

Adaptive Behavior

<http://adb.sagepub.com>

Acquiring Rules for Rules: Neuro-Dynamical Systems Account for Meta-Cognition

Michail Maniadakis and Jun Tani
Adaptive Behavior 2009; 17; 58
DOI: 10.1177/1059712308101739

The online version of this article can be found at:
<http://adb.sagepub.com/cgi/content/abstract/17/1/58>

Published by:



<http://www.sagepublications.com>

On behalf of:



[International Society of Adaptive Behavior](#)

Additional services and information for *Adaptive Behavior* can be found at:

Email Alerts: <http://adb.sagepub.com/cgi/alerts>

Subscriptions: <http://adb.sagepub.com/subscriptions>

Reprints: <http://www.sagepub.com/journalsReprints.nav>

Permissions: <http://www.sagepub.co.uk/journalsPermissions.nav>

Citations <http://adb.sagepub.com/cgi/content/refs/17/1/58>

Acquiring Rules for Rules: Neuro-Dynamical Systems Account for Meta-Cognition

Michail Maniadakis, Jun Tani

Laboratory for Behavior and Dynamic Cognition, Brain Science Institute, RIKEN

Both animals and humans use meta-rules in their daily life, in order to adapt their behavioral strategies to changing environmental situations. Typically, the term meta-rule encompasses those rules that are applied to rules themselves. In cognitive science, conventional approaches for designing meta-rules follow human hardwired architectures. In contrast to previous approaches, this study employs evolutionary processes to explore neuronal mechanisms accounting for meta-level rule switching. In particular, we performed a series of experiments with a simulated robot that has to learn to switch between different behavioral rules in order to accomplish given tasks. Continuous time recurrent neural networks (CTRNN) controllers with either a fully connected or a bottleneck architecture were examined. The results showed that different rules are represented by separate self-organized attractors, while rule switching is enabled by the transitions among attractors. Furthermore, the results showed that neural network division into a lower sensorimotor level and a higher cognitive level enhances the performance of the robot in the given tasks. Additionally, meta-cognitive rule processing is significantly supported by the embodiment of the controller and the lower level sensorimotor properties of environmental interaction.

Keywords meta-level cognition · meta-cognitive dynamics · evolutionary self-organization · cognitive robotics · dynamical systems · recurrent neural networks

1 Introduction

In order to optimize their gains from environmental interaction, animals and humans are equipped with the ability to manipulate their own behavioral strategy, adjusting it properly to the given situation. This kind of behavioral plasticity requires the development of mental processes that apply active control over other cognitive procedures. The higher level processes involved in directing one's cognitive operation are generally referred to as "meta-cognitive." The continuous evaluation and control of own thoughts, memories, action,

and so forth, facilitates adaptivity to complex circumstances. Therefore, meta-cognitive processes support the effectiveness of individuals in everyday life (Swanson, 1990), and they have been proposed as a major mechanism for regulating strategy selection (Dunlosky & Hertzog, 1998).

The term meta-cognition established by the psychologist J. Flavell during 1970s (Flavell, 1971, 1976), encompasses the capacity of a subject to influence his own mental processes whose content determines other internal cognitive states of the subject. Meta-cognition is usually referred to as an internally directed proce-

Correspondence to: Michail Maniadakis, Laboratory for Behavior and Dynamic Cognition, Brain Science Institute (BSI), RIKEN, Japan.
E-mail: mmaniada@brain.riken.jp

Copyright © 2009 International Society for Adaptive Behavior (2009), Vol 17(1): 58–80.

DOI: 10.1177/1059712308101739

Figures 3, 9, 10 appear in color online: <http://adb.sagepub.com>

dure.¹ Generally speaking, “meta-X” is translated to “X about X.” This means that meta-memory concerns memory about memory (e.g., I remember that I used to memorize the lyrics of this song), meta-learning concerns learning about learning (e.g., I learned that mathematics are better studied by solving many exercises), meta-knowledge concerns knowledge about knowledge (e.g., I raise my hand, because I am sure that I know the answer to teacher’s question), and so on.

Many research efforts addressing meta-cognitive phenomena investigate the borders between cognitive and meta-cognitive processes (Dunning, Johnson, Ehrlinger, & Kruger, 2003). These endeavors are supported by modeling studies on meta-cognition. In a typical model, distinct computational components are utilized to accomplish reasoning and meta-reasoning tasks (Conitzer, 2008). For example in Russell (1991) a probabilistic process is used to accomplish self-monitoring; Gordon and Hobbs (2003) present a set of axioms interpreted in first-order logic to manipulate memory-related processes; in Dastani, Governatori, Rotolo, Song, and Torre (2007) rules with different priorities are used to describe the interaction of an agent’s possible actions; and in Gordon, Hobbs, and Cox (2008) an expectation-based meta-reasoning process is employed for controlling cognitive processes. Other application-oriented meta-models have also appeared in the literature, covering a large range of topics from meta-memory (Ramesh, 1997), to meta-learning (Giraud-Carrier, Vilalta, & Brazdil, 2004; Klinkenberg, 2005), meta-rules (Bogacz & Giraud-Carrier, 1998; Cazenave, 2003; Weischedel & Sondheimer, 1983) and meta-knowledge (Bessiere, Freuder, & Regin, 1999; Garner, 2000). Previous computational approaches to meta-processing are reviewed by Cox (2005) and Anderson and Oates (2007).

This study concentrates on meta-rule processing, investigating the development of higher level cognitive rules used to apply control over behavioral rules (Clancey & Bock, 1985). Meta-rules constitute a basic component of meta-cognition. They accompany a great deal of our daily activities, for example when a traffic jam makes us change the route for driving home, when we try different strategies to attract the attention of a mate, when we change the style of our speaking depending on the mood of our boss, and many others. One of the oldest and most famous examples regarding meta-rule manipulation in animals was

introduced by G. Bateson (1972). He described a dolphin that learned behavior scheme rules and simultaneously higher level rules which switch behavior rules from one to another, in order to maximize rewards. Similarly, in this study we investigate the acquisition of meta-level cognitive rules that switch behavioral rules.

In the field of brain science, rule-switching mechanisms have been investigated by using the well known Wisconsin Card Sorting (WCS) task (Greve, Stickley, Love, Bianchini, & Stanford, 2005; Mansouri & Tanaka, 2003; Milner, 1963). In this test, the experimental subject is presented with cards that display symbols in specific shapes, colors, and numbers, such as three green circles, four yellow triangles, and so forth. The task of the subject is to sort the cards into different piles without knowing the criteria for a correct sorting. The subject is given feedback about the correctness of his/her sort, after each card has been placed. Once he/she has discovered the sorting-rule (e.g., that the cards should be sorted by color), the sorting dimension is changed by the experimenter, and the subject then has to discover the new rule (e.g., that the cards are to be ordered by shape). Through decades of research the WCS test has been established as a standard cognitive test that targets subjects’ abilities to switch between cognitive rules. A large number of studies with humans and monkeys have shown that subjects with prefrontal lobe impairments lose their efficiency in WCS switching. Therefore, it is widely considered that the prefrontal lobe is the core site for processing meta-rules (Fernandez-Duque, Baird, & Posner, 2000).

The majority of neural network models used to explain meta-rule processing consist of specialized components responsible for memory retrieval, matching, working memory, long-term memory and rule alternation, trying to explain prefrontal lobe functionality (Kaplan, Sengr, Grvit, Genc, & Gzelis, 2006; Stemme, Deco, & Busch, 2007). In a typical explanation of the rule switching mechanism, a rule is retrieved from long-term memory to working memory in order to be applied to the current situation. If the rule matches the situation (i.e., the subject avoids punishment) it is kept in the working memory. Otherwise, the rule is switched. Unfortunately, these typical cognitive explanations sound like algorithmic processes of modern computers rather than cognitive processes of the real brains. Despite the success of the above mentioned models in given tasks, their algorithmic architecture can hardly

provide any insights in exploring the actual brain mechanisms for meta-cognition. This is because in the conventional brain modeling approaches, modelers think that if task explanation seems explicit and logical, then the corresponding algorithms implemented in the models would resemble the natural biological mechanisms of the brain. However, this is not always true with the internal functionality of the brain. Varela (1979) has described this problem as the gap between the external observer side and the internal operational side. Therefore, conventional brain modeling approaches can sometimes be regarded as similar to application-specific meta-processes using arbitrary interactions between behavior level and meta-level in order to enhance system performance (Bogacz & Giraud-Carrier, 1998; Cazenave, 2003; Weischedel & Sondheimer, 1983), and they can hardly provide any new implication about naturally emergent processes in biological systems.

Because of their potential arbitrariness, the above-mentioned hardwired architectures should not be considered as the only approach for implementing meta-rule cognition. For example, these approaches exclude the possible implicit nature of meta-rules and their close interaction with lower level rules, as is suggested by Cary and Reder (2002). Therefore, alternative new explanations should be investigated. In particular, it is necessary to explore new mechanisms capable of encoding and manipulating meta-rules, which are not constrained by human assumptions. These mechanisms may provide a basis for formulating theories that sufficiently explain general cortical meta-cognitive processes in the brain. This is the approach followed in the current work.

More specifically, we investigate meta-rule phenomena following a minimum constraint approach, avoiding the abstraction of predefined roles at different parts of the cognitive system. Therefore, the dynamics of the system are free to self-organize in any appropriate way, revealing new and potentially more natural mechanisms for explaining meta-level cognition. To this end, we have designed a robotic task that requires the development of meta-level rules to be used for the manipulation of simple behavioral rules. In short, according to our experimental scenario a simulated robotic agent has to consider unpredictably changing reward signals in order to switch between behavioral rules, choosing the one that is considered correct at a given time period. Therefore, in our experiments, the

formulation of rules which are used for manipulating other rules account for meta-level cognition.

Instead of hand designing the details of the computational model, in this study we evolve continuous time recurrent neural network (CTRNN) robot controllers that accomplish the above-mentioned rule-switching task. We have conducted multiple statistically independent runs using both fully connected and bottleneck (Paine & Tani, 2005) CTRNN topologies, in order to investigate (a) the appropriateness of the network structure and (b) the self-organization of internal network dynamics encoding meta-level rule switching mechanisms.

Our research methodology is based on the combination of evolutionary robotics (Nolfi & Floreano, 2000) and dynamic neural networks (Beer & Gallagher, 1992; Kelso, 1995). The first is essential for exploring embodied artificial central nervous systems that accomplish complex situated behaviors, while the latter provides an adequate framework for investigating the temporal characteristics of cognitive functionality (Van Gelder, 1998). Similar approaches have been followed in previous works exploring systems capable of accomplishing complex cognitive tasks (Di Paolo & Harvey, 2003; Paine & Tani, 2005; Tuci, Quinn, & Harvey, 2002; Yamauchi & Beer, 1996). However, to the best of our knowledge, it is the first time that dynamic neural networks are evolved to investigate behaviors that require meta-level rule manipulation.

