Subsystem Formation Driven by Double Contingency

Bernd Porr • University of Glasgow, UK • bernd.porr/at/glasgow.ac.uk
Paolo Di Prodi • University of Glasgow, UK • robomotic/at/gmail.com

> Purpose • This article investigates the emergence of subsystems in societies as a solution to the double contingency problem. **> Context** • There are two underlying paradigms: one is radical constructivism in the sense that perturbations are at the centre of the self-organising processes; the other is Luhmann's double contingency problem, where agents learn anticipations from each other. **> Approach** • Central to our investigation is a computer simulation where we place agents into an arena. These agents can learn to (a) collect food and/or (b) steal food from other agents. In order to analyse subsystem formation, we investigate whether agents use both behaviours or just one of these, which is equivalent to determining the number of self-referential loops. This is detected with a novel measure that we call "prediction utilisation." **> Results** • During the simulation, symmetry breaking is observed. The system of agents divides itself up into two subsystems: one where agents just collect food and another one where agents just steal food from other agents. The ratio between these two populations is determined by the amount of food available.

> Key words • **Social systems, constructivist paradigm, cybernetics, double contingency, symmetry breaking, emergence.**

1 Introduction

« 1 » Central to constructivist theory is the formation of subsystems, ranging from subsystems in organisms (Maturana & Varela 1980) to subsystems in society (Luhmann 1984: 80). In this paper, we will demonstrate subsystem formation on the level of neuronal systems operating on interacting agents. We then present a novel measure that is able to identify and quantify this subsystem formation.

« 2 » We will now introduce, step-by-step, our understanding of how subsystem formation occurs. We will then use it to set up a computer simulation with multiple agents.

2 Towards subsystem formation

« 3 » We first need to look at a single agent and how it establishes its self-referential operation and how it is able to learn. Figure 1a shows the essential building blocks of an agent in its environment, where the agent creates a closed loop that includes its sensors, its motor output and the environment itself. From a mathematical point of view, this is a cybernetic control system (Ashby 1956; Porr & Wörgötter 2003; Porr, von Ferber & Wörgötter 2003) that simulates the self-referentiality of the nervous system (Foerster 2003), where its elements, in the sense of Luhmann (1984: 41), are electrical signals that reproduce themselves in a self-referential manner. This system exists because there are unpredictable events in the environment that we call "perturbations," in the spirit of Humberto Maturana and Francisco Varela (1980). These perturbations act on the loop, which has the task of restoring its desired state or homoeostasis. Note that these loops exist because there are perturbations in the environment and these loops need to react to them.

« 4 » In accordance with the constructivist paradigm, the loop makes it impossible for the organism to distinguish between organism (h_{reflex}) and environment (e_{reflex}) because these transfer functions can easily be turned into different transfer functions without changing the dynamics of the loop at all. In the extreme case, the environment e_{reflex} can turn into the identity. The transfer function $H_{reflex,\,new}$ of the organism then absorbs both the environment and the reflex ($H_{reflex,\,new} = H_{reflex} / E_{reflex}$; $E_{reflex,\,new} = 1$). This means that the environment has been absorbed into the organism. The world is in our heads or, in other words: nervous signals create nervous signals and so on (Foerster 2003; Gadenne 2010), which are the elements in this simulation (Luhmann 1984: 41).

« 5 » However, the perturbation cannot be absorbed into the loop. It will always stay outside of the loop and will be perceived by the agent as the actual environment (Porr & Wörgötter 2005). The loop has to be adjusted in such a way that it acts appropriately against the perturbation p. This leads to one of the central aspects of constructivist theory: the loop has be a construction of the perturbation so that it is able to eliminate it successfully. The organism constructs its own world by its loops, which act against perturbations. This could be a threat but also something pleasurable, for example finding food. The appropriate action in this case would be to eat the food to go back to the "normal" state, namely no food. Instead, Luhmann calls perturbations "irritations" originating from the environment, which stresses more the fact that the closed loop needs to react to them (Luhmann 1996). Recall that for the organism a certain type of food only exists if it can sense a perturbation and then react to it appropriately. Otherwise, it will be not part of its construction of the world. This will become important later when we are talking about subsystem formation.

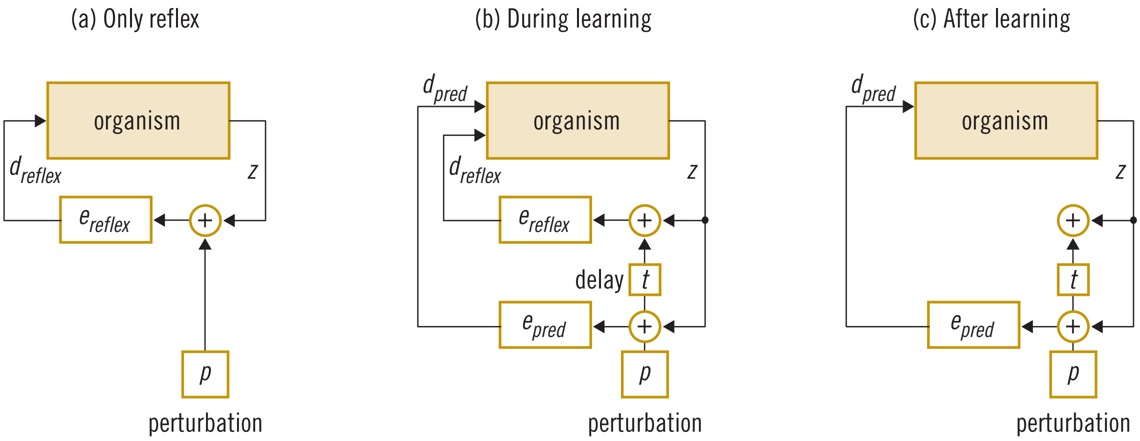

Figure 1: (a) The organism is connected to the environment via the motor output z and the reflex sensory input d_{reflex}. The environment introduces a perturbation p via the transfer function e_{reflex}, which in turns changes the reflex d_{reflex}. The organism wants to keep the reflex at its desired state. **(b)** An organism can learn to keep its desired state by using a predictive input d_{pred}, providing that the perturbation p acts on the reflex d_{reflex} with a delay of t. **(c)** After learning, the organism should have reduced the reflex to zero by using the predictive information d_{pred}.

« 6 » So far we have only discussed loops that are fixed. Such loops might have been established during evolution or in our simulation designed by the experimenter so that the agents can act in their environment. The other option is that loops are created or removed while the agent is acting in its environment by using a closed-loop learning algorithm (Porr & Wörgötter 2003; Porr, von Ferber & Wörgötter 2003). Figure 1b shows a typical scenario, where we have one loop via d_{reflex} and one loop via d_{pred}. In this example, the loop via d_{reflex} is superseded by the loop via d_{pred}. This example focuses on predictive learning, where the loop via d_{pred} can react faster against the perturbation than the loop via d_{reflex} because there is a delay between the outer loop and the inner loop. For example, a dangerous situation can be avoided by predicting it or food can be targeted and not just encountered by accident. Consequently, learning should identify the faster-reacting loop and then replace the loop via d_{reflex} with a loop via d_{pred}. In Figure 1c, this has been perfectly accomplished. The loop via d_{reflex} is no longer needed and the loop via d_{pred} has taken over. Coming back to our food example: the loop via d_{reflex} represents an agent that just bumps into food accidentally and then eats it; the loop via d_{pred} represents vision so that an agent can see food from the distance and then plan to get to the food. Equivalently, an agent can learn to anticipate adverse events and avoid them before getting into imminent danger.

« 7 » Coming back to the constructivist framework, this means that the agent now constructs the perturbation via a different loop and therefore has a different construction of its perturbation in its environment. In other words, the experience of a perturbation is turned into an explanation of itself (Kenny 2009). This argument can also be turned around: perturbations have the potential to create loops and thus create constructs about the perturbations. As shown before, organisms can construct more and more of their environment by creating more and more loops. On the other hand, organisms can also choose to remove loops so that they are no longer aware of this aspect of the environment (Luhmann 1984: 250). This will become important for subsystem formation.

« 8 » Agents may not be alone in their environment. As soon as there is more than one agent, the agents will perturb each other. If learning is possible, agents will try to learn from each other and generate more loops that include other agents. However, every new loop an agent creates causes another perturbation for another agent, which in turn needs to be learned. This leads to the double contingency problem, which states that reciprocally-anticipating agents create even more unpredictability (Parsons 1968: 436). Having many agents will make it impossible to generate reliable predictions and therefore stable loops.

« 9 » A solution to the double contingency problem is that agent-agent interaction does not aim to predict every aspect of the other agent but concentrates only on a few aspects instead. Having a group of agents then makes it easier to generate predictable behaviour because in the simplest case only one loop/perturbation is used to change an agent's behaviour or to react to it. Luhmann (1984: 177) then proposed that this subsystem formation emerges in a self-organised manner: perhaps electricians talk only about light bulbs and bakers only about bread making.

« 10 » Figure 2 shows a formalisation of two agents interacting with each other. Let us first concentrate on the organism on the left. We will find again the aspects shown in Figure 1, consisting of an inner fixed reflex loop (now using dotted lines) and an outer predictive loop that can be switched on or off via learning. Let us now consider the special case where the perturbation p is generated by the other agent and vice versa. While in Figure 1 the perturbation was from an unpredictable origin, in Figure 2 it comes from the observable behaviour (utterances/actions) of the other agent and vice versa. This means that Agent 1 will be perturbed by Agent 2 and Agent 2 will be perturbed by Agent 1. Remember that both agents use learning to supersede their reflex loops with faster anticipatory loops. In this case, this means that the agents generate expectations of expectations and change their behaviour in a recursive fashion (Luhmann 1984: 420). This is a version of the "double contingency problem" because the agents perturb each other and at the same time adjust their behaviour continuously (Parsons 1968: 436).

« 11 » The two-agent system is still too simple to arrive at subsystem formation. We require agents that incorporate more than one perturbation and therefore have the opportunity to generate and/or remove loops.

