR → SS . . . OS	
	Task 2: CTRNN reset- SS → NR → SS → OS . . . NR	
	Task 3: CTRNN reset- OS → SS → OS → NR . . . SS	Multiple Phases
701-1200	*Task 4*: CTRNN reset- OS → NR → SS → OS . . . SS	
	Task 5: CTRNN reset- NR → OS → NR → SS . . . NR	
	Task 6: CTRNN reset- NR → SS → OS → NR . . . OS	

Table 1: The incrementally more complex tasks used in the evolutionary procedure investigating switching between three rules.

Rule	**Failure Probability**	**Average Betting**
SS	1.78%	0.73
OS	2.99%	0.42
NR	0.0%	0.97

(a) Type-A

Rule	**Failure Probability**	**Average Betting**
SS	0.01%	0.93
OS	0.08%	0.77
NR	0.0%	0.98

(b)Type-B

Table 2: The relationship between failure probability and betting, for the Type-A and Type-B CTRNNs. In both cases, we see that the agent bets higher for more stable rules.