The rest of the article is organized as follows. In Section 2 we discuss the motivation behind our experiments. In Section 3 we present the CTRNN architectures used in our study, and Section 4 describes how they are connected to the sensors and actuators of the simulated robotic agent. The behavioral tasks of rule-switching are discussed in Section 5. In Section 6 we present the evolutionary procedure used to explore configurations of CTRNN robot controllers. Experimental results addressing robot switching between two behavioral rules are given in Section 7. Additionally, we briefly describe experiments for robot switching between three rules in Section 8. A detailed discussion in Section 9 highlights the main finding of our work, formulating suggestions for the organization of natural and artificial meta-cognitive processes. Finally, conclusions and suggestions for further work are presented in Section 10.

2 Motivation and Research Methodology

The motivation for our experiments was to provide self-organization pressure on simple neural network models in order to examine possible neuronal mechanisms that account for meta-level cognition and in particular for switching between behavioral rules. Our work is based on tasks that resemble the Wisconsin Card Sorting test, but additionally emphasize sensorimotor interaction. Following this approach, meta-level cognition is integrated with sensorimotor activities, highlighting the inseparable nature of these processes in real behavioral tasks.

More specifically, a simulated robotic agent has to learn a number of sample–response rules and apply them to the given environmental situations. At any given time, only one of the rules is correct. Punishment signals that have been properly specified by the human experimenter indicate the rules that should be avoided. The robotic agent has to explore the environment in order to find out which sample–response rule is considered correct at a given time. This rule should be repeatedly applied to specify robot decisions for a number of forthcoming behavioral trials. However, at some unpredictable future time, the experimenter changes the rule considered correct, by relocating punishment signals. In the next trial, the robot, which is not aware of this change, will respond according to the previous rule, and it will be punished. Therefore, the agent has to revise the adopted response strategy, considering that in order to avoid punishments, it has to quit the previous rule and discover the new correct one. When the robot switches to the new rule giving correct responses for some trials, the correct rule is changed again by the experimenter, and the robot has to re-adapt its response rule strategy. This process of unexpected rule switches by the experimenter, is continued for a large number of agent's trials.

To the best of our knowledge, this is the first time that WCS has been interpreted as a mobile robot test.² However, other related robotic tasks have appeared in the literature investigating situated robotic behaviors. In particular, context-dependent action selection, that addresses the “what to do next?” question for robotic agents, describes the class of problems more closely related to our task. This type of problem is commonly tackled by multi-expert approaches (Bryson, 2001; Hu & Edwards, 2006) where a set of specialized behavio-

ral modules compete for gaining the focus of processing (the current context of the robot is used to bias the selection of the appropriate behavioral module). Multi-expert approaches are widely applied and they seem adequate also to address the problem investigated in this study. Well known drawbacks of such approaches are the predefined number of the specialized modules used in the system, and the poor generalization that is usually achieved from each module. Additionally, they follow a rather artificial, very much solution-oriented approach, that can hardly be parallelized to natural brain cognitive processes, where a single module switches behaviors depending on context. A rather small number of works have investigated compound neural network structures (rather than a set of experts) switching functionality according to the current environmental situation. For example Meeden (1996) uses an Elman network to investigate a two-level behavior with periodically alternating action goals (either approach a light source, or avoid it) that are indicated by environmental reward signals. This is similar to Ziemke (1996) who employed second order recurrent networks with explicit input units selecting the appropriate goal-directed behavior at a given time. Both works show context-dependent functionality which is, however, switched by explicit external inputs. In a more complicated situation Ziemke (1999) investigated the suitability of first and higher order neural network types for behavior switching. The designer of the neural networks assigns different roles to the components of the models, specifying a memory unit that is activated or deactivated based on the current location of the agent. Depending on the state of the memory unit, the behavior of the agent can switch from avoiding to hitting simulated obstacles and vice versa. The solution with the explicitly defined memory unit is adequate for following one of two possible behaviors but it can not address tasks described by a combination of circumstances producing a larger number of choices.

The works discussed above and the majority of those appearing in the literature, investigate tasks described by a single decision level (i.e., they address context-dependent “action” selection). For example, in Ziemke (1999) the activation or deactivation of a single decision unit is sufficient to accomplish the task because the agent needs to remember only one thing. However, the task investigated in our study can only be described by using multiple decision levels (i.e., it addresses context-dependent “rule” selection). In our

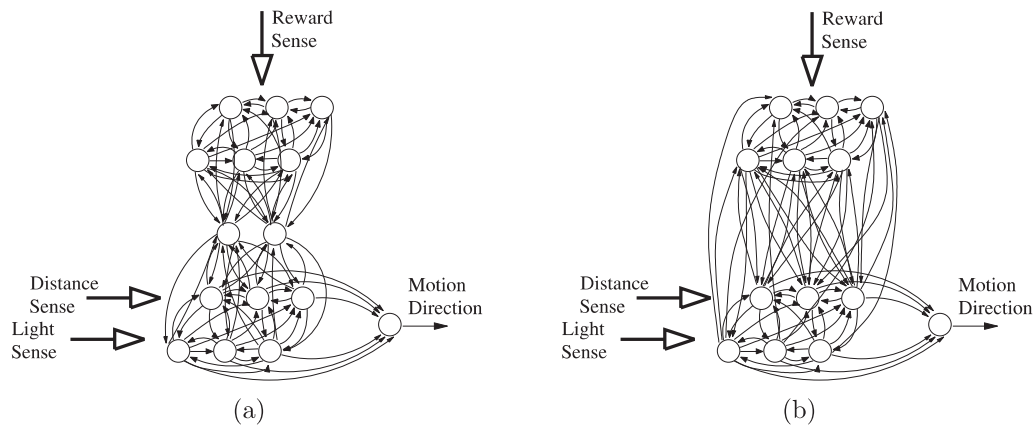


Figure 1 Schematic representation of (a) the bottleneck CTRNN and (b) the fully connected CTRNN.

task consisting of a series of sample–response trials, the agent has to memorize (a) the currently adopted response rule, and (b) the side (left or right) of the sample cue. Thus, the agent is required to combine the states of two memories interpreting successfully all possible sample–response actions. Furthermore, in order to switch successfully between behavioral rules, the agent has to evaluate its own state (i.e., the currently adopted response strategy) implementing a mechanism to reset it appropriately. Therefore, the task investigated in this study combines context-awareness (i.e., sample cue) with self-awareness (i.e., current rule). Clearly, following the X-about-X interpretation of meta-cognition, the above mentioned two-level structure that uses rules about rules, corresponds to meta-cognitive processing.

In summary, in order to solve the task investigated in this study, the robotic agent has to develop meta-level cognition capable of (a) monitoring its own rule strategy, and (b) changing the currently adopted rule, avoiding punishment signals during rule alternation. Although this sounds like a very complex computational task in the conventional cognitive science sense, we will show that embodied neuro-dynamical systems can solve the problem in a rather simple manner.

In this study we employ evolutionary algorithms to investigate meta-level switching between two- and three-sample–response rules. However, we keep our main focus on the two-rule case because (a) we can more easily study the neural network models getting a clear insight of the self-organization of internal dynamics, and (b) the evolutionary processes can have higher

success rates, providing increased validity to our findings. The three-rule switching experiments are investigated in our work in order to evaluate the generalization of our conclusions in a more complex case.

3 CTRNN Configurations

In order to address the continuous mode of brain operation we need an artificial neural network capable of simulating the temporal nature of cognition. The continuous time recurrent neural network (CTRNN; Beer, 1995; Yamauchi & Beer, 1996) is one of the most widely used network types for addressing dynamic phenomena in the literature (Arie, Namikawa, Ogata, Tani, & Sugano, 2006; Bown & Lexer, 2006; Tuci, Trianni, & Dorigo, 2004). Therefore, in this study, we use this type of network to investigate possible mechanisms of rule switching in meta-level, and the contextual temporal changes of cognitive processes. In particular, we want to investigate how meta-level mechanisms self-organize in CTRNN neuronal dynamics under the given task pressure.

We employ both bottleneck (BN; Paine & Tani, 2005) and fully connected CTRNN topologies (see Figure 1), in order to explore what kind of network structure is essential for achieving meta-level functions. As shown in Figure 1a, a bottleneck CTRNN is squeezed in the middle in order to loosely separate the network into upper and lower parts that interact through bottleneck neurons. Thus, information processing is partly segregated in two different levels with BN neurons maintaining interaction between them. In contrast, in

the fully connected case (see Figure 1b) information processing levels can hardly be differentiated. As is shown in Figure 1, the CTRNN consists of $N = 15$ neurons for the case of the bottleneck structure and $N = 13$ neurons for the case of fully connected structure. In both cases, neural topologies are organized with full neuron-to-neuron and input-to-neuron connectivity. Synaptic weights are determined by an evolutionary procedure (described below) and they remain constant during task testing.

Additionally, we are interested to include in our model some of the basic characteristics of cortical organization. In the mammalian brain, it is well known that the reward information is projecting (through VTA) to the prefrontal cortex, which is a module with higher level cognitive responsibilities, while other somatosensory modalities are directly connected to lower level motor modules such as primary motor cortex (Kandel, Schwartz, & Jessell, 2000). In order to abstractly mimic this architecture, we separate reward from the other sensory modalities. The simulated reward signal is directly connected to the upper part of the CTRNN responsible for higher level cognition, while the wall distance and lighting information is linked to the CTRNN part with primary motor responsibilities. This is graphically depicted in Figure 1.

Similarly to previous studies (Paine & Tani, 2005; Yamauchi & Beer, 1996) CTRNN neurons are governed by the standard leaky integrator equation:

$$\frac{d\gamma_i}{dt} = \frac{1}{\tau} \left(-\gamma_i + \sum_{k=1}^R w_{ik}^s I_k + \sum_{m=1}^N w_{im}^p A_m \right) \quad (1)$$

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 presynaptic neurons A weighted by w^p . After estimating the neural state by Equation 1, then the activation of the i th neuron is calculated by the non-linear sigmoid function according to:

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

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

One important characteristic of the CTRNN is the contextual memory that is represented by internal neuron dynamics. In particular, in our experimental setup the neuronal state γ_i is initialized only once, and then neuronal dynamics proceed continuously for the remaining steps of robot behavior (i.e., without resetting γ_i). This property makes CTRNN particularly appropriate for addressing (at an abstract level) the temporal nature of cognition, since cortical processing is also continuous.

4 Robotic Platform and Input–Output Connectivity

In order to investigate embodied rule switching, we employ a two-wheeled simulated robotic agent equipped with eight uniformly distributed distance, light, and reward sensors. The experiments discussed here have been carried out using YAKS,³ 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 type of sensor that supports feeling the special environmental signals simulating negative rewards).