« 12 » An agent-agent system with two agents having two feedback loops and corresponding predictive loops is shown in Figure 3. Again, the reflex loops are shown as dotted lines and the predictive loops as solid ones. The adaptive controllers have been collapsed into black boxes called "C." The interesting aspect is now the interaction between the two agents. Because we have two loops, we can have cross interactions between them. For example, loop 1,2 can perturb loops 2,2 and 2,1 of the other agent. This will inevitably lead to more uncertainty because there are two different learners in every agent that will use any information from the other agent to predict its behaviour. Overall, the double contingency problem will be much more difficult to solve because the behaviours of the agents become more complex and will be more difficult to predict by the other agent. A way of reducing the complexity of the interaction is to stop using all possible loops and, thus, shut down certain sensor inputs that represent certain aspects of the environment. This is the first step towards subsystem formation, namely removing certain loops to reduce complexity.

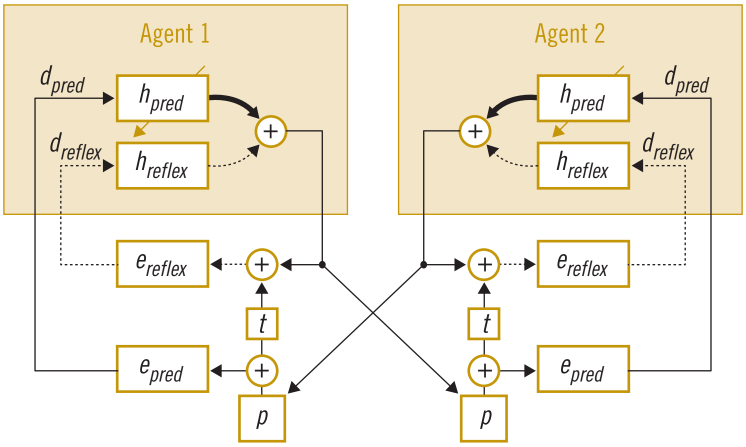

Figure 2: Double contingency formalised. h_{pred} and h_{reflex} represent the agents' controllers for anticipatory and reflex behaviour respectively.

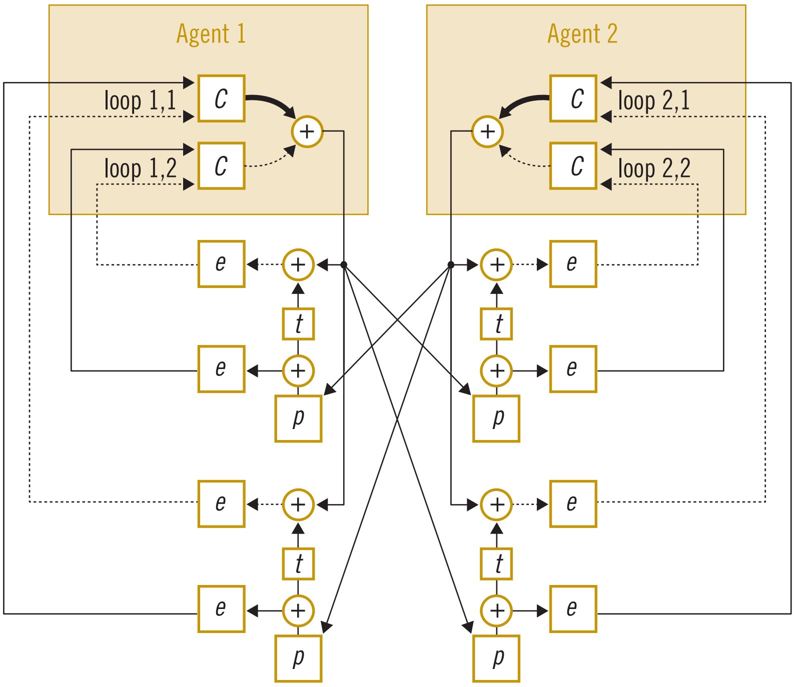

Figure 3: Agent-agent interaction with four loops: two for each agent, which allows subsystem formation. Here, *C* is the learning controller, processing the reflex and predictive signals as before.

« 13 » The double contingency will become even more challenging to solve when there are more than two agents. In this case, every agent tries to predict the behaviour of many other agents to adjust its own behaviour and vice versa. In addition, different stimuli can cause a change in behaviour and learning. The result is that agents will basically just perceive noise, which will make it impossible to anticipate other agents' actions. An agent will have no other choice than to resort to its most inefficient loops and be reactive. In order to solve this problem of having to calculate over-complex predictions, Luhmann proposed that every system forms subsystems, where agents only concentrate on one aspect of their environment and ignore all others (Luhmann 1984: 250).

« 14 » As said earlier, agents can choose to remove loops, which in turn removes the construct of the perturbation for the agent. So, by removing a loop, an agent no longer reacts to certain perturbations in the environment and therefore this part of the environment no longer exists for the agent.

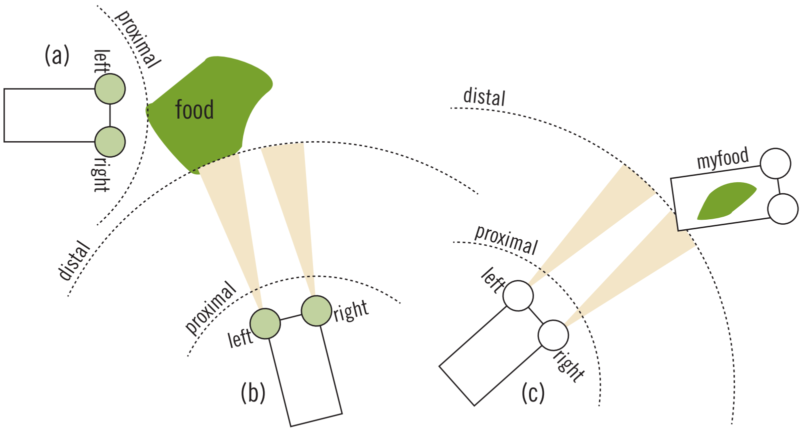

Figure 4: The robot scenario. (a) The proximal food sensors of the robot are triggered. (b) The distal food sensors of the robot are triggered. (c) Distal steal-food sensors of the robot are triggered.

Remember that agents learn to anticipate perturbations. However, they can also remove certain loops to no longer anticipate a perturbation. Luhmann calls an agent's decision not to use certain aspects of the environment a "selection" (Luhmann 1984: 384). In the next section, we will use as an example a system of agents where every agent has to collect food and eat it. The agents can either look for food or steal it from other agents. All agents have the ability to learn both finding food and stealing it. However, we will show that two subsystems emerge, consisting of agents that only steal and agents that only collect food.

3 The social system model

« 15 » The simulation model is a two dimensional arena bounded by walls. It contains agents and food sources. The arena contains *N* agents and *M* food disks, which can be collected by the agents and then are slowly eaten up. Once a food disk has been collected, a new one will appear at another random location. Agents can steal the food from each other.

« 16 » Agents have different sensors that enable them to sense food disks, food on other agents and obstacles. This is shown in Figure 4 for the food attraction. For each of these qualities, there exist two kinds of sensors:

- Proximal sensors, which make the agent react at a short distance and can be seen as reflexes that are hard wired and cannot be changed. An example is shown in Figure 4a, where the robot is sensing the food disk from a short distance and then collects the food. All of these loops are designed in such a way that they have the requisite variety (Ashby 1956; Di Paolo 2005) to deal with the specific perturbations, namely accidentally finding food and bumping into other agents or into walls. These loops guarantee the self-referential operation of the agent. Either way, the trigger of these loops with perturbations is not desirable from the agent's point of view because they create a deviation from the agent's desired state (see Figure 1a). On a more positive note, the trigger of these loops can be seen as irritations that trigger learning in the agent. Learning is achieved by modifying the loops, which in our case is achieved by introducing distal sensors and their loops.
- Distal sensors, which provide information from a distance and allow the agents to generate anticipatory actions. These actions need to be learned by the agents and can also be removed again during learning. These are the actions that are crucial for subsystem formation. Two cases are shown in Figure 4b and c. In 4b the robot has learned to use its distal sensors to approach the food disk from a distance. This corresponds to Figure 1b and c, where the agent is able to reduce its requisite variety by rendering the loop via the proximal sensor obsolete. Being able to learn such predictions is highly desirable for self-referential systems and is featured in virtually any system, ranging from cells to communication systems (Luhmann 1984: 420).

« 17 » The distal/proximal signals for the sensors are generated by simple potential fields, which decay at a rate of r^2 from their origin. For the proximal sensor, the potential field decays to zero within a few coordinated steps between the agent and the signal source (food or obstacles), whereas the distal field is adjusted in such a way that it spans the whole arena.

« 18 » Agents move with a differential drive system (Braitenberg 1984), which allows error-driven navigation. This is achieved by combining sensor signals to error signals in such a way that the agent is either attracted to objects or avoids them. Here, the error signals are generated by subtracting the right sensor signal from the left sensor signal, which is used to generate a steering angle and to control learning. For example, in Figure 4b this results in a positive error signal (the left sensor signal is stronger than the right sensor signal), whereas in Figure 4c this results in a negative error signal (right is stronger than left). This setup guarantees that the agent always interacts with its environment and that processing is ongoing all the time (Froese & Ziemke 2009). This is in accordance with a good/bad normativity that avoids explicit goals that would eventually be reached (Georgeon, Marshall & Manzotti 2013). Thus, the agent continuously performs self-referential actions that never stop, but at the same time the agent experiences good or bad moments conveyed by its error signals.

« 19 » Learning is performed by the Input Correlation learning (ICO) rule (Porr & Wörgötter 2006), which belongs to the class of differential Hebbian learning rules and which essentially makes the agent learn to replace late reflexes with anticipatory actions. Here, ICO learning replaces the late reflex generated by the proximal sensors with an anticipatory action generated by the distal sensors. Learning is driven by the correlation of the proximal sensors with the distal sensors. After learning, the reflexes should no longer be in use and should be replaced with anticipatory actions. ICO learning was chosen because it only correlates the agents' sensor inputs with each other but does not use the motor output. From a constructivist perspective, this is desirable because firstly, the agent can only observe its inputs and secondly, learning is triggered by error signals at the inputs of the agent but not at its outputs (Froese & Ziemke 2009).

« 20 » We will now explain the different behaviours.

3.1 The behaviours

« 21 » Figure 5 shows the behaviour-generating circuit. We have robots with two wheels that drive straight ahead as long as both motors receive the same voltage, and turn if the voltages are different. Three different behaviours based on the following sensors are generated:

- Food sensors detecting food in the arena;
- Food sensors detecting food on other robots;
- Obstacle sensors that generate avoidance behaviour in robots.

« 22 » The agents' behaviours are now described one by one. The mathematical formalism is presented in the appendix.