The connection of the network to the sensors and the actuators of the simulated robot is illustrated in Figure 1. All neurons in the lower part of the CTRNN receive information from light and wall distance sensors. The upper part neurons receive punishment information as a tonic input which works as a neuromodulator to the neural activations. The two layers of the network have to communicate in order to accomplish the meta-level rule-switching task. The neurons of the lower level, project to a motor neuron that sets the relative speed of the left and right robot wheels, specifying steering. Like all other neurons, the motor neuron is also governed by Equations 1 and 2. Let us assume that at a given time step, the activation of the motor neuron is A_m . Then, the left and right wheel speed of the simulated robot is given by:

$$\begin{aligned} speed_l &= 0.4 + 0.6A_m \\ speed_r &= 0.4 + 0.6(1 - A_m) \end{aligned} \quad (3)$$

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

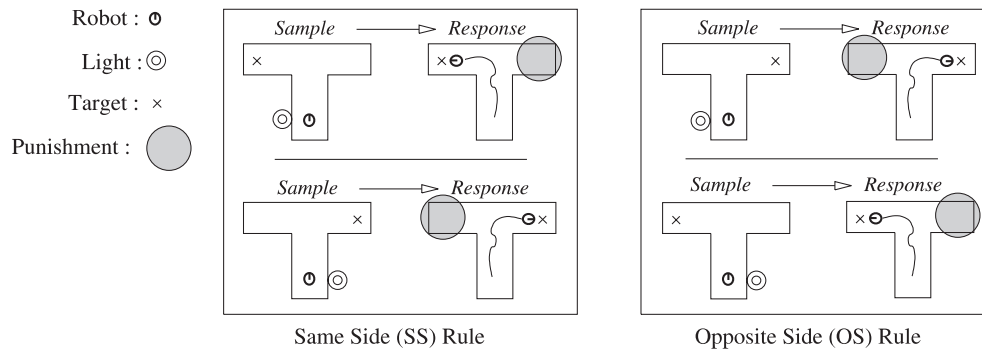


Figure 2 A graphical interpretation of the two behavioral rules used in our experiments. Light cues are represented by double circles, target locations by an \times , punishment areas by gray circles, and the robot path with a black line starting from the bottom of the T-maze.

5 Behavioral Tasks

The tasks investigated in this study are based on a robotic version of Wisconsin Card Sorting (WCS) test that emphasizes dynamic sensorimotor interaction. Therefore, the mobile robot WCS test described below considers both meta-level cognition and sensorimotor coupling as inseparable parts of a complex behavioral problem. Similar setups have appeared in rodent WCS-like experiments (Joel, Weiner, & Feldon, 1997).

Specifically, we assume that a simulated robotic agent is located in the lower part of a T-maze (width 62 cm, height 48 cm), and at the beginning of a trial, a light sample appears at its left or right side (Ziemke & Thieme, 2002). The agent has to respond by moving to the end of the corridor and making a 90° left or right turn, depending on the side of the light sample. Similarly to Maniadakis and Trahanias (2006), two different response rules are defined (see Figure 2). 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. The complementary response rule named opposite-side (OS), implies that the robot has to turn to the opposite direction from the light source sides. For both rules, when the agent responds incorrectly, it drives into a negative reward area where it receives a punishment indicating that it is not following the correct rule. During the task, the above-described trial is repeated many times by resetting the robot to the start position (the trial $t + 1$ starts immediately after trial t , without any time delay).

From time to time, unknown to the robot, the experimenter switches the correct rule (from OS to SS and vice versa) in an unpredictable manner. For every rule switch, the robot will obviously produce some incorrect responses because of the unpredictable change. In these incorrect trials, the agent will receive punishments indicating it is no longer following the correct rule. Then, for the subsequent trials, the robot has to adapt its response strategy to the new rule. After some more trials the rule is switched again, and so on. It is noted that we do not reset the neural state of the robot controller when passing from one trial to the other, thus keeping neurodynamics continuous across trials.

The details of the experimental procedure are described below.

5.1 Task Setup

The overall task is structured into $P \in \{1 \dots 10\}$ phases, with each phase p including T_p trials. The number of trials $T_p \in \{8, 10, 12, 14\}$ is randomly specified, so that the agent cannot predict the end of a phase. During phase p , the agent has to follow the same response rule for all T_p trials, as defined by the experimenter. Let us assume, for example, that the experimenter has selected the SS as the correct rule for the current phase. Each one of the T_p trials tests the response of the robot after the light sample appearance at its left or right side (their order is randomly chosen). When a trial starts, the robot senses the light and then it moves to the end of the corridor where it makes a turn choice. According to the SS rule, the response is correct when the robot

turns toward the side of light sample. If the robot makes the correct choice, it drives close to the target location where no punishment exists. In the case when the robot turning is not correct, it will drive into a punishment area receiving negative reward indicating that the currently adopted rule is wrong and it should be switched. During phase p , the robot is given six “free” exploratory trials to discover the current correct rule. We call the first six trials of the phase “free,” because the corresponding responses given by the robot are not considered in the evaluation of the overall task accomplishment (see also below). In other words, the robot is allowed to give at most six incorrect responses, without cost. In the remaining $T_p - 6$ trials the performance of the robotic agent is evaluated in terms of following the desired response rule specified by the experimenter. If any of these trials is incorrect, the task is immediately terminated (without completing the current phase, and without investigating the next phases).

If phase p is completed successfully, the robot moves to phase $p + 1$, where the experimenter switches the correct response rule, to OS for our example. This means that the punishment signals have been relocated, and they are now positioned according to the OS rule. However, the agent is not informed that the rule has been switched and thus in the first trials of the phase $p + 1$ it will continue responding according to the previous rule. Obviously, the agent will drive into the punishment areas, indicating it is not following the correct rule. In order to avoid punishments, the robot must reconsider its rule choice, adopting the OS response strategy. In phase $p + 1$, the robot is given again six “free” exploratory trials to discover rule switching. In the remaining $T_{p+1} - 6$ trials agent’s responses are evaluated according to the current correct rule (as defined by the experimenter).

If phase $p + 1$ is completed successfully, the robot moves to phase $p + 2$, where the response rule is switched again—to SS for our example—and the same experimental procedure is repeated. Overall, the task evaluates the agent’s ability to switch between behavioral rules for a maximum of P phases (if all of them are completed successfully).

6 Evolutionary Procedure

In order to explore how the dynamics of rule switching self-organize in CTRNNs, we use genetic algo-

rithms.⁴ In short, we use a population of artificial chromosomes encoding CTRNN controllers (their synaptic weights and neural biases are as described in Section 3). Each candidate solution encoding a complete CTRNN is tested on tasks examining the ability of the network to switch between rules. The tasks investigate robot behavior for several phases each one consisting of many trials as described above (consecutive phases correspond to different correct rules). Fitness values are assigned to each CTRNN controller evaluating its performance on the given tasks. The scores accomplished by the controllers are used to sort and evolve the population of chromosomes, therefore producing a new generation of CTRNN controllers that is ready for evaluation. This iterative procedure is repeated for a predefined number of generations. The details of the evolutionary procedure are described below.

As we are interested in the broader set of possible mechanisms that can give rise to meta-level rules, we do not explicitly specify any internal dynamics in the model. Therefore, the network is allowed to self-organize in any appropriate way, developing partial functionalities that support the accomplishment of the given task. We run several independent evolutionary processes in order to find common network features appearing in all the proposed solutions, which are probably characteristics for the desired CTRNN configuration.

6.1 Incremental Evolution

Because of the complexity of the investigated task it is difficult for the evolutionary process to converge successfully when evolving neural controllers from scratch whilst examining all the details of the problem. Therefore, in order to support the success of the evolutionary procedure we follow an incremental approach, investigating gradually more complex versions of the rule switching task. This is summarized in Table 1. The first generations of the evolutionary procedure aim at CTRNN controllers capable of adopting both SS and OS rules. Two different tasks—each one consisting of only one phase—are employed to evaluate robot performance. At the beginning of each task, the states of all CTRNN neurons are reset to zero (i.e., the robot is in a neutral state, without following any rule). Then, the robot explores the environment in order to discover which rule it has to adopt for the successful completion of the task, avoiding punishment signals.

Table 1 The incrementally more complex tasks solved in different parts of the evolutionary procedure investigating switching between two rules.

Evolutionary procedure For punishment-guided two-rule switching		
Generations	Task type	Description
1–60	<i>Task 1</i> : CTRNN reset- SS <i>Task 2</i> : CTRNN reset- OS	Single Phase
61–140	<i>Task 1</i> : CTRNN reset- SS → OS <i>Task 2</i> : CTRNN reset- OS → SS	Two Phase
141–300	<i>Task 1</i> : CTRNN reset- SS → OS → SS → OS ... SS → OS <i>Task 2</i> : CTRNN reset- OS → SS → OS → SS ... OS → SS	Multiple Phase

The accomplishment of *Task1* implies that the robot can adopt the SS rule, while the accomplishment of *Task2* implies that the robot can adopt the OS rule.

In the next generations, the tasks become more complex, searching for controllers capable of switching between rules. Specifically, during generations 61–140, we explore tasks consisting of two phases, selecting controllers capable of making one rule-switching step. *Task1* investigates the ability of CTRNN controllers to first adopt the SS rule and then switch to OS. In a similar way, *Task2* examines controllers' ability to adopt OS and then switch to SS. In each of the two phases, properly positioned punishment signals indicate the response strategy that should be avoided. We note that 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, like the natural cognitive processes, special memory pathways have to develop in order to support rule switching from OS to SS and vice versa.

Finally, during generations 141–300, we explore the stability of the rule-switching mechanism. In particular, we investigate the performance of CTRNN controllers under multiple and unpredictable changes of the correct rule. Both *Task1* and *Task2* consist of a 10-phase sequence. The performance of the agent is evaluated on phase p only if it has been successful on phase $p - 1$. Similarly to the previous evolutionary 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 (i.e., switching between SS and OS).

6.2 Fitness Measure

The accomplishment of the task is evaluated on a trial-by-trial basis, based on properly specified target positions. For every trial, a target position is defined on the top-left or top-right side of the T-maze, according to (a) the current rule, and (b) the side of the light sample. This is graphically depicted in Figure 2. The switching of rules between consecutive phases will specify a different set of target positions in the corresponding trials.

Let us 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 $d_{min} \in [0, D]$ can be used for measuring the success of the robot turning choice in a single trial. We use this target-reaching measure over all trials and all phases, to evaluate the performance of the agent on tasks 1 and 2. Overall, the ability of the CTRNN controller to switch between rules during the p phases of a task i , is measured by:

$$E_i = \sum_{q=1}^p \left(\sum_{t=7}^{T_q} \left(1 - \frac{d_{min}}{D} \right) \right) \quad (4)$$

The evaluation starts from trial $t = 7$ because the first six trials of each phase are exploratory and they are not considered in the evaluation. The higher the value of E_i the more rule switches the agent has accomplished.

All individuals encoding CTRNN controllers are tested on the incrementally more complex versions of

Task1 and *Task2* described above (see Table 1). The accomplishment of each task is separately evaluated according to Equation 4. Then, the total fitness of the individual is estimated by:

$$fit = E_{Task1} \cdot E_{Task2} \quad (5)$$

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

6.3 Computational Details

In order to evolve CTRNN configurations, we have used populations of 500 individuals. Real-value 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 has been set to 0.25 for all neurons.

Each candidate CTRNN configuration is tested on both tasks described above, evaluating agent's rule-switching capacity over several phases. At the beginning of each trial, the robot is located at a predefined starting position with its direction randomly specified in the range $85-95^\circ$ (the direction of the corridor corresponds to 90°). The robot is kept in the same initial position for five simulation steps, and then it is allowed to navigate freely in the environment for 165 more simulation steps. 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 experiment for the next trial.

A standard genetic algorithm with mutation, but without crossover, evolves populations, driven by the fitness function described in Equation 5. In particular, at the end of each epoch, the $S = 30$ best individuals of the population are used as a basis for producing the individuals of the next generation. The new individuals are generated by randomly selecting and mutating one of the S individuals. Mutation corresponds to the addition of up to 30% noise, in the parameters encoded to the chromosome, while each parameter has a probability of 4% to be mutated.

7 Results

In our experiments we have investigated possible meta-level rule-switching mechanisms for both the fully con-

nected and the bottleneck CTRNN, conducting 10 independent evolutionary runs for each network topology. For the case of the bottleneck CTRNN, eight out of the 10 evolutionary processes converged successfully producing controllers capable of accomplishing the given tasks. However, only three out of the 10 evolutionary processes converged successfully for the case of the fully connected CTRNN. These results highlight the advantageous effect of bottleneck neurons that divide the network into partially segregated parts, each one developing a different role in the functionality of the whole system. Because of the significantly better performance of the bottleneck CTRNN, for the rest of the section we will concentrate our study on the results of the bottleneck topology.

The behavior of the robotic agent for one representative bottleneck CTRNN is demonstrated in Figure 3. During trials 1–4 the robot is successfully following the opposite side (OS) rule. Then, in the fifth trial the rule is unexpectedly changed to same side (SS), and the agent produces a wrong response driving into the punishment area. At that time, the agent has to understand that its current response strategy is no longer correct, and it has to adopt another response rule. The agent immediately switches to the SS rule, responding successfully for the next 11 trials, avoiding punishment signals. The rule is unexpectedly changed again in trial 17, where the robot gives a wrong response again driving into the punishment area. This time it takes two trials for the agent to switch back to the OS rule. After that, the agent keeps the same rule giving correct responses in the subsequent trials.

Interestingly, robot paths are significantly correlated with the currently adopted rule. For example, every time the robot turns left according to the SS rule it follows very similar trajectories (compare trials 8, 9, and 15 in Figure 3). The same is also true when it turns to the right for the same rule (see trials 12, 13, and 16 in Figure 3). A similar relationship can be observed for the paths of the OS rule (e.g., compare right turns in trials 3, 20, and 21, and additionally compare left turnings in trials 19, 26, and 27). However, by comparing same-side turnings of different rules, we can see different trajectory characteristics (e.g., comparing trials 12 and 13 with trials 24 and 25). This means that robot trajectories are somehow involved in distinguishing the two rules. In other words, the CTRNN controller takes advantage of its embodiment and environmental interaction in order to keep track of the currently

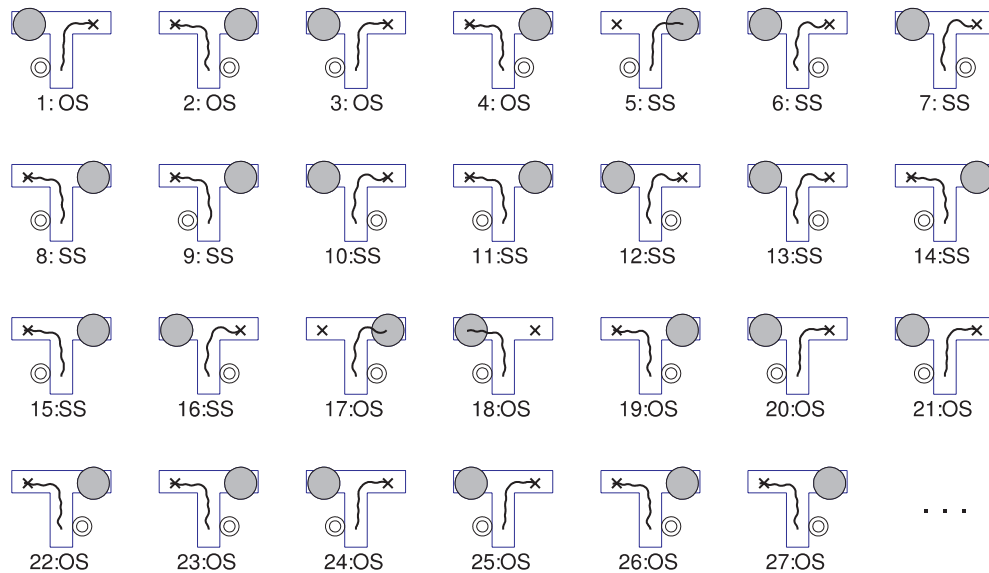


Figure 3 The behavior of the agent in a sequence of trials. The notation is the same as in Figure 2. In this figure we follow a more compact representation of a sample–response trial than the one shown in Figure 2, in order to depict an adequately large number of robot trials.

adopted rule. Taking a closer look at the behavior of the agent, we see that it is reacting in different ways to the appearance of the light at the beginning of trials. In particular, when the light source appears to the left side of the robot, it moves slowly right when following the OS rule (e.g., trial 1), while it makes a faster rotation to the same side when following the SS rule (e.g., trial 8). In contrast, when the light appears to the right side of the robot, it moves very fast to the left when following the OS rule (e.g., trial 4), while it moves rather slowly to the left when following the SS rule (e.g., trial 7). These reactions at the beginning of trials are adequate to bring the robot into distinct sensorimotor states which are easily tracked in the remaining steps of environmental interaction.

Additionally, all the trials depicted in Figure 3 show that the robot has developed a wall-avoidance functionality that is properly triggered when it drives close to the walls. This emergent behavioral component, that is not explicitly described in the fitness function of the evolutionary process (see Equations 4 and 5), has developed because of the robot's need to reach target locations, navigating to the left and the right side of the T-shaped environment. Therefore, it is the triggering of wall avoidance that shapes the distinct patterns of robot–environment interaction.

We note that behaviors similar to the one described above have been shown by the controllers obtained in all successful evolutionary runs. In fact, despite statistical independence of the evolutionary procedures, they have all produced CTRNNs with very similar internal dynamics. Therefore, the characteristics that consistently appear in all CTRNN solutions can be parts of a valuable alternative scenario for the cortical processes involved in meta-level rule switching. This is described below.

In order to obtain insight into CTRNN dynamics, we have investigated internal neural activity separately for the two rules. We found that the activations of neurons in the higher and the lower levels show different qualitative characteristics (see Figure 4). In particular, the activity of higher level neurons remains almost constant during the whole trial, and largely the same for the left and right turnings. Therefore, we conclude that their activation level is directly linked with the currently adopted rule. In contrast, the activity of lower level neurons varies during the trial, indicating their involvement in the execution of higher level rule orders, also taking into account environmental interaction issues, such as wall avoidance.

It is noted that the neurons in the higher level show a two-state activation mode, selecting the rule that is

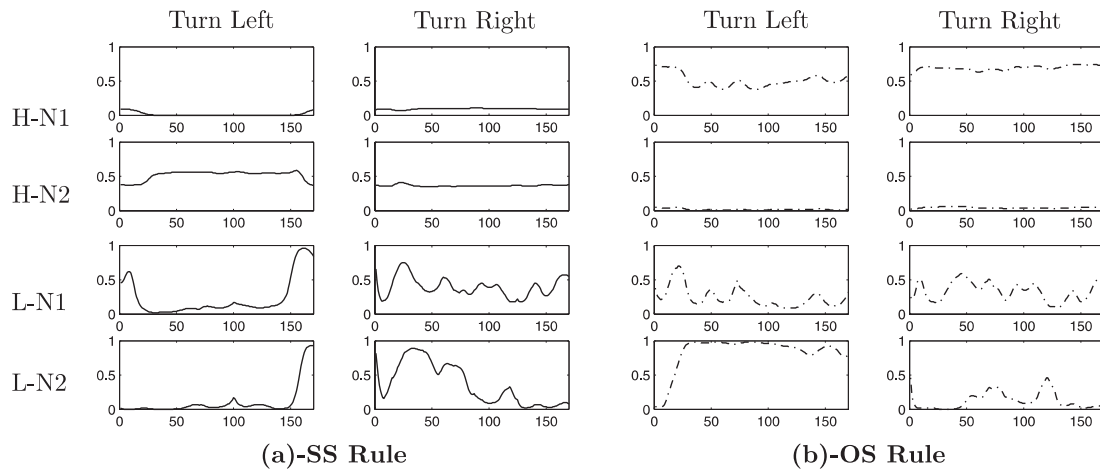


Figure 4 The activation of two higher (H-N1, H-N2) and two lower (L-N1, L-N2) level neurons when the agent follows (a) the SS rule and (b) the OS rule. SS is depicted with a solid line while OS is depicted with a dashed line.

currently adopted by the CTRNN controller. The different patterns of higher level activity bias the functionality of the whole system to respond according to the given rule. However, the lower part of the network is responsible for applying the rules. In particular, lower level neurons exhibit different activation patterns when the agent turns left or right in the context of the same selected rule, which means that they have also taken into account light information.

Overall, a two-step rule-manipulation mechanism accounts for the meta-level functionality of the CTRNN controller (i.e., the rules being applied on top of other rules). Specifically, during unexpected rule changes, the agent has to switch the rule that is currently selected in the higher level, implying also a change in the dynamics of the lower level. Therefore, the rules being applied on top of other rules are as follows. In the behavioral plane, the SS and OS response rules are stored. At the same time, in the meta-cognitive plane, the rule “if you are punished, then find a new behavioral rule, otherwise keep following the previous behavioral rule” is implemented. The interaction of the behavioral and meta-cognitive plane modulates the behavior of the agent accomplishing the desired tasks.

We turn back now to the results of Figure 3, and their relationship to the neural activities of Figure 4. We have commented above that we observed very similar behaviors every time the robot responds to the same side, following a given rule (e.g., for all left turns of the SS rule). Additionally, very similar activation patterns are observed in the higher and lower

level neurons in each one of these cases. This means that the composite CTRNN controller has stored internally a set of different behavioral procedures which are properly selected and expressed based on the activity of the higher level neurons and the sensory light input. This emergent functionality is similar to that of parametric bias neurons (Nishimoto & Tani, 2004; Tani & Ito, 2003) that has been shown to facilitate storing and recalling many behaviors to the same network.