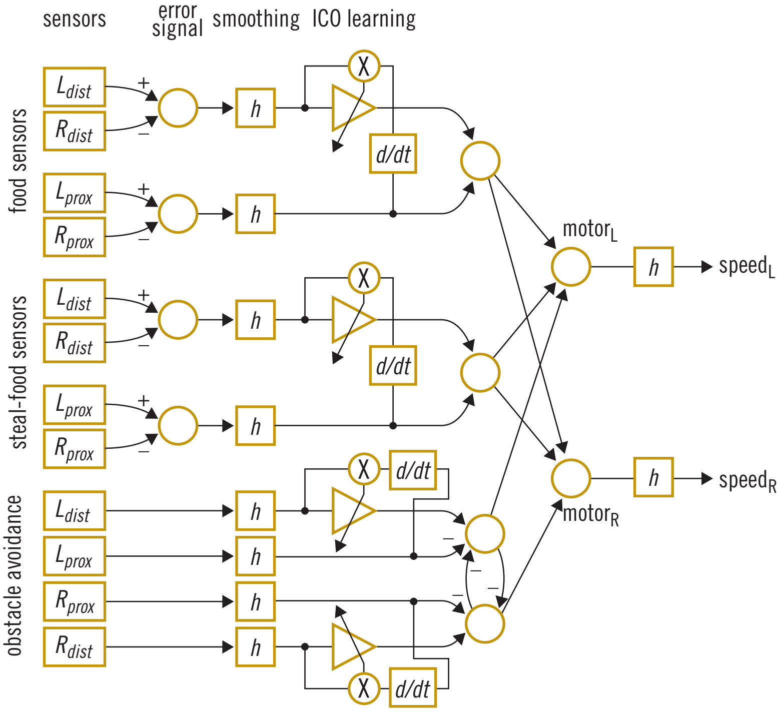

Figure 5: The data flow diagram of the robot controller.

3.2 Food attraction

« 23 » The first behaviour we describe is food attraction, where the agent learns to find food from a distance and collect it. Before learning, the agent has only its inert reactive behaviour, which steers the agent towards food that is close by, which relates to Figure 1a and section 2. This is achieved by subtracting the right proximal sensor signal from the left proximal sensor signal and using this as an error signal. This makes the agent drive towards the food that is at close range. The encounter with the food is the perturbation here because this action always happens too late. In order to eliminate this non-optimal reaction via the proximal sensor, we use the distal sensor, which allows anticipatory behaviour (Luhmann 1984: 420). In order to learn to steer to a food disk from a distance, the robot utilises the signals from the distal sensors in similar manner than before. Again, we generate an error signal based on the difference between the left and right distal sensor. However, in contrast to the reflex behaviour, the signal from the distal sensor is weighted by the factor $\omega_{food, pred}(t)$, which is initially set to zero so that the agent only uses its proximal reflex. Learning then enables the robot to use its proximal sensors also, by increasing the weight $\omega_{food, pred}$ if first the distal and then a little later the proximal sensor has been triggered. In other words, ICO learning tries to pre-empt the trigger or "irritation" of the late reflex with the help of the distal sensor. Learning continues until the agent's late proximal sensors are no longer needed and steering is achieved by the predictive distal signals. This corresponds to Figure 1c where ideally only the predictors are being used, which Luhmann calls "Erwartungsstrukturen" (structures of expectations) (Luhmann 1984: 404). In more complex scenarios, predictors of predictors or a combination of predictors can also be learned.

« 24 » After learning, the agents should be able to drive towards food from a distance by using their distal sensors. The higher the weights $\omega_{food, pred}$, the stronger is the attraction to food seen from a distance. In the same way, learning takes place for the other behaviours, in particular the stealing of food.

3.2.1 Carrying food

« 25 » Having introduced the way agents learn to find food, we can now introduce how agents carry food and then how other agents can steal it from them.

« 26 » Agents have a food reservoir called "myfood," which they slowly eat up and which eventually decays to zero. When the agent encounters food, its own food state is restored back to its maximum value (*myfood* = 1). Every agent broadcasts its *myfood*-state to all the other agents and this can be detected from a distance in the form of a decaying field, in the same way as the food disks.

3.2.2 Stealing food from other agents

« 27 » When an agent bumps into another agent and the other agent has food (*myfood* > 0), this causes a reflex reaction and the agent "steals" the food from the other agent. This irritation is then used for ICO learning to generate an anticipatory action by sensing the *myfood* state of other agents from a distance. As stated before, the agents broadcast their *myfood* state to the other agents and excite the sensors that detect food from a greater distance. Again, we have a weight $\omega_{stealfood,\,pred}$, which controls whether the distal food sensors are allowed to steer the agent towards food. This weight is changed by the ICO learning rule by correlating the food-related reflex with the sensor that detects the information broadcast about the other agents' food reservoir (*myfood*). After learning, an agent is able to drive towards other agents that carry food and steal the food from them. The higher the weight $\omega_{stealfood,\,pred}$, the stronger the attraction towards other agents carrying food.

« 28 » In addition, obstacle avoidance is implemented but is not used in the following analysis. It has only been introduced to generate a more natural behaviour, preventing robots from crashing into each other and into walls (see the appendix for a detailed description). The behaviours towards food and other robots to steal food are crucial for our subsystem model. We will show that robots either learn to collect food ("seekers") or to steal it from other robots ("parasites"), which means that they are forming subsystems. In order to identify the subsystems, we use a new measure that we call "prediction utilisation," which will now be introduced.

4 Prediction utilisation

« 29 » As described in the introduction, subsystem formation is achieved by removing certain sensor/action loops so that this aspect of the environment no longer exists for the agent. On the other hand, learned behaviour is about anticipation: namely, anticipating what the other agent is going to do, or more generally, what is going to happen in the agents' (subjective) environment. Thus, we introduce a measure called "predictive utilisation," which reflects how much an agent uses a certain loop to anticipate perturbations in its environment.

« 30 » In previous works, we have used the weights connecting the sensor inputs to the agents' motor outputs for this purpose (Porr, Egerton & Wörgötter 2006). Here, we treat the agent as a black box and just observe which sensor events correlate to motor events and which create successful loops.

« 31 » The formal framework here is information theory, where we look at the information in the loops and whether desired states have been maintained. Using an information theoretical framework makes it possible to generalise to other systems, such as the communication system. We are going to use two information measures and combine them into a single measure, "prediction utilisation," which indicates whether anticipatory behaviour has been achieved and to what level.

4.1 Symbols and conventions

« 32 » The symbols used in this section follow the following conventions:

- Capital letters, such as X, indicate a random discrete variable.
- The symbols of the random variable are indicated by the set $X = \{x_1, x_2, \ldots, x_S\}$, where S is the number of symbols in this set.
- Non-capital letters, such as x, indicate the corresponding discrete time series $x(k)$ from which we estimate the density $P(X)$ of the corresponding random variable X.
- The estimated entropy of a random discrete variable X is identified by $H(X)$ and is measured in bits.

« 33 » We then identify our in-/outputs using the following variables:

D_{reflex} a random variable for reflex input d_{reflex}. In our simulation we have two different variables: one for the food disks and one to steal food (e.g., $d_{food/stealfood,\,reflex}$). If we refer to d_{reflex} we mean either of them.

D_{pred} a random variable for predictive input d_{pred}.

Z a random variable for steering angle z or an error signal generated by the ICO neuron.

« 34 » We first start with measures that quantify the information at the input of the agent and then measure the information in the closed loop that the agent and the environment comprise. Finally, we combine them to form our new measure, "prediction utilisation."

4.2 Input reflex entropy

« 35 » We first look at the entropy $H(D_{reflex})$, which is the uncertainty at the reflex input d_{reflex}, which in turn is part of a reflex loop (e.g., $d_{food/stealfood/avoid,\,reflex}$). Remember that $d_{food/stealfood,\,reflex}$ are essentially error signals recording changes from the desired state, which drives the reflex (see Figure 1), and that these error signals could be understood as an ongoing measure of the agent's good/bad state (Froese & Ziemke 2009) or interpreted as irritations (Luhmann 1996). Because these are reflex loops, there is always a lag such that the input d_{reflex} cannot be maintained in its desired state all the time. Thus, the entropy at d_{reflex} reflects the entropy originating from the perturbation p, which is the law of requisite variety (Ashby 1956). Let us assume that the reflex has an alphabet of three symbols such that $d_{reflex} = \{-1, 0, 1\}$, which encode an error to the left, no error or an error to the right, respectively. However, while an error of $d_{reflex} = 0$ is desirable, the condition $H(D_{reflex}) = 0$ does not imply that the error is zero:

$$d_{reflex}(t) \quad = \{1,1,1,1\} \quad \rightarrow H(D_{reflex}) = 0 \tag{1}$$

$$d_{reflex}(t) \quad = \{0,0,0,0\} \quad \rightarrow H(D_{reflex}) = 0 \tag{2}$$

$$d_{reflex}(t) = \{-1,-1,-1,-1\} \rightarrow H(D_{reflex}) = 0 \tag{3}$$

« 36 » We can exclude conditions Eq. 1 and Eq. 3 because they only arise when the feedback loop has failed completely. In order to have successful learning, we require at least a working feedback loop (Froese & Ziemke 2009; Porr & Wörgötter 2005), where its values will be non-zero before learning and converge to zero after successful learning. This non-zero entropy before learning at d_{reflex} reflects the perturbation injected into the system via p (see Figure 1). It reaches a maximum of 1.6 bits in our simple case, illustrated with the following two time series:

$$d_{reflex}(t) = \{-1,0,1,0,-1,1\} \rightarrow H(D_{reflex}) = 1.6 \tag{4}$$

$$d_{reflex}(t) = \{-1,-1,0,0,1,1\} \rightarrow H(D_{reflex}) = 1.6 \tag{5}$$

where in these cases the input has a uniform distribution.

« 37 » Without learning, the reflex input entropy $H(D_{reflex})$ will reflect merely the entropy of the perturbation. However, after learning, the error signal d_{reflex} should be zero all the time because the agent is using its predictive inputs d_{pred} instead to find food or to avoid obstacles. This means that before learning, the entropy at the reflex input should be in the order of the perturbation (law of requisite variety) and after learning the reflex input entropy $H(D_{reflex})$ should ideally be zero. This could be used to measure the transition from reflex-based reactions to proactive-based actions.