It is worth emphasizing that after conducting attractor analysis, neural characteristics correlated to SS and OS rules are identified in both the higher and the lower part of the CTRNN. Specifically, for each rule, we have asked the agent to perform 1,000 random turning trials (either to the left or to the right) after multiple perturbations are applied to neural states in the initial trial. We observed that after each perturbation, the agent’s behavior always converges back to the OS or SS response strategy (depending on the punishment signal positioned by the experimenter, which specifies the currently correct rule), implying that rule-based attractors have probably emerged in the network dynamics. Therefore, we investigated the phase plots of the higher and lower level neurons. They are demonstrated in Figure 5. For each rule, the same shape of attractor appears in the plot, regardless of the random perturbation. As expected, separate invariant sets of dynamical trajectories states are observed in the higher level. Additionally, we can clearly see that distinct invariant sets have also emerged in the lower level, each one corresponding to a different rule. The higher

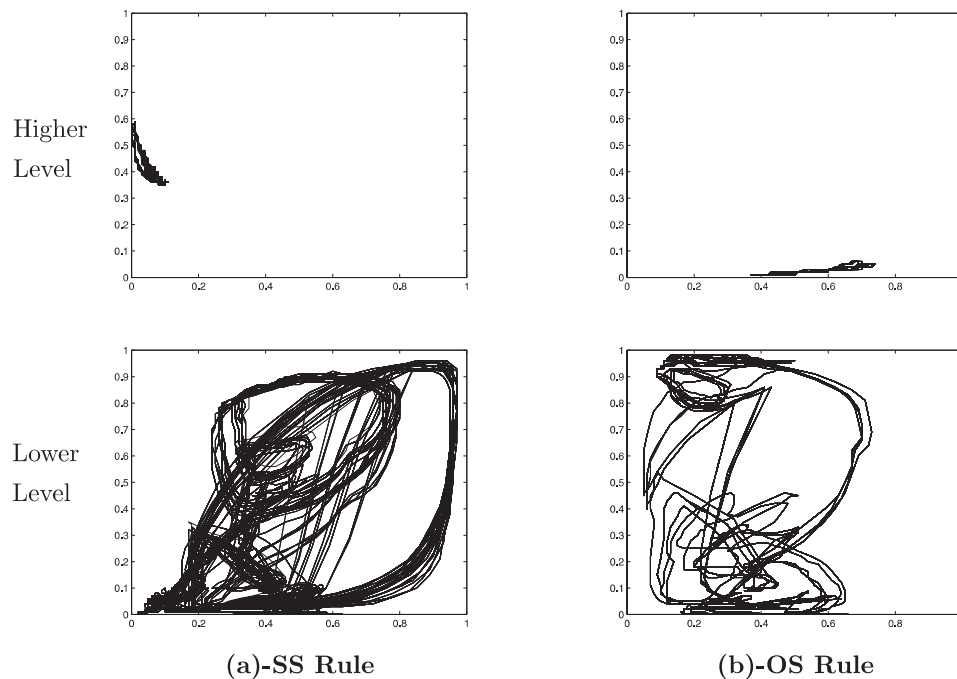


Figure 5 The phase plots of higher and lower level neural activity when the agent follows (a) the SS rule, and (b) the OS rule. In the upper figures the axes x, y corresponds to the activity of neurons H-N1 and H-N2, while in the lower figures the axes x, y corresponds to the activity of neurons L-N1 and L-N2 (all neurons are the same with those depicted in Figure 4). Obviously, neural activities stabilize to attractors having distinct shapes for each case.

and lower level attractors that appeared for each one of the rules are the same for all random initial perturbations. It is important to note that the development of rule-correlated dynamical states has been observed in the CTRNN of all successful evolutionary runs, implying that attractor dynamics might be an important general mechanism for manipulating rules in continuous time systems.

Finally, we have investigated neural dynamics during rule switching (see Figure 6 and Figure 7). According to the experimental scenario used in our study, unpredictable rule changes will make the robot produce wrong responses driving into the punishment area. In that case, the robot will receive an unexpected punishing signal that destabilizes the attractor state of the system. The effect of punishment is more drastic to the higher network because the punishment provides direct neuromodulations to those neurons. The rule-state instability facilitates rule transition from SS to OS and vice versa. This is the dynamical mechanism accounting for rule switching. For example, in Figure 6 two wrong responses are necessary for the robot to accomplish rule switching, while in Figure 7 one wrong

response is enough for the agent to make a rule transition.

Recall that the current experimental setup provides the agent with six free trials to adjust to an unpredictable rule change. This is done in order to support the convergence of the evolutionary procedure. For most of the CTRNN controllers obtained, one or two trials are enough to accomplish rule switching. A small number of successful controllers need more time, with five trials being always enough for a rule state transition. This is in accordance with preservation (low flexibility) phenomena observed in human WCS studies (Kaplan et al., 2006). It is worth noting here that, according to our experiments, we can easily decrease rule transition time by making more evolutionary epochs. However, this approach might result in overfitted and fragile neuro-controllers which are not eligible for the current study. Since our work does not focus on the speed of switching between rules, that is, our experimental setup and the fitness function (see Equations 4 and 5) are not designed in this direction, we have avoided examining CTRNNs that accomplish rule transitions in a single trial.

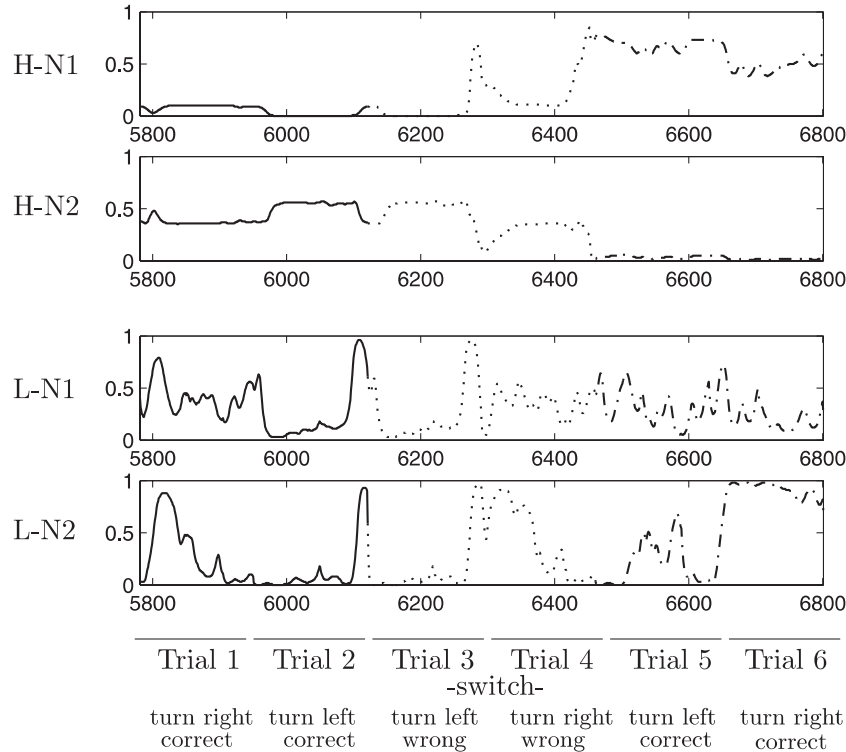


Figure 6 Neural activity during switching from the SS to the OS rule. In the first two trials the agent follows the SS rule giving correct responses (solid line). In the third trial the rule has changed to OS, therefore the agent gives an erroneous response (dotted line). However, this is not enough for switching the rule adopted by the robot. Therefore, the fourth trial is also wrong. In the fifth trial the agent tries the OS rule that is correct (dash-dotted line), and therefore it continues with the same rule for the subsequent trials.

8 Additional Experiments: A Three-Rule Switching Study

In order to explore if we can consider dynamical systems as a general mechanism of meta-level rule switching, we have conducted further extended experiments investigating the case of switching between three behavioral rules. Specifically, in addition to the OS and SS response rules discussed above, we introduce one more rule, named no response (NR). According to NR, the robot should ignore light information staying close to the starting position. The introduction of the third rule modifies slightly the experimental setup followed in our robotic tasks. This is because in each trial of the robot, two punishment areas now have to be specified, in addition to the target position. The experimental setup followed for each rule case is illustrated in Figure 8.

The tasks used in this set of experiments are in principle similar to those described above, but now

the agent has to accomplish rule switching between NR, SS, and OS. Therefore, the current tasks are much more difficult than the two-rule case, because during switching the robotic agent has more than one choice for finding the new correct rule specified by the experimenter. Each trial is again separated into P phases. Because of the high complexity of the given tasks, each phase consists of $T_p \in \{12, 14, 16, 18\}$ trials, and the agent is given 10 free exploratory trials to discover the correct rule.

Similarly to the two-rule case, in order to support the successful convergence of the evolutionary procedure we follow an incremental approach, investigating gradually more complex versions of the three-rule switching problem. Additionally, it is important to simultaneously explore all possible switching combinations from one rule to the other. Therefore, we use six different tasks and we select for CTRNN controllers capable of accomplishing all of them. The tasks are described in Table 2. In generations 1–100, we ask

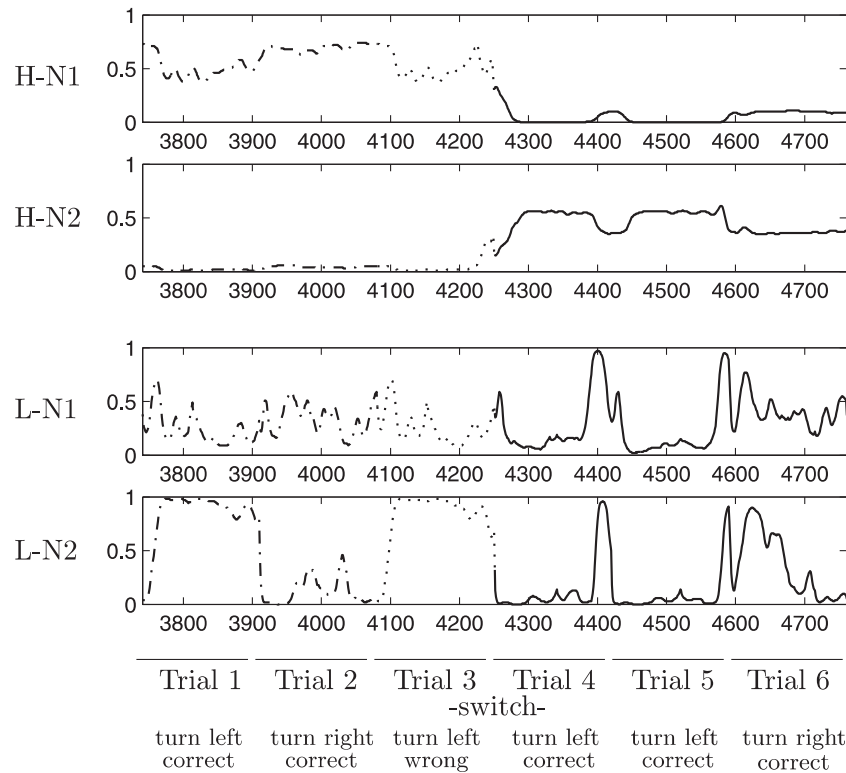


Figure 7 Higher level neural activity during switching from the OS to the SS rule. In the first two trials the agent responds successfully following the OS rule (dash-dotted line). In the next trial the rule has changed to SS, therefore the agent gives an erroneous response (dotted line). In the fourth trial the agent switches to the SS rule that is correct (solid line), and therefore it continues with the same rule for the next trials).

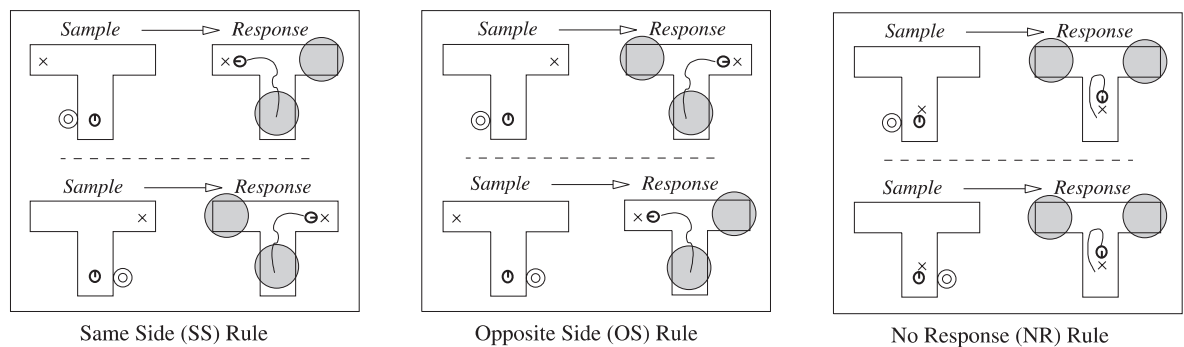


Figure 8 The three behavioral rules used in our second set of experiments. The notation is the same as in Figure 2. Two punishment areas are necessary for indicating the currently correct rule.

agents to separately adopt each one of the given rules. Then, in generations 101–300 the evolutionary process searches for controllers capable of accomplishing a single switching step, investigating all possible transition possibilities. Finally, in generations 301–600

we evaluate repeated rule switching on random rule sequences.

Similarly to the two-rule experiments, and Equation 4 we use target distance information to evaluate if the robot is following the correct rule. Therefore, the

Table 2 The incrementally more complex tasks used in the evolutionary procedure investigating switching between three rules.

Evolutionary procedure for punishment-guided three-rule switching		
Generations	Task type	Description
1–100	<i>Task 1:</i> CTRNN reset- SS <i>Task 2:</i> CTRNN reset- SS <i>Task 3:</i> CTRNN reset- OS <i>Task 4:</i> CTRNN reset- OS <i>Task 5:</i> CTRNN reset- NR <i>Task 6:</i> CTRNN reset- NR	Single Phase
101–300	<i>Task 1:</i> CTRNN reset- SS → OS <i>Task 2:</i> CTRNN reset- SS → NR <i>Task 3:</i> CTRNN reset- OS → SS <i>Task 4:</i> CTRNN reset- OS → NR <i>Task 5:</i> CTRNN reset- NR → OS <i>Task 6:</i> CTRNN reset- NR → SS	Two Phase
300–600	<i>Task 1:</i> CTRNN reset- SS → OS → NR → SS ... OS → SS <i>Task 2:</i> CTRNN reset- SS → NR → SS → OS ... NR → OS <i>Task 3:</i> CTRNN reset- OS → SS → OS → NR ... SS → NR <i>Task 4:</i> CTRNN reset- OS → NR → SS → OS ... SS → OS <i>Task 5:</i> CTRNN reset- NR → OS → NR → SS ... NR → SS <i>Task 6:</i> CTRNN reset- NR → SS → OS → NR ... OS → NR	Multiple Phase

success on task i evaluating robot behavior for p phases, is given by:

$$E_i = \sum_{q=1}^p \left(\sum_{t=11}^{T_q} \left(1 - \frac{d_{min}}{D} \right) \right) \quad (6)$$

The evaluation starts from trial $t = 11$ because the first 10 trials of each phase are exploratory and they are not considered in the evaluation. Then, like Equation 5 the total fitness of an individual encoding a CTRNN controller is estimated by:

$$fit = E_{Task1} \cdot E_{Task2} \cdot E_{Task3} \cdot E_{Task4} \cdot E_{Task5} \cdot E_{Task6} \quad (7)$$

8.1 Results

We have again evolved both bottleneck and fully connected CTRNN controllers, this time using populations of 1,000 individuals. Because of the high complexity of the investigated task, the increased size of the population significantly facilitates the convergence of the

evolutionary procedure. For the case of the bottleneck networks, three out of the 10 evolutionary processes converged successfully, while only one of the processes evolving fully connected networks has been successful. We note that we can enhance the success of the evolutionary procedures by increasing the number of individuals per population; however, the success rates are sufficient to demonstrate the suitability of the two architectures for accomplishing meta-level cognition.⁵ In particular, our results consistently show that, compared with the fully connected configurations, bottleneck CTRNNs are more suitable for addressing meta-cognitive processes (the two-rule experiments also verify this conclusion).

In this article, we will not present a detailed study of the results, because they are similar to the two-rule case discussed above. In particular, CTRNN controllers have developed three different attractors each one corresponding to one of the available rules. Additionally, since we have only one successful controller for the fully connected CTRNN and this number is very limited for inferring safe and valuable conclusions, we

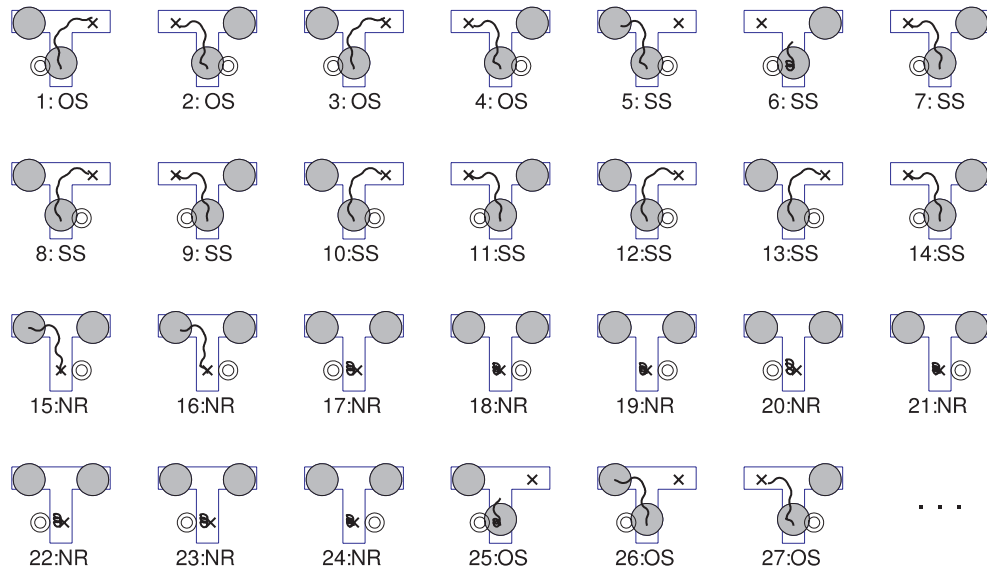


Figure 9 The behavior of the agent in a sequence of three-rule switching trials. The notation is the same as in Figure 2. In this figure we follow a more compact representation of a sample–response trial than the one shown in Figure 8, in order to depict an adequately large number of robot trials.

will concentrate our discussion on the bottleneck topology.

A sample result of the agent behavior while switching between rules NR, SS, and OS, is demonstrated in Figure 9. In the first four trials the robot is successfully following the opposite side (OS) rule. However, in the fifth trial the rule is unexpectedly changed to same side (SS), and the agent, which continues responding according to OS, drives into the punishment area. In the sixth trial the agent explores the no response (NR) rule that is not correct, and it is again punished. In the seventh trial the agent tries the SS rule avoiding punishment, and therefore, this rule is adopted for the next trials. In the 15th trial the rule is unexpectedly changed to NR, making the robot give a wrong response. In the next trial the agent tries the OS rule that is not correct. In trial 17 the robot responds according to NR, avoiding punishment, and therefore it continues with it until trial 25 when the rule is changed to OS. The agent needs two trials to identify the correct rule, avoiding punishment, and so on.

8.2 Rule Transition Dynamics

For the current experimental scenario it is very interesting to explore rule transitions (i.e., the order in which each rule is explored during switching steps),

because during rule changes the CTRNN controller has more than one switching choice (i.e., from NR to SS or OS, from OS to SS or NR, and from SS to OS or NR). In order to investigate rule transition preferences, we have conducted “always-punishment” experiments. In this condition, it is expected that the controller will repeatedly switch the currently adopted rule from the one trial to the other. For each experiment consisting of 1,000 trials, we keep constant the side that the light cue appears on. The results for the case of left light are demonstrated in the top plot of Figure 10. In this case, the SS seems to be the dominant rule, which, however, can switch to OS or NR intermittently without periodicity. This non-periodic rule transition dynamics is also found in other successful CTRNN solutions. Sometimes we observed non-periodic rule switching when the light appeared constantly on one of the sides (e.g., left), while the rules switch periodically, that is, $SS \rightarrow OS \rightarrow NR \rightarrow SS \rightarrow OS$ and so on, when the light appears constantly to the other side (e.g., right).

However, when the same “always-punishment” analysis was conducted for the case of the two-rule switching experiments, it always generated a behavior with a period of two, that continuously switches between SS and OS (see the lower plot in Figure 10). This behavioral difference might be the result of the obvious, straightforward choices in the two-rules

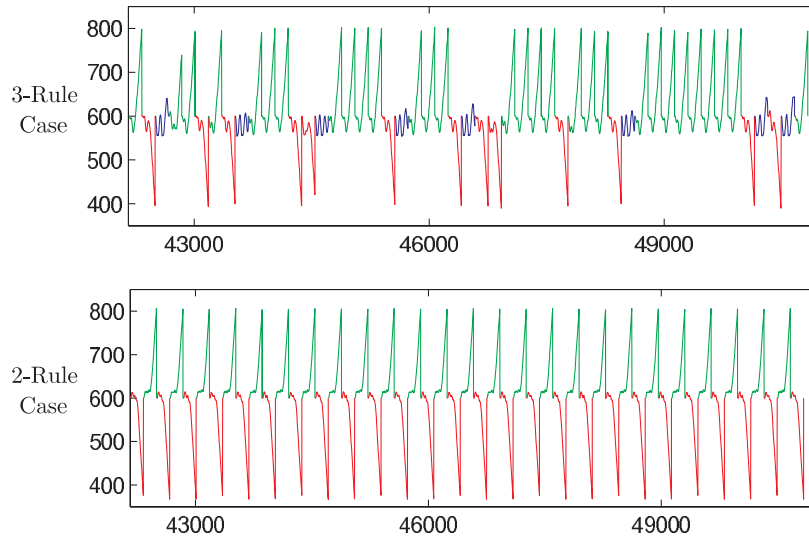


Figure 10 The path of the robot during continuous punishment, with the light appearing always to the left. Each plot demonstrates robot turnings (i.e., its position on the x-axis), against simulation steps. The green parts correspond to the SS rule, the red to the OS rule, and the blue to the NR rule. The top plot shows rule transitions for the 3-rule case, while the bottom plot shows transitions for the 2-rule case.

switching case, which are extended to fuzzy, non-straightforward choices between the remaining two rules in the case of three-rule experiments. Therefore, we believe that the complexity in the task with three rules, pressures the evolved controller to develop more complex dynamics.