« 38 » However, the input entropy $H(D_{reflex})$ does not tell us anything about the effort that the controller is making to keep its desired goal ($d_{reflex}(t)=0$). This is because it might be that the agent is not moving at all, that the environment is very simple or that the robot is spinning around its own axis. For that reason, we now need to define a measure that looks at the correlations between two different points in the loop: is the robot able to keep its own sensor inputs predictable for its own actions?

4.3 Mutual information

« 39 » So far, we have been dealing with pure input measures. In this section we introduce measures that are calculated between the inputs and outputs of the agent and, thus, measure whether a sensor input actually causes a motor reaction in a predictive way (from the agent's point of view). More specifically, we look at the mutual information that can be used as a performance measure of how the agent reacts to the signals at the reflex and predictive inputs.

« 40 » The mutual information can be formulated with the help of conditional entropies $H(D_{reflex}|Z)$ and $H(D_{pred}|Z)$:

$$MI(Z, D_{reflex}) = H(D_{reflex}) - H(D_{reflex}|Z) \quad (6)$$

$$MI(Z, D_{pred}) = H(D_{pred}) - H(D_{pred}|Z) \quad (7)$$

« 41 » The conditional entropy $H(D_{reflex}|Z)$ is a measure of how much the motor output Z has no influence over the reflex input D_{reflex}. This is the amount of uncertainty remaining in the reflex input D_{reflex} after a motor action Z has been chosen. The right hand side of Eq. 6 can be read as the amount of uncertainty in the reflex input D_{reflex}, minus the amount of uncertainty in D_{reflex} after Z has been chosen. From the robot's point of view, the mutual information $MI(Z, D_{reflex})$ measures the quantity of information that the robot is able to recover from its inputs, given its outputs. For example, in a company, that could be the publishing of an advertisement that then has a certain effect on sales figures. If the advert had no effect on the sale figures, the mutual information would be zero.

« 42 » Here, we use the following mutual information measures:

1 | $MI(Z, D_{reflex}, \tau, n)$: mutual information of the reflex loop

2 | $MI(Z, D_{pred}, \tau, n)$: mutual information of the predictive loop

where the parameter τ is the temporal difference between the motor output z and the sensory input d_{reflex}, d_{pred} and n is the sensory integration window in time steps. Specifically, when computing $MI(Z, D_{reflex}, \tau, n)$, we are considering the motor output $z(k)$ and the sensory input sequence $d_{reflex}(k + \tau)$, $d_{reflex}(k + \tau + 1)$, …, $d_{reflex}(k + \tau + n - 1)$.

« 43 » Omission of τ and n indicate that the mutual information has been maximised over those two parameters:

$$MI(Z, D_{reflex/pred}) = \max_{\tau,n} MI(Z, D_{reflex/pred}, \tau, n) \quad (8)$$

« 44 » The mutual information $MI(Z, D_{reflex})$, $MI(Z, D_{pred})$ is a measure of how much the agent can control its own sensor input. To demonstrate this, we show the two extreme cases:

1 | If $MI(Z, D_{reflex/pred})=0$ then Z and $D_{reflex/pred}$ are independent. It means that there is no correlation between the actions and the inputs of the robot. Imposing a motor value does not give a desired input. For example, the series:

$$z(t) = \{1, 2, 3, 4, 5, 6\} \quad (9)$$

$$d_{reflex/pred}(t) = \{-1, 0, -1, 1, -1, 0, 1\} \quad (10)$$

has zero mutual information $MI(Z, D_{reflex/pred}, \tau=1) = 0$ bits. Note that this is not permitted for the reflex ($MI(Z, D_{reflex}, \tau=1)=0$) because we assume that the reflex loop is working properly and will be able to keep to its desired state as closely as possible, given a certain perturbation.

2 | If $MI(Z, D_{reflex/pred}) = max(H(Z), H(D_{reflex/pred}))$ then Z and $D_{reflex/pred}$ are perfectly dependent. This means that this time, when the robot imposes a motor action, it will achieve the desired input. For example if $MI(Z, D_{reflex/pred}) = 1.6$ bits, then the robot can choose a motor action and read a desired input in an average run. But if the robot loses 0.6 bits and goes to $MI(Z, D_{reflex}) = 1.0$ bits, then in the average run one particular motor action will yield two equiprobable inputs at $D_{reflex/pred}$ and thus the robot will have less control over its environment.

« 45 » However, neither the mutual information via the reflex nor the predictive pathways themselves are sufficient to determine whether anticipatory learning has been successful. Remember that we would like to measure the success of learning by using the predictive pathway via d_{pred} to eliminate the pathway via d_{reflex}. We need to check whether the mutual information has been transferred from the reflex to the predictive pathway and, thus, the error of the reflex has been reduced to $d_{reflex}=0$ after learning.

4.4 Measuring subsystem formation: Prediction utilisation

« 46 » The "prediction utilisation" (PU) measure is computed with the help of the information measures introduced in the previous sections, before learning ($t=0$) and after learning ($t=\infty$). In practice we evaluate the values after learning, once the weights have been stabilised.

« 47 » The prediction utilisation is then computed with help of the input entropy and the mutual information as:

$$PU = \frac{H(D_{reflex})_{t=0} - H(D_{reflex})_{t=\infty}}{H(D_{reflex})_{t=0}} \cdot \frac{MI(Z, D_{pred})_{t=\infty}}{MI(Z, D_{reflex})_{t=0}} \quad (11)$$

« 48 » The first factor of Eq. 11 provides a measure reflecting the reduction of the entropy of the error signal D_{reflex}, which drives the reflex. Remember that the goal of learning is to avoid the reflex, which in an ideal case will lead to no trigger of the reflex or $d_{reflex}=0$. In a realistic scenario, the reflex entropy will decrease but will never reach zero because the agent will make mistakes from time to time. After successful learning, the entropy should be lower than before: $H(D_{reflex})_{t=0} \geq H(D_{reflex})_{t=\infty}$. Thus, the first factor in Eq. 11 will be 1 for perfect avoidance of the reflex (best performance) and 0 for no change in the reflex entropy (worst performance).

« 49 » The second factor measures whether the agent controls its own actions before ($MI(Z, D_{reflex})_{t=0}$) and after ($MI(Z, D_{pred})_{t=\infty}$) learning. Remember that before learning, this is achieved via the reflex input, which is part of a working reflex loop. After learning, the agent should be still be in control of its own actions via the predictive inputs. Ideally, these two mutual information values should be similar, meaning that predictability before and after is guaranteed.

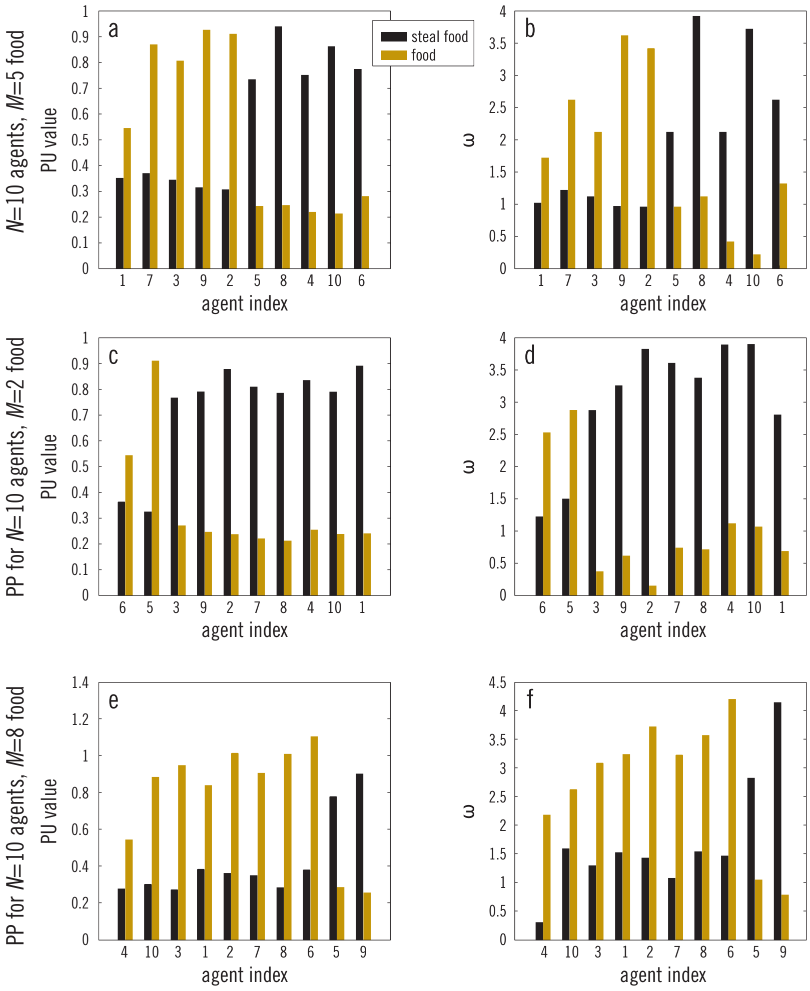

Figure 6: Predictive Utilisation computed in the social system for: (a/b) 10 agents and 5 food sources, (c/d) 10 agents and 2 food sources, (e/f) 10 agents and 8 food sources. The left column contains the PU measures, with white bars for the food attraction behaviour ("food") and black bars for the stealing behaviour ("steal food"). The right column shows the final weights after the system has stabilised for the same behaviours: food attraction and stealing food.

	$MI(Z, D_{reflex})$	$MI(Z, D_{pred})$	$H(D_{reflex})$
before learning	**2.5 bits**	1.5 bits	3 bits
after learning	0 bits	**2.5 bits**	**0 bits**

Table 1: Information values for a perfect learner.

« 50 » An agent with the values shown in Table 1 is an example of a perfect learner, which results in a prediction utilisation of $PU=1$ (see Eq. 11). The important values are in bold. The mutual information is completely transferred from the reflex pathway to the predictive pathway and the error in d_{reflex} is reduced to zero bits, which is reflected by $H(D_{reflex})_{t=\infty}=0$ and $MI(Z, D_{reflex})_{t=0} \cong MI(Z, D_{pred})_{t=\infty}$ because the predictor should be able to provide information that has to be learned or exploited by the controller.

« 51 » In the following section, we apply this measure to our social system in order to find out whether agents specialise and utilise only certain loops and ignore other ones. This will result in low PU values for some of the loops, which will indicate that this aspect is no longer part of the agent's perceptive world.