9 Discussion

The experiments described in this article investigate tasks requiring meta-level rule manipulation. We need to stress here, that the goal of our study was not the pure solution of the rule-switching problem. This can be accomplished rather easily with many human hard-wired approaches. For example, previous works have used mixture of expert neural networks each one specialized on a different rule, being activating by an appropriate gating mechanism (Bryson, 2001; Hu & Edwards, 2006). However, these solution-oriented approaches cannot sufficiently address important characteristics of brain functionality such as (a) the coexistences of different levels of operational processes (i.e., cognitive and meta-cognitive) in a single network consisting of massively parallel interactive elements, (b) the encoding of multiple behavioral strategies (i.e., SS, OS, and NR rules) on the same neural units, and (c) the contin-

uous and smooth flow of natural cognitive procedures. In this study we explored the broader set of possible mechanisms accounting for meta-cognitive rule switching, following a minimal hypothesis approach. Our experimental setup emphasized the ability of internal dynamics to self-organize in any appropriate way, eliminating arbitrary human designs.

The results showed that neural mechanisms based on the principles of dynamical systems can be used as an explanatory scenario of meta-level cognition, and particularly for meta-level rule processing (i.e., rules applied on rules). This dynamical systems framework is not very often adopted for explaining brain processes in cognitive science, and we believe it merits attracting more scientific interest.

Previous studies have also investigated behavioral switching by evolving CTRNN controllers. For example sequential turn taking of predator-prey behaviors are examined by Ikegami and Iizuka (2003), and behavioral choices accounting for agent's preferences are examined by Iizuka and Di Paolo (2007). However, the tasks investigated in these works do not require any kind of contextual memory. Other works exploring short term storage of sensory cues in dynamical systems (Ziemke & Thieme, 2002), do not address behaviors based on memory combinations (i.e., in the current experiment the agent memorizes both the side of the

light cue and the adopted response rule). In terms of context-dependent switching, our study can be related to those of Meeden (1996) and Ziemke (1999). However, those works focus on context-dependent switching of behavior primitives such as light-seeking or light-avoiding rather than the switching of rules that can only be described using logical operators (e.g., if you follow the OS rule *and* the light appears from the left, then you should turn right). Furthermore, our work addresses the emergence of complex interactions among different rule levels. In particular, our task enforced the self-organization of meta-rules selecting the correct behavior level rule (either SS, or OS, or NR). The investigation of complex multi-level rule interactions using embodied neuro-dynamic systems is a distinct characteristic of the current study.

In a previous study, we have also investigated rule switching based on positive reward (rather than punishment) signals, which gave us similar results regarding the self-organization of attractor dynamics (Maniadakis & Tani, 2008). Despite the fact that the positive and negative reward signals used in the two studies are simulated with similar artificial sensory modalities, there are significant differences between the two experiments that give separate scientific value to the results of each study. In particular, the two experimental protocols require different dynamic characteristics to develop in each solution. The positive and negative reward experiments differentiate because:

- The two signals appear in complementary temporal patterns to the robot. In the experiments of this study the robot senses a punishment signal rarely, only in case of an erroneous response (normally after unpredictable rule changes). In contrast, in the experiments of Maniadakis and Tani (2008) the robot senses the positive reward frequently, every time it gives a correct response.
- The information that the two reward signals (positive or negative reward) provide to the robot is different. The punishment signals used in this study indicate that the currently adopted rule is incorrect and must switch. In contrast, the positive rewards used in Maniadakis and Tani (2008) indicate that the currently adopted rule is correct and the robot must proceed with it.

Despite the difference between the experiments, attractor dynamics have been observed in all the solutions.

Therefore, the possibility of encoding rules into distinct attractors, with rule switching being represented by state transition from one attractor to the other is a valid and naturally emerging mechanism. We note that the capability of CTRNN models to keep continuous neural states similarly to the internal brain processes provides added value to the proposed dynamical systems explanation of meta-rules, because the same explanatory framework can be used to address a range of meta-level cortical phenomena.

In this study, we have investigated what is the most appropriate system architecture for addressing meta-cognition. Traditionally, modelers use two distinct levels representing pure cognition and meta-cognition. However, according to our findings this is not a strict requirement. Our experiments showed that meta-cognition can be accomplished without explicitly providing functional levels in the architecture. The fact that the investigated task can be solved by the fully connected architectures (success rate is lower in that case) suggests that meta-rules can self-organize even without anatomical or architectural levels. In other words, the hierarchical organization that we interpret in the observed behaviors can be generated by less structured implicit mechanisms, involving whole system dynamics. We need to note here that we do not propose to underestimate anatomical distinction of the motor hierarchy in the brain. Our findings showed that a loose segregation of modules facilitates the emergence of functional levels and additionally enhances the functionality of the global system, as is the case with our bottleneck architectures, showing higher success rates. This may account for why prefrontal cortex, which is loosely segregated but still connected to other motor regions, is involved in meta-level cognition more than other cortical areas. We stress that superior success rates for the bottleneck CTRNNs have been obtained also for the case of positive reward experiments (Maniadakis & Tani, 2008).

Because of the continuous flow of cognition, attractor dynamics are shaped in the higher and lower parts of the model, which operate jointly rather than in isolation. The low level sensorimotor dynamics are significantly correlated with meta-level processes, shaping distinct dynamical states for each rule. Additionally, we would like to remind the reader that every time the agent responds according to a given rule, it follows the same trajectories, which are different than the trajectories followed for the other rule. This means

that the agent's response is interpreted as a particular behavior based on the interactive dynamics of the sensorimotor level and the rule-selection level. The internal sensorimotor states provide decisional cues for higher level mental actions (i.e., switch or not, the adopted rule) as is proposed by Nelson and Narens (1990). Therefore, we argue that physical behaviors of robots can significantly support the meta-level cognitive abilities of the composite system. This is in contrast to existing symbolic approaches on meta-cognition that ignore the importance of embodiment and environmental interaction using human designed schemes to interpret meta-cognitive process as knowledge bases (Conitzer, 2008; Gordon et al., 2008; So & Sonenberg, 2004).

The embodied nature of meta-cognition is in agreement with the view that implicit mechanisms are sufficient for realizing meta-cognitive functionality (Cary & Reder, 2002; Proust, 2003; Reder & Schunn, 1996). This view has been adopted in recent studies modeling meta-cognitive phenomena (Sun, Zhang, & Mathews, 2006). The implicit approach to meta-cognition argues that it is not necessary to have a specialized monitoring subsystem to develop meta-cognitive skills. According to Reder and Schunn (1996), while meta-cognitive strategies are explicit, the mechanism for selecting strategies might be implicitly implemented in the system. This is also the case for our results. Although robot response strategies seem explicit and "logic-based" for external observers, their mechanisms require neither explicit representation of rules, nor logical algorithmic manipulation. What actually exist in their neural domain are self-organized dynamics coupled with sensorimotor realities. Of course, it is necessary to comment here that meta-cognition may incorporate much more complex cognitive processes, and it is not limited to behavioral rule switching. Therefore, it remains open for our future research to investigate more complex meta-cognitive tasks.

Finally, we would like to note that the results of our study can significantly support research efforts in the field of biologically inspired robotics, because they investigate how a single cognitive system can combine higher level cognition with real-time environmental interaction. In particular, we have found that the dynamical systems computational framework can be very important for learning and recalling different behavio-

ral strategies in the same computational system. Moreover our results indicate that loosely interconnected (but not independent) neural components are appropriate for capturing higher level cognition and particularly for representing the interaction between cognitive and meta-cognitive processes.

10 Conclusions

In this study we employed CTRNN models to explore meta-cognitive processes, and particularly those accounting for rule switching. Our experiments have addressed switching between two or three rules showing that a new mechanism based on the principles of dynamical systems can sufficiently explain meta-level cognitive processes. Overall, the conclusions of this study are summarized as follows:

- A set of distinct behavioral rules can be embedded in multiple dynamic attractors with distributed neural representation.
- The meta-level manipulation of behavioral rules does not require the implementation of separate mechanisms, but it can be realized in the dynamics of the overall system by self-organizing attractor-switching mechanisms.
- The bottleneck segregation of information flow between the high level and the low level, enhances the overall network performance. This finding may account for the higher cognitive roles of prefrontal lobe in humans.
- Embodiment is essential even for meta-level cognitive processes because sensorimotor dynamics actively participate in organizing whole system dynamics including meta-level ones.

In the future we will apply the results of this work to humanoids which are complex robotic platforms providing a rather realistic framework for exploring meta-cognition. Our future work is scheduled to follow two main directions. First, we will explore the role that explicit monitoring processes can have in rule-switching tasks. Second, we will explore the generalization of dynamical systems approach investigating if it can sufficiently address other meta-level cognitive processes (e.g., meta-memory and meta-learning).

Note

- 1 Social aspects have also be given to the term meta-cognition (e.g., see Yzerbyt, Lories, & Dardenne, 1998). However, in this study we do not investigate social meta-cognition.
- 2 Our interpretation resembles the tasks used in rodent WCS studies (Joel, Weiner, & Feldon, 1997).
- 3 The simulator has been developed at the University of Skovde, Sweden, and can be downloaded at <http://www.his.se/iki/yaks>
- 4 In this 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.
- 5 We have also performed experiments with 500 individuals per population, giving one successful convergence in 10 runs for the bottleneck architecture, and no success for the fully connected architecture.