4.5 Results

« 52 » Having introduced the prediction utilisation, we can now use it to analyse subsystem formation.

« 53 » As described in the introduction, we are hoping to see a specialisation into subsystems. How is the prediction utilisation computed when there are different behaviours? Looking at Figure 5, we have three reflexes and three predictive behaviours that can be learned. Here, we are interested in collecting food and stealing food, which are two reflexes and two predictive behaviours (the first two ICO learners from the top). The prediction utilisation is computed for these two behaviours by comparing the signals of its reflexes, the predictive inputs and the motor output before and after learning. For example, prediction utilisation for finding food is calculated by taking the food eating reflex ($D_{food, reflex}$), the food sensor that picks up the food field ($D_{food, pred}$) and the agent's differential motor output. These three values are then fed into Eq. 11 before learning ($t=0$) and after the weights have stabilised ($t=\infty$).

« 54 » We have run 3 different scenarios:

1 | A group of $N=10$ agents and $M=5$ food sources.

2 | A group of $N=10$ agents and $M=2$ food sources.

3 | A group of $N=10$ agents and $M=8$ food sources.

This means that food is sparse in the second and abundant in the last scenario.

« 55 » Figure 6 shows the results of our simulations, where the left column shows the prediction utilisation for targeting food (white bars) and for targeting other agents to steal food (black bars). The right column shows the weight development of these two behaviours (see Eq. 24 for the weight for stealing food and Eq. 17 for the attraction towards food in the arena).

« 56 » First, we show how the ratio between agents (N) and food (M) determines the ratio between those agents that seek food and those who steal from other agents. We call the latter "parasites."

1 | A group of $N=10$ agents and a moderate amount of $M=5$ food sources results in five seekers and five parasites and thus a balanced number of seekers and parasites. The agents identified with 1, 7, 3, 9 and 2 have $PU_{food} > PU_{stealfood}$ and are therefore seekers. The agents identified with 5, 8, 4, 10 and 6 have $PU_{food} < PU_{stealfood}$ and are therefore parasites.

2 | The group of $N=10$ agents with only $M=2$ food sources generates two seekers and eight parasites. The agents identified with 6 and 5 have $PU_{food} > PU_{stealfood}$ and are therefore seekers. The agents identified with 3, 9, 2, 7, 8, 4, 10 and 1 have $PU_{food} < PU_{stealfood}$ and are therefore parasites. This distribution arises because initially all agents learn to collect food but this causes a rush of all agents towards food disks, which disrupts their proactive behaviour. As a consequence, predictive learning for collecting food is not successful for most agents.

3 | The group of $N=10$ agents with a large number of $M=8$ food sources generates eight seekers and two parasites. The agents identified with 5 and 9 have $PU_{food} < PU_{stealfood}$ and are therefore parasites. The agents identified with 4, 10, 3, 1, 2, 7, 8 and 6 have $PU_{food} > PU_{stealfood}$ and are therefore seekers. This case is interesting because it is not directly clear why agents prefer to collect food instead of stealing it from other agents. However, the reason why collecting food is the preferred proactive behaviour is that inanimate objects can be predicted easily whereas the interaction with other agents always causes a double contingency problem. The perceived uncertainty is much lower for stationary food disks than for other agents, which adapt their behaviour all the time.

« 57 » In summary, we have shown that the agent/food ratio determines the seeker/parasite ratio. In all three cases, agents chose the role that causes the lowest uncertainty.

« 58 » There is an important observation about the discrepancy between weight levels and prediction utilisation: high weights do not necessary mean high predictive utilisation. In other words, high weights do not guarantee that an agent is able to steer successfully towards a food disk or another agent. Higher weights might even render the agent's behaviour more unpredictable. This can especially be seen in the first row for $N=10$, $M=5$. For example, agent 7 has relatively low weights but its prediction utilisation is as high as that of agent 9. For example, a political party might react to certain perturbations from the environment (i.e., from journalists or other parties) very strongly, which corresponds to high weights in our scenario. However, the predictive utilisation might be small because a) the party might experience a massive backlash and b) it might lose the control it needs to counteract further against perturbations (Luhmann 1996).

« 59 » In summary, we have shown that two social subsystems emerge, namely agents that collect food (seekers) and other agents that steal food from other agents (parasites). In order to distinguish between seekers and parasites, we have developed a new measure, which we call "prediction utilisation." Subsystem organisation is an emergent property and is not programmed into the system, but agents self-organise themselves into one of these groups.

4.6 Discussion

« 60 » In this work we have shown that anticipatory learning under constraints leads to subsystem formation in a multi-agent system. In order to identify subsystems, we have used the measure prediction utilisation.

« 61 » Luhmann defines subsystem formation as system formation within a system (Luhmann 1984: 37). Here, subsystem formation is the active removal of anticipatory closed loops and their perturbations so that agents no longer deal with these uncertainties. Because we have a social system of anticipating agents, the main source of perturbations is the mutual agent-agent interaction that is called double contingency (Parsons 1968: 436). The formation of subsystems allows uncertainty to be reduced by experiencing double contingency only within a subsystem but not in the system as a whole, which is central to Luhmann's system theory (Luhmann 1984: 259).

« 62 » Multi-agent systems can be modelled on different levels of abstraction, ranging from cellular automata (Leydesdorff 2005) to point agents exchanging messages (Dittrich, Kron & Banzhaf 2003) to behaviour based systems (Heat, Hill & Ciarallo 2009). We have modelled our system with behaving agents that move in an environment. This allows the modelling of actual physical actions alongside utterance generation. While non-constructivist agent-based language research has focused more recently on the interaction of language and physical actions (Cangelosi et al. 2007), constructivist agent models usually focus purely on the communication system in the form of symbolic interactions between agents (Dittrich, Kron & Banzhaf 2003; Barber et al. 2006). In this paper we have attempted to combine an action system with a communication system. This is still in its infancy because the utterances are broadcasts and have no directed intentionality towards individual agents. On the other hand, our scenario demonstrates the underlying challenges of how these systems practically interpenetrate each other. The standard way of implementing social communication is anticipatory learning during mutual interactions (Dittrich, Kron & Banzhaf 2003; Barber et al. 2006). Instead, in our model the communication is one-to-many: this has an operative advantage in terms of fast response times if the system has to adapt to environmental changes. However, on the other hand an agent cannot directly communicate towards another agent, which would allow the agent to use anticipations especially learned about a specific agent. This is a subject for further research.

« 63 » Utterances can take different forms, for example, symbols or just analogue sounds. Symbolic interaction allows clear cut decisions, basically using a lookup table. Some even argue that symbols are essential for subsystem formation (Salgado & Gilbert 2008). However, we use analogue communication, where agents can gradually react to signal/utterances and integrate them into their anticipatory learning. For behaving agents this has the advantage that mistakes arise gradually and not at full force, which is a substantial advantage over symbolic interaction. On the other hand, the subsystem formation never generates a black and white situation where one agent is just a seeker or another just a parasite (Grant 2002).

« 64 » Our agents self-organise into two subsystems, namely seekers and parasites. This requires symmetry breaking, which arises from the interaction between positive and negative feedback (Luhmann 1984: 261; Leydesdorff 2005). The positive feedback in our model is the progressive weight increase of the synapse, which orientates the agents towards one behaviour, such as food seeking. The negative feedback in our model is the collisions resulting from a crowded group of agents competing for food. The balance between these two processes has been shown to be a stable and flexible decision system (Beekman et al. 2009; Meyer, Beekman & Dussutour 2008). Such self-organising properties have been extensively studied for animals (Franks et al. 2003; Beekman et al. 2009; Meyer, Beekman & Dussutour 2008; Kernbach et al. 2009), bacteria colonies (Reading & Sperandio 2006) and economic markets (Weisbuch & Stauffer 2000). In our computational model the agent needs to decide whether to obtain some food itself or steal the food from the others. Because each agent has no initial preference or bias, the agent needs to make a decision at the individual level based on his memory (synaptic weight/filter response) and actual sensory inputs. The constraining conditions control the emergence of the subsystems (Salgado & Gilbert 2008; Emmeche, Koppe & Stjernfelt 2000), which in our case are the number of available food sources. However, there is no guarantee that emergence happens; social systems might just not differentiate into subsystems (Camazine et al. 2001), giving systems that Luhmann calls "simple social systems" (Luhmann 1984: 263).

« 65 » In order to identify subsystems, we have used a measure that we call Predictive Utilisation (PU). This is a further departure from our Predictive Value (Porr, Egerton & Wörgötter 2006), where the value was calculated by using the synaptic weights. Our new measure just observes the inputs and outputs of the agent, treating it as a black box, which is much more in line with the constructivist paradigm (Luhmann 1984: 275). While in our previous work (Porr, Egerton & Wörgötter 2006) we used the agents' weights, coding anticipatory action to derive a performance measure, we now just use what can be observed from the outside. This is still not completely satisfactory because an external observer is only able to observe the motor output, while the agent can only observe its inputs. However, the agent can solve this issue by calculating the mutual information at two points in the loop, where one is the agent's input and the other is towards the agent's outputs. This is, for example, achieved with efference copies or the production of a press release in a company. Our prediction utilisation seems to be related to performance measures for feedback systems (Touchette & Lloyd 2004). However, we are not interested in closed loop performance per se because we assume that all feedback loops already work and are stable (see introduction). Rather, we are interested in the transformation from reflexive to proactive actions, which can be understood as the development of forward models that use predictions to guide the agent to its goal (Porr, von Ferber & Wörgötter 2003). Therefore we call our measure "prediction utilisation," measuring the agent's transformation into an anticipating agent. As an outlook, it would be very interesting to apply the information theoretic framework devised by Jost et al. (2007) to our social system model to detect subsystems and then in a more relevant social context, for example the financial system.

« 66 » Having two different subsystems that are characterised by agents that select only certain aspects of the environment is in line with Luhmann's understanding of subsystems (Luhmann 1984: 37). However, he goes further, namely that later in the process, the subsystem develops a binary code that defines whether the agent is part of this subsystem or not. A binary code is not developed in our model because both the underlying control system and the communication system use linear weights and the actions are gradual, which means that the belonging to one or the other subsystem is gradual. One could also argue that in order to create binary codes, the agents need to be able observe their own loops (or their omissions). This ability is needed not only to change their behaviour but also to create a communication system that is able to self-organise a binary code that then in turn could be used to inform the action system. However, subsystem formation still starts with the removal of loops, which Luhmann calls selection, so that even though we cannot develop a binary code or a symbolic generalised communication medium, we can still talk about subsystem formation.

« 67 » Even though a binary code is not developed in our model, we can still use our PU to analyse social systems but using it to compare how they react to perturbations. Remember that the PU is a measure that tests whether a subsystem uses predictors to combat certain perturbations that can be generalised to other systems. For example, in a company the PU can be used to investigate how much it cares about perturbations from the outside. A competitor might produce a similar product, which in turn will diminish revenue, which will force the company to react. In terms of the PU, this is our non-optimal reflex reaction and in the worst case will render the company bankrupt. In order to prevent this, the company needs to develop predictors so that a product launch from their competitor can be counteracted before it is too late. This reaction can be termed a "predictor." If this action has been successful, then the reflex is no longer needed. On the other hand a company might simply ignore another competitor because it has decided to specialise in a certain product area because reacting to everything on the market creates too much uncertainty within the company.

5 Conclusion

« 68 » In this paper we have taken cybernetic control systems, creating agents that comply with the constructivist approach. We have opted here for a purely linear control system, where processing is performed similarly to in Ashby's original

BERND PORR
has degrees in physics and journalism and a Ph.D in computational neuroscience from the University of Stirling. Since 2004, he has been a lecturer in electronics and electrical engineering at the University of Glasgow. His research interests range from neurophysiology, through biologically-inspired robotics, to social systems. He developed a range of closed loop learning rules (ISO, ISO3 and ICO learning), is the creator of the RunBot a biped robot which uses reflexes to walk, has been investigating the role of neuromodulators in action selection and has designed/commercialised electronic devices such as data acquisition equipment and deep brain stimulators. When he is not doing research he writes and directs films both, factual and fiction.

PAOLO DI PRODI
Paolo Di Prodi's past research was the application of information theory to computational neuroscience with focus on STDP and multi agent systems. He currently works as a research analyst for a security firm and is using adaptive learning for classification of malware and intrusion detection. He also holds a patent for an intelligent sensor that monitors moisture in building walls and has developed a portable all in one weather station for predicting approaching storms.

homeostat. Such a control approach is in accordance with the constructivist paradigm because it establishes an ongoing process that has no final goal but is rather driven by intrinsic motivations defined by desired states that in turn are maintained by sensor/motor loops. A purely control-theoretical approach has the advantage that it emphasises the ongoing processing of loops (Froese & Ziemke 2009). However, the generation of more complex actions, the switching of actions and the sequencing of actions is virtually impossible in a simple linear control context. More flexible actions, new actions and sequences of actions can be achieved by reinforcement learning (Sutton & Barto 1998), for example. However, standard reinforcement learning algorithms work towards an extrinsically defined reward. This usually means that the life of the animal is just directed towards this single moment in time but will not code an ongoing intrinsic motivation. Richard Sutton et al. (2011) have recently suggested a solution to the problem, where an agent is governed by a large number of reinforcement learners that implement intrinsic rewards and cooperate with each other. The authors in Sutton et al. (2011) also pointed out another important aspect that could be implemented in a more advanced model: exploration. Ongoing processing in terms of the constructivist approach does not necessarily mean that an agent continuously improves its happiness level or minimises the effects of perturbations. It also requires a phase where the agent is not directly improving its behaviour but rather learns via exploration ("off policy learning"). Another aspect to consider is the state space. Discrete actions and sensor events allow a straightforward clearly distinguished action selection and learning, in contrast to our approach, where the actions are established by linear weights. Selection of sensor/motor schemas can be selected in a binary fashion, which is often implemented in the actor in reinforcement learning. Avoiding the pitfalls of extrinsic rewards by defining an intrinsic "satisfaction," one can create constructivist models that act in discrete space and allow the combination of schemas and the creation of scheme hierarchies (Georgeon, Marshall & Manzotti 2013).

« 69 » In this work we started with predefined reflexes; learning then took place driven by these reflexes. Reflexes guarantee that agents can react to perturbations from the outset, which in turn guarantees that their autopoiesis continues. These reflexes then drive learning directly via ICO learning. Coming back to the previous paragraph, one could use a much more loosely coupled approach where the agent largely performs learning "off policy," as in Sutton et al. (2011), and performs mostly low level exploratory behaviour. This approach still prevents the agent from disintegrating but at the same time can generate new sensor/motor loops to improve its "satisfaction." However, as shown in this paper, this would lead to more and more unpredictable behaviour and thus in turn again would impact on the robot's satisfaction levels. In such a more indirect learning scheme, the agents again need to create more predictability to specialise. Looking at the definition of our PU, one could define the low level sensorimotor coordinations as reflexes and then the learned sensorimotor behaviours as the predictive pathways.

Appendix: The behaviours

« 70 » We have two motor neurons $motor_{L/R}(t)$ that receive input from the three different behavioural sub-circuits: food, stealfood and avoid:

$$\begin{aligned} motor_L(t) = {} & B - ICO_{avoid,L}(t) \\ & - ICO_{food}(t) - ICO_{stealfood}(t) \\ motor_R(t) = {} & B - ICO_{avoid,R}(t) \\ & - ICO_{food}(t) - ICO_{stealfood}(t) \end{aligned} \quad (12)$$

where $B = 2.4$ is a constant bias for the motors so that the agents drive straight ahead with no other input. The signals ICO_{food} and $ICO_{stealfood}$ can be interpreted as steering angles (or error signals), where a zero value means straight ahead and a non-zero represents turning behaviour. The signals $ICO_{avoid,L/R}$ are coded slightly differently because they generate a retraction behaviour after the robot has crashed into a wall and thus need to be able to reverse the speed of the robot. The output of each motor neuron $motor_{L/R}(t)$ is then fed into a sigmoid function that normalises the output to $[-1, 1]$ to control the speed of the robot motors:

$$\begin{aligned} speed_L &= speed_{\max} \frac{1}{1+e^{-motor_L(t)}} \\ speed_R &= speed_{\max} \frac{1}{1+e^{-motor_R(t)}} \end{aligned} \quad (13)$$

where $speed_{max} = 1$ is the maximum speed of the robot in pixels per time step.

« 71 » The agents' behaviours that generate the signals $ICO_{avoid,L/R}(t)$, $ICO_{food}(t)$ and $ICO_{stealfood}(t)$ are now presented one by one. We also show how anticipatory learning is achieved in these three cases.

A.1 Food attraction

« 72 » Before learning, the agent has only its inert reflex behaviour:

$$\begin{aligned} d_{food,reflex}(t) &= x_{food,prox,L}(t) - x_{food,prox,R}(t) \\ u_{food,reflex}(t) &= d_{food,reflex}(t) * h_{food}(t) \end{aligned} \quad (14)$$

« 73 » The signal $d_{food,reflex}$ is an error signal generated by subtracting the right proximal sensor signal from the left proximal sensor signal. The lowpass filter h_{food} simply smooths out the error signal, which is required for learning and also generates the trajectory towards the target. It is a second order filter with a cut-off frequency $f_0 = 0.3$ and a Q-factor of $\sqrt{2}$ (see Porr & Wörgötter 2002 for details).

« 74 » In order to learn to steer to a food disk from a distance, the robot utilises the signals from the distal sensors.

$$\begin{aligned} d_{food,pred}(t) &= x_{food,dist,L}(t) - x_{food,dist,R}(t) \\ u_{food,pred}(t) &= d_{food,pred}(t) * h_{food}(t) \end{aligned} \quad (15)$$

« 75 » Again, we first generate an error signal $d_{food,pred}$ and then smooth it out with the lowpass filter h_{food}. This smoothing out of the signal also acts as a memory trace so that learning can correlate the later reflex reaction with this predictive signal. Both proximal and distal smoothed-out error signals then converge on the ICO neuron:

$$\begin{aligned} ICO_{food}(t) = {} & u_{food,reflex}(t) \cdot \omega_{food,reflex} + \\ & u_{food,pred}(t) \cdot \omega_{food,pred}(t) \end{aligned} \quad (16)$$

« 76 » The weight $\omega_{food,reflex} = 2.7$ is fixed and generates the inert reflex reaction to approach food close to the agent. The weight $\omega_{food,pred}$ is changed by the ICO learning rule:

$$\frac{d\omega_{food,pred}}{dt} = \mu \cdot u_{food,pred}(t) \frac{du_{food,reflex}(t)}{dt} \quad (17)$$

by correlating the derivative of the reflex signal $u_{food,reflex}(t)'$ with the signal from the distal sensors $u_{food,pred}(t)$. Learning continues until the agent's reflex is no longer needed ($u_{food,reflex}(t)' = 0$) and steering is achieved by the predictive signal $u_{food,pred}(t)$, only.

A.2 Carrying food

« 77 » Agents eat up food slowly at a rate defined by $\tau_{consumption}$:

$$myfood_i(t) = e^{-(t-t_b)/\tau_{consumption}} \quad (18)$$

where *myfood* is the agent's food reservoir it is carrying around and $\tau_{consumption}$ its the consumption rate. t_b corresponds to the moment when an agent touches a food source and is set to t every time the agent encounters food:

$$t_b := t \quad \text{if} \quad |x_{food,prox,L}(t_b) + x_{food,prox,R}(t_b)| > \theta_{food} \quad (19)$$

or touches an agent carrying food:

$$t_b := t \quad \text{if} \quad |x_{myfood,prox,L}(t_b) + x_{myfood,prox,R}(t_b)| > \theta_{agent} \quad (20)$$

« 78 » Every agent broadcasts its *myfood*-state to all other agents. This state can be detected from a distance in the form of a decaying field in the same way as food disks.