References

- Anderson, M., & Oates, T. (2007). A review of recent research in metareasoning and metalearning. *AI Magazine*, 28, 7–16.
- Arie, H., Namikawa, J., Ogata, T., Tani, J., & Sugano, S. (2006). Reinforcement learning algorithm with CTRNN in continuous action space. In *Proceedings of the International Conference on Neural Information Processing, ICONIP-2006*. (pp. 387–396).
- Bateson, G. (1972). *Steps to an ecology of mind: Collected essays in anthropology, psychiatry, evolution, and epistemology*. Chandler Publishing Company.
- Beer, R. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3, 471–511.
- Beer, R., & Gallagher, J. (1992). Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1, 92–122.
- Bessiere, C., Freuder, E., & Regin, J.-C. (1999). Using constraint metaknowledge to reduce arc consistency computation. *Artificial Intelligence*, 107, 125–148.
- Bogacz, R., & Giraud-Carrier, C. (1998). Learning meta-rules of selection in expert systems. In *Proceedings of the 4th World Congress on Expert Systems* (pp. 576–581).
- Bown, O., & Lexer, S. (2006). Continuous-time recurrent neural networks for generative and interactive musical performance. In *Proceedings of EvoWorkshops 2006* (pp. 652–663).
- Bryson, J. (2001). *Intelligence by design: principles of modularity and coordination for engineering complex adaptive agents*. Ph.D. Thesis, Massachusetts Institute of Technology.
- Cary, M., & Reder, L. (2002). Metacognition in strategy selection: giving consciousness too much credit. In M. Izaute, P. Chambres, & P. J. Marescaux (Eds.), *Metacognition: Process, function and use* (pp. 63–78).
- Cazenave, T. (2003). Metarules to improve tactical go knowledge. *Information Science*, 154, 173–188.
- Clancey, W., & Bock, C. (1985). *Representing control knowledge as abstract task and metarules*. Technical Report, STAN-CS-87-1168, Department of Computer Science, Stanford University, Stanford, CA.
- Conitzer, V. (2008). Metareasoning as a formal computational problem. In *AAAI-08 Workshop: Metareasoning: Thinking About Thinking*.
- Cox, M. (2005). Metacognition in computation: a selected research review. *Artificial Intelligence*, 169, 104–141.
- Dastani, M., Governatori, G., Rotolo, A., Song, I., & Torre, L. van der. (2007). Contextual deliberation of cognitive agents in defeasible logic. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS-2007)*.
- Di Paolo, E., & Harvey, I. (2003). Decisions and noise: The scope of evolutionary synthesis and dynamical analysis. *Adaptive Behavior*, 11, 284–288.
- Dunlosky, J. T., & Hertzog, C. (1998). Training programs to improve learning in later adulthood: Helping older adults educate themselves. In J. Hacker, J. Dunlosky, & A. C. Graesser (Eds), *Metacognition in educational theory and practice* (pp. 249–275). Mahwah, NJ: Lawrence Erlbaum Associates.
- Dunning, D., Johnson, K., Ehrlinger, J., & Kruger, J. (2003). Why people fail to recognize their own incompetence. *Current Directions in Psychological Science*, 12, 83–87.
- Fernandez-Duque, D., Baird, J. A., & Posner, M. I. (2000). Executive attention and metacognitive regulation. *Consciousness and Cognition*, 9, 288–307.
- Flavell, H. (1971). First discussant's comments: What is memory development the development of? *Human Development*, 14, 272–278.
- Flavell, H. (1976). Metacognitive aspects of problem solving. In *The nature of intelligence* (pp. 231–235).
- Garner, B. (2000). Meta-knowledge acquisition strategies in asynchronous learning frameworks. In *Proceedings of the International Workshop on Advanced Learning Technologies (IWALT-00)* (pp. 247–248).
- Giraud-Carrier, C., Vilalta, R., & Brazdil, P. (2004). Introduction to the special issue on meta-learning. *Machine Learning*, 54, 187–193.
- Gordon, A., & Hobbs, J. (2003). Coverage and competency in formal theories: A commonsense theory of memory. In *Proceedings of AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*.
- Gordon, A., Hobbs, J., & Cox, M. (2008). Anthropomorphic self-models for metareasoning agents. In *AAAI-08 Workshop: Metareasoning: Thinking About Thinking*.

- Greve, K., Stickley, T., Love, J., Bianchini, K., & Stanford, M. (2005). Latent structure of the Wisconsin Card Sorting Test: A confirmatory factor analytic study. *Archives of Clinical Neuropsychology*, 20, 355–364.
- Hu, X., & Edwards, D. (2006). Context-dependent structure control for adaptive behavior selection. In *Proceedings of the Workshop on Bio-inspired Cooperative and Adaptive Behaviors in Robots, 9th International Conference on the Simulation of Adaptive Behavior, (SAB-06)*.
- Iizuka, H., & Paolo, E. A. D. (2007). Toward spinozist robotics: exploring the minimal dynamics of behavioral preference. *Adaptive Behavior*, 15, 359–376.
- Ikegami, T., & Iizuka, H. (2003). Joint attention and dynamics repertoire in coupled dynamical recognizers. In *Proceedings of the 2nd International Symposium on Imitation in Animals and Artifacts (AISB-03)*. (pp. 125–130).
- 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.
- Kandel, E. R., Schwartz, J., & Jessell, T. M. (2000). *Principles of neural science*. McGraw Hill.
- Kaplan, G., Sengör, N., Gürvit, H., Genç, I., & Güzelis, C. (2006). A composite neural network model for perseveration and distractibility in the wisconsin card sorting test. *Neural Networks*, 19, 375–387.
- Kelso, J. (1995). *Dynamic patterns: The self-organization of brain and behavior*. Cambridge, MA: MIT Press.
- Klinkenberg, R. (2005). Meta-learning, model selection, and example selection in machine learning domains with concept drift. In *Proceedings of Lernen, Wissensentdeckung und Adaptivität (LWA-05)* (pp. 164–171).
- Maniadakis, M., & Tani, J. (2008). Dynamical systems account for meta-level cognition. In *Proceedings of the 10th International Conference on the Simulation of Adaptive Behavior (SAB-08)*.
- Maniadakis, M., & Trahanias, P. (2006). Hierarchical cooperative coevolution facilitates the redesign of agent-based systems. In *Proceedings of the 9th International Conference on the Simulation of Adaptive Behavior, (SAB-06)* (pp. 582–593).
- Mansouri, F., & Tanaka, K. (2003). Wisconsin Card Sorting Test with macaque monkeys. *International Congress Series*, 1250, 105–118.
- Meeden, L. A. (1996). An incremental approach to developing intelligent neural network controllers for robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 26, 474–485.
- Milner, B. (1963). Effect of different brain lesions on card sorting. *Archives of Neurology*, 9, 90–100.
- Nelson, T., & Narens, L. (1990). Metamemory: A theoretical framework and new findings. In *The Psychology of Learning and Motivation*, 125–173.
- Nishimoto, R., & Tani, J. (2004). Learning to generate combinatorial action sequences utilizing the initial sensitivity of deterministic dynamical systems. *Neural Networks*, 17, 925–933.
- Nolfi, S., & Floreano, D. (2000). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. Cambridge, MA: MIT Press/Bradford Books.
- Paine, R., & Tani, J. (2005). How hierarchical control self-organizes in artificial adaptive systems. *Adaptive Behavior*, 13, 211–225.
- Proust, J. (2003). Does metacognition necessarily involve metarepresentation? *Behavior and Brain Sciences*, 26, 352–355.
- Ramesh, B. (1997). Towards a meta-model for representing organizational memory. In *Proceedings of the 30th Hawaii International Conference on System Sciences (HICSS-97)* (pp. 320–329).
- Reder, L., & Schunn, C. (1996). Metacognition does not imply awareness: strategy choice is governed by implicit learning and memory. In L. Reder (Ed.), *Implicit Memory and Metacognition* (pp. 45–78). Mahwah, NJ: Erlbaum.
- Russell, S. (1991). Principles of metareasoning. *Artificial Intelligence*, 49, 361–395.
- So, R., & Sonenberg, L. (2004). Situation awareness in intelligent agents: Foundations for a theory of proactive agent behavior. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)*.
- Stemme, A., Deco, G., & Busch, A. (2007). The neuronal dynamics underlying cognitive flexibility in set shifting tasks. *Journal of Computational Neuroscience*, 23, 313–331.
- Sun, R., Zhang, X., & Mathews, R. (2006). Modeling metacognition in a cognitive architecture. *Cognitive Systems Research*, 7, 327–338.
- Swanson, H. L. (1990). Influence of metacognitive knowledge and aptitude on problem solving. *Journal of Educational Psychology*, 82, 306–314.
- Tani, J., & Ito, M. (2003). Self-organization of behavioral primitives as multiple attractor dynamics: a robot experiment. *IEEE Transactions on Systems, Man, and Cybernetics Part A*, 33, 481–488.
- Tuci, E., Quinn, M., & Harvey, I. (2002). An evolutionary approach to the study of learning behavior using a robot-based model. *Adaptive Behavior*, 10, 201–221.
- Tuci, E., Trianni, V., & Dorigo, M. (2004). Feeling of time through sensory-motor coordination. *Connection Science Journal, Special Issue on Evolutionary Robotics*, 16, 301–324.
- Van Gelder, T. (1998). The dynamical hypothesis is cognitive science. *Behavioral and Brain Sciences*, 21, 615–628.

- Varela, F. J. (1979). *Principles of biological autonomy*. McGraw-Hill/Appleton & Lange.
- Weischedel, R. M., & Sondheimer, N. K. (1983). Meta-rules as a basis for processing ill-formed input. *American Journal of Computational Linguistics*, 9, 161–177.
- Yamauchi, B. M., & Beer, R. D. (1996). Spatial learning for navigation in dynamic environment. *IEEE Transactions on Systems, Man, and Cybernetics Part A*, 26.
- Yzerbyt, V., Lories, G., & Dardenne, B. (1998). *Metacognition: Cognitive and social dimensions*. London: Sage Publications Ltd.
- Ziemke, T. (1996). Towards adaptive behaviour system integration using connectionist infinite state automata. In Maes et al. (Eds.) *From Animals to Animats 4* (pp. 145–154). Cambridge, MA: MIT Press.
- Ziemke, T. (1999). Remembering how to behave: Recurrent neural networks for adaptive robot behavior. In I. Medsker & I. Jain (Eds.), *Recurrent neural networks: Design and applications* (pp. 359–389).
- Ziemke, T., & Thieme, M. (2002). Neuromodulation of reactive sensorimotor mappings as short-term memory mechanism in delayed response tasks. *Adaptive Behavior*, 10, 185–199.

About the Authors



Dr Michail Maniadakis received a B.Sc. in informatics (1996) from the University of Piraeus, an M.Sc. in systems automation (1999) from the Technical University of Crete and a Ph.D. in cognitive robotics (2006) from the University of Crete. His research interests are mainly focused on the higher level cognitive capacity of robots and the study of emergent phenomena in brain-like artificial systems.



Jun Tani received his B.S. degree in mechanical engineering from Waseda University, his dual M.S. degree in electrical engineering and mechanical engineering from the University of Michigan and his D.Eng. from Sophia University. He started his research career in Sony Computer Science Laboratory in 1990. He has been a team leader in the Laboratory for Behavior and Dynamic Cognition, Brain Science Institute, RIKEN, Tokyo since 2000 where he is currently leading seven post-doctoral researchers. He was also a visiting associate professor at the University of Tokyo from 1997 to 2002. He has studied the problems of robot learning with theoretical modeling of complex adaptive systems and neuronal networks for more than 15 years. He has also been interested in phenomenological problems of “self-consciousness” and his studies have addressed those problems from the view of embodied cognition. Five years ago, he started neuroscience studies on behavior learning processes in real brains utilizing both a scheme of human brain imaging and animal electrophysiology. His vision is to establish “brain-inspired robotics” by integrating these approaches. *Address:* Laboratory for Behavior and Dynamic Cognition, Brain Science Institute (BSI), RIKEN, Japan, *E-mail:* tani@brain.riken.jp