A.3 Stealing food from other agents

« 79 » When an agent bumps into another agent and the agent has food ($myfood > 0$) this then causes a reflex reaction and the agent "steals" the food from the other agent:

$$\begin{aligned} d_{stealfood,reflex}(t) = {} & x_{stealfood,prox,L}(t) \\ & - x_{stealfood,prox,R}(t) \\ u_{stealfood,reflex}(t) = {} & d_{stealfood,reflex}(t) \\ & * h_{stealfood}(t) \end{aligned} \quad (21)$$

where $d_{stealfood,reflex}(t)$ is the error signal generated when one agent steals food from the another agent and $h_{stealfood} = h_{food}$. This error signal is then used for ICO learning to generate an anticipatory action by sensing the *myfood* state of other agents from the distance. As stated before, the agents broadcast their *myfood* state to the other agents and excite the sensors $x_{stealfood,dist,L}(t)$ and $x_{stealfood,dist,R}(t)$ from a greater distance:

$$\begin{aligned} d_{stealfood,pred}(t) = {} & x_{stealfood,dist,L}(t) \\ & - x_{stealfood,dist,R}(t) \\ u_{stealfood,pred}(t) = {} & d_{stealfood,pred}(t) \\ & * h_{stealfood}(t) \end{aligned} \quad (22)$$

« 80 » These signals then converge in a weighted fashion on our ICO neuron responsible for the stealing behaviour:

$$\begin{aligned} ICO_{stealfood} = {} & u_{stealfood,reflex}(t) \cdot \omega_{stealfood,reflex} + \\ & u_{stealfood,pred}(t) \cdot \omega_{stealfood,pred}(t) \end{aligned} \quad (23)$$

where the weight $\omega_{stealfood,reflex} = 3.1$ is fixed and the weight $\omega_{stealfood,pred}$ is changed by the ICO learning rule:

$$\frac{d\omega_{stealfood,pred}}{dt} = \mu \cdot u_{stealfood,pred}(t) \cdot \frac{du_{stealfood,reflex}(t)}{dt} \quad (24)$$

by correlating the reflex $d_{stealfood,reflex}(t)$ with $u_{stealfood,pred}(t)$, which contains broadcast information about the other agents' food reservoir (*myfood*). After learning, an agent is able to drive towards other agents that carry food and steal the food from them. The higher the weight $\omega_{stealfood,pred}$, the stronger the attraction towards other agents carrying food.

A.4 Avoidance behaviour

« 81 » The avoidance behaviour simply learns avoiding crashing into the walls and other robots by learning to use anticipatory sensor signals. This behaviour is not used in the social system analysis but is still required so that agents receive predictable sensor signals that are not disrupted by multiple collisions.

« 82 » As mentioned before, ICO learning is used to generate the steering of the left and right motors by using the anticipatory and reflex inputs:

$$\begin{aligned} ICO_{avoid,L}(t) = &\\ &\omega_{avoid,reflex,L} \cdot u_{avoid,reflex,L}(t)\\ &+\omega_{avoid,pred,L}(t) \cdot u_{avoid,pred,L}(t)\\ &+\omega_{self,L}(t) \cdot ICO_{avoid,self,L}(t-1)\\ &+\omega_{R2L}(t) \cdot ICO_{avoid,R2L}(t) \end{aligned} \tag{25}$$

$$\begin{aligned} ICO_{avoid,R}(t) = &\\ &\omega_{avoid,reflex,R} \cdot u_{avoid,reflex,R}(t)\\ &+\omega_{avoid,pred,R}(t) \cdot u_{avoid,pred,R}(t)\\ &+\omega_{self,R}(t) \cdot ICO_{avoid,self,R}(t-1)\\ &+\omega_{L2R}(t) \cdot ICO_{avoid,L2R}(t) \end{aligned} \tag{26}$$

with $\omega_{avoid,reflex,L} = -0.9$, $\omega_{avoid,reflex,R} = -1.0$ so that hitting an obstacle causes a retraction reaction. The slight differences in the weight cause the robot to turn slightly when hitting an object dead on. Here, the ICO neurons have recurrent synaptic connections and a push pull mechanism between left and right motor neuron as $\omega_{R2L} = -0.42$, $\omega_{L2R} = -0.3$, $\omega_{self,R} = 0.4$, $\omega_{self,L} = 0.4$ to implement a hysteresis effect. This effect causes the controller not to follow signals with a slight delay, as shown in Wischman Pasemann & Hülse (2004) and Hülse & Pasemann (2002). It means reactions on an incoming signal are time shifted. This is useful to enable agents to escape from corners: if there were no hysteresis mechanism, an agent would get stuck in a corner forever, turning left and right alternately.

« 83 » Learning generates anticipatory actions by using the information from the long range sensors $x_{avoid,pred,L/R}$ and updating their corresponding weights in such a way that the agent steers away from a wall before it crashes into it. This is achieved by ICO learning, which updates the weights $\omega_{avoid,pred,L/R}$:

$$\frac{d\omega_{avoid,pred,L}}{dt} = \mu \cdot u_{avoid,pred,L} \cdot \frac{du_{avoid,reflex,L}}{dt}$$
$$\frac{d\omega_{avoid,pred,R}}{dt} = \mu \cdot u_{avoid,pred,R} \cdot \frac{du_{avoid,reflex,R}}{dt}$$

Received: 5 October 2012
Accepted: 16 January 2014

Open Peer Commentaries

on Bernd Porr & Paolo Di Prodi's "Subsystem Formation Driven by Double Contingency"

Learning by Experiencing versus Learning by Registering

Olivier L. Georgeon
Université de Lyon, France
olivier.georgeon/at/liris.cnrs.fr

> Upshot • Agents that learn from perturbations of closed control loops are considered constructivist by virtue of the fact that their input (the perturbation) does not convey ontological information about the environment. That is, they learn by actively experiencing their environment through interaction, as opposed to learning by registering directly input data characterizing the environment. Generalizing this idea, the notion of learning by experiencing provides a broader conceptual framework than cybernetic control theory for studying the double contingency problem, and may yield more progress in constructivist agent design.

« 1 » Ernst von Glasersfeld differentiated radical constructivism from realist epistemology by the relation between knowledge and reality:

> “Whereas in the traditional view of epistemology, as well as of cognitive psychology, that relation is always seen as a more or less picture-like (iconic) correspondence or match, radical constructivism sees it as an adaptation in the functional sense.” (Glasersfeld 1984: 20)

This suggests differentiating constructivist artificial agents from realist artificial agents by the relation between their input data and their environment. As illustrated by Bernd Porr and Paolo Di Prodi's implementation, in cybernetic theory, the agent's input (called perturbation) does not hold an "iconic correspondence" with the environment but rather consists of feedback from the agent's output (called action). In contrast, as we shall develop below, most machine-learning algorithms implement this iconic correspondence because they implement and exploit the agent's input as if it directly characterized the environment, thus representing a direct access to the ontological essence of reality.

« 2 » Here, we call *learning by experiencing* those learning mechanisms that implement and exploit input data as feedback from the agent's output, and *learning by registering* those learning mechanisms that implement and exploit input data as a direct observation of the environment (either a simulated environment or the real world, in the case of robots). This formulation complies, for example, with Etienne Roesch et al.'s formulation that constructivist epistemology considers knowledge as resulting from experience of interaction with the environment, as opposed to existing "in an ontic reality […] available to registration from the physical world" (Roesch et al. 2013: 26).

« 3 » Partially Observable Markov Decision Process models (POMDP; Kaelbling, Littman & Cassandra 1998) well exemplify learning by registering because they typically formalize the agent's input as a function of the environment's state only. A similar argumentation can show that many other machine-learning approaches learn by registering, even supposedly constructivist approaches based upon schema mechanisms (e.g., Drescher 1991)[1] and many approaches based upon multi-agent systems, such as Roesch et al.'s (2013) agents, as we discussed in our open peer commentary (Georgeon & Hassas 2013).

« 4 » For the sake of argument, consider a POMDP in which the agent's input (called observation) is reduced to a single bit. A subset *S0* of the set *S* of all the environment's states are observed as "0", and the states in the complementary subset *S1* are observed as "1". Because of stochastic noise, some elements of *S0* may occasionally be observed as "1" and the other way around. Yet the observation statistically reflects the state of the environment, and the agent's policy generally exploits this assumption to try to construct an internal model of the agent's situation. To our knowledge, there is no POMDP implementation that would exhibit interesting behaviors with as little observation as a single bit when the number of states is great. This limitation is known as the perceptual aliasing problem (Whitehead & Ballard 1991), and is inherent to learning by registering.

« 5 » Note that some variations of POMDPs have been proposed in which the scope of the observation depends on the previous action, thus involving a form of active perception (e.g., McCallum 1996). However, the observation still reflects the state of the environment, as if the environment was observed through a filter that varied with the action.

« 6 » In contrast, mechanisms of learning by experiencing implement the agent's input such that it conveys information about the effect of an "experiment" performed by the agent. In the case of a single input bit, this bit indicates one out of two possible outcomes of the experiment. The same particular state of the environment induces different input bits depending on the experiment initiated by the agent. No partitioning of the set of states *S* can be made according to the input bit because all states may induce input "0" or "1," depending on the experiment. In this case, the learning algorithm must not exploit the agent's input as if it statistically and partially corresponded to the state of reality, because it does not. In contrast with learning by registering, there exist single-input-bit learning-by-experiencing agents that exhibit interesting learning behaviors (e.g., Georgeon & Hassas 2013; Georgeon & Marshall 2013).

« 7 » Besides cybernetic control theory, in §68, Porr and Di Prodi mention other examples of learning by experiencing: Richard Sutton et al.'s (2011) Horde architecture, and our work. Horde relies on a swarm of reinforcement-learning agents to learn hierarchical temporal regularities of interaction through experience. More broadly, learning by experiencing implements a form of conceptual inversion of the perception-action cycle recommended by some authors (e.g., Pfeifer & Scheier 1994; Tani & Nolfi 1999). In learning by experiencing, however, calling the input a perception or an observation is misleading because the input does not hold a direct correspondence with reality.

« 8 » Concerning our approach, we shall clarify that it does not only "act in discrete space," as Porr and Di Prodi wrote in §68. Instead, our agents are indifferent to the structure of their environment's space, which is precisely an advantage of learning by experiencing. We demonstrated that our algorithms could control agents in continuous two-dimensional simulated environments (Georgeon & Sakellariou 2012) and robots in the real world (Georgeon, Wolf & Gay 2013). It is true that our agent's set of possibilities of experience (the relational domain defined by the coupling between the agent and the environment, e.g., Froese & Ziemke 2009) is discrete, but this does not prevent the agent from learning interesting behaviors in continuous space.

« 9 » Since learning-by-experiencing (LbE) agents do not directly access the state of the environment, they incorporate no reward function or heuristics defined as a function of the state of the environment. This places LbE agents in sharp contrast with reinforcement-learning agents and problem solving agents. Notably, LbE agents even differ from reinforcement-learning agents with an intrinsic reward (e.g., Singh, Barto & Chentanez 2005), which consider some elements of the state of the world to be internal to the agent. As a generality, an LbE agent gives value to the mere fact of enacting interactive behaviors rather than to the state resulting from behaviors. We expect LbE agents to demonstrate that they learn to "master the laws of sensorimotor contingencies" (O'Regan & Noë 2001). Consequently, as some authors in the domain of intrinsic motivation also argued (e.g., Oudeyer, Kaplan & Hafner 2007), we recommend assessing LbE agent's learning through behavioral analysis rather than through a measure of their performance in reaching specific goals.

« 10 » In accordance with our view on LbE agent assessment, Porr and Di Prodi assess their agent's learning through behavioral analysis (Section 4). Their agents are motivated to interact with entities present in the environment by controlling sensorimotor loops (approaching food or other agents, §18). For each sensorimotor loop, Porr and Di Prodi define Prediction Utilization as a measure of the agent's commitment to control this loop. We wish to support their effort in specifying this kind of measure. This effort contributes to defining general quantifiers that could be used with other learning-by-experiencing approaches to characterize the agent's engagement in interactive behaviors.

« 11 » As Porr & Di Prodi noted in §68, simple linear control theory does not realize "the generation of more complex actions, the switching of actions and the sequencing of actions." However, other learning by experiencing approaches tackle these issues.

[1] Gary Drescher (1991) modelled Piagetian schemes as triplets – <pre-observation, action, post-observation>. The argument that Drescher's agent's observation reflects the environment's state is similar to our argument about POMDPs.

Addressing the double contingency problem with approaches that generate such learning would allow more sophisticated subsystem organization because each subsystem could control more sophisticated interactions than a linear control loop. Therefore, we anticipate that addressing the problem of subsystem formation driven by double contingency within the general framework of learning by experiencing would allow more advances in constructivist agent design.

Olivier L. Georgeon is currently an associate researcher at the LIRIS Lab, with a fellowship from the French Government (ANR-RPDOC program). He received a Masters in computer engineering from Ecole Centrale de Marseille in 1988, and a PhD in psychology from the Université de Lyon in 2008.

Received: 19 February 2014
Accepted: 19 February 2014

Aligning Homeostatic and Heterostatic Perspectives

Patrick M. Pilarski
University of Alberta, Canada
pilarski/at/ualberta.ca

> **Upshot** • There is merit to the continuous-signal-space homeostatic viewpoint on subsystem formation presented by Bernd Porr and Paolo Di Prodi; many of their ideas also align well with a heterostatic constructivist perspective, and specifically developments in the field of reinforcement learning. This commentary therefore aims to identify and clarify some of the linkages made by the authors, and highlight ways in which these interdisciplinary connections may be leveraged to enable future progress.

« 1 » Learning to perceive, predict, and act based only on continuous-valued sensorimotor inputs and outputs is a challenging and important pursuit that deserves our focused attention. While subsystem formation and evaluation are the principal listed contributions of Bernd Porr and Paolo Di Prodi's target article, the nature of the signals and predictions in the paper's problem domain are crucial points that impact how we interpret the comparisons made in the paper and the ways its insights may be applied to work in other domains. I use Porr and Di Prodi's problem setting and agent formulation as a starting point for assessing some of the key statements made in their work, and build toward a specific look at prediction utilization as presented by the authors. This assessment is supplemented with comparisons to related work from the recent computational and biological literature.

Heterostasis and homeostasis

« 2 » Porr and Di Prodi's setting of agents interacting in a reflexive and predictive manner via continuous inputs and outputs is a natural one, albeit one that is often ignored in favour of the perceived clarity and mathematical benefits of discrete sensation and action spaces. Their specific setting is in fact a problem domain that resonates well with other robot-related constructivist demonstrations from the machine learning literature – e.g., learned multi-robot food foraging behaviour (Matarić 1997), robot learning applications as surveyed by Grondman et al. (2012), and robot knowledge acquisition as per Modayil, White & Sutton (2014) and Sutton et al. (2011, as cited by the authors). It is important to note, however, that many of these like-minded explorations are rooted in a rather different starting point: that of the learning system or systems attempting to maximize some aspect of its experience – in other words, an agent seeking to increase its long-term expected reward or learning progress, as in the intertwined fields of computational and biological reinforcement learning (Sutton & Barto 1998). This maximization, or *heterostatic* goal-seeking behaviour (after Harry Klopf's *The Hedonistic Neuron*, 1982) is at first glance in contrast with an agent's "task of restoring its desired state to homeostasis," as posed by the authors (§3). However, for our current discussion, it may be beneficial to explore the similarities between these viewpoints in terms of the authors' work, as opposed to the differences.

« 3 » Let us first examine the statements made in the authors' concluding remarks, suggesting that the homeostatic linear control approach in the paper "establishes an ongoing process that has no final goal but is rather driven by intrinsic motivations that are defined by desired states." (§68) After acknowledging the need and potential for more complex actions and action sequences, as potentially provided by techniques from reinforcement learning, the text of §68 continues by stating that the extrinsically defined reward used in standard reinforcement learning "usually means that the life of the animal is just directed toward this single moment in time but will not code an ongoing intrinsic motivation."[1] This sequence of text sets up a natural contrast between extrinsic and intrinsic reward – motivation or satisfaction derived from the world or from within the agent, respectively. At the same time, it reinforces a distinction between heterostatic and homeostatic optimization by an intelligent system.

« 4 » Intrinsic motivation is held to be a powerful way to drive exploration and potentially accelerate the learning of predictions, control behavior, and better representations (Schmidhuber 1991; Oudeyer, Kaplan & Hafner 2007). However, much like the actual boundary between an agent and its environment is often less of a boundary and more an opinion on the part of the system designer (or examiner), boundaries between what are considered intrinsic and extrinsic reward have been placed at different points by different authors. Is the distinction between these types of feedback actually useful to our discussion of the present paper, or does it further cloud the understanding of how Porr and Di Prodi's agents react to perturbations in their sensorimotor streams?

« 5 » One high-level view we could be inclined to take based on the statements made in §68 is that an intrinsic approach to motivation allows ongoing, life-long learning without the need for endpoints or imposed valuations of an agent's stream of experience (e.g., transient or final goals). However, it is interesting to refer again to the aforementioned text in §68 indicating

[1] This statement seems to assume a terminal or discrete reward, and passes over the way that standard reinforcement learning often utilizes temporally extended expectations of future reward (e.g., *discounted future return;* Sutton & Barto 1998) or average reward (discussed below). However, a detailed discussion of all these points is best left outside the present commentary.

that an agent has "desired states." We also see this allusion to desire, or the relative valuation of changes to an agent's input, phrased as "irritations" (e.g., §5 and §16). Whether intrinsic or extrinsic, valuation and goal seeking seems to play a role in the authors' learning system and the way they describe that system.

« 6 » Here again, language may be clouding our view on the learning setting of interest. Let us return to a heterostatic view via reinforcement learning, where goal-based desire and state-action valuation is commonplace. Reinforcement learning is a form of optimal control, where prediction change and policy change are driven by temporal-difference error signals derived from the incoming stream of data. One signal is labeled as special, namely, the reward. This is not so different from the learning presented by the authors, wherein the difference between two sensors on their simulated robot is formulated as an error signal (thought of as a negative outcome, §5 and §16) and used to drive weight change in another unit. This differential learning signal could also be phrased as reward – extrinsic or intrinsic, depending on where an examiner decides to place their boundaries. This suggestion to convert a problem from the homeostatic to heterostatic viewpoint and back again is perhaps not a new one, but it is worth recalling when we think on the potential extensions and impact of Porr and Di Prodi's paper.

« 7 » Can we then appeal to the continuous, ongoing nature of a control approach or signal space, for example, Ashby's homeostat (Ashby 1956), as a clear difference between the authors' outlined constructivist viewpoint and the constructivist viewpoint embodied by computational reinforcement learning? Here again, the distinction is not easy to pin down without invoking specific (and possibly distracting) language. A well-known setting in reinforcement learning is that of *average reward*, as described in Sutton & Barto (1998),[2] Grondman et al. (2012), and specifically for a continuous-action-space control learner in Degris, Pilarski & Sutton (2012). In the average reward setting, best thought of in terms of continuing problems with no clear end or stopping point (e.g., life-long control learning that proceeds from raw sensorimotor data), the error signals that drive an agent's valuations of the goodness or badness of a situation are calculated in terms of the difference between the instantaneous reward (arriving at every slice of time from either a privileged reward channel or as computed from a combination of the agent's sensors) and an adaptable baseline that has been acquired through ongoing experience – the state-invariant average reward. In other words, in the average-reward setting, "one neither discounts nor divides experience into distinct episodes with finite returns. […] one seeks to obtain the maximum reward *per time step*" (Sutton & Barto 1998: 153; my emphasis). This can perhaps also be thought of as minimizing the unpleasant divergence from a steady homeostatic state, as posed by the authors as the engagement of reflex actions in §23 and §27.

Prediction-driven behaviour

« 8 » With these possible symmetries in mind, let us turn to Porr and Di Prodi's use of learned predictions as inputs, and as incrementally engaged motor alternatives to hardcoded reflex actions. This usage is another key point and incredibly valuable perspective put forward by the authors' work, and it echoes findings in the study of the human brain. In particular, the authors' experimental formulation links nicely with the way predictions are thought to be incrementally learned (e.g., via the cerebellum) and then mapped in a fixed way to influence or determine animal action selection – i.e., the "Pavlovian action selection" of David Redish (2013) and work by David Linden (2003). Though Porr and Di Prodi's predictions are perhaps best thought of as long-range sensors that gradually come online, as opposed to predictions that are built up over time, the anticipatory ideas are largely the same. Their approach can be framed as a view into the gradual acquisition and use of *predictive representations of state* (Littman, Sutton & Singh 2002).

« 9 » As one specific example, Porr and Di Prodi's ideas resonate with our group's recent demonstration of how predictions that are learned and adapted in real time can help remove control delays – well viewed as negative outcomes as per §5 – during the operation of a physical human-robot interface (Pilarski, Dick & Sutton 2013). By combining continuous-action reinforcement learning control approaches with knowledge acquisition methods from Sutton et al. (2011) and Modayil, White & Sutton (2014), it was shown that a robot learner could use its predictive inputs to actuate joints preemptively. The robot learned to act prior to stumbling upon its motor objective and being forced rapidly to correct its orientation. This example has an interesting symmetry with Porr and Di Prodi's setting of using predictions to steer toward a food source or other agent preemptively so that the system is not forced to take rapid, reflexive action at a future point. Effecting Porr and Di Prodi's shift from reflex to anticipatory actuation in the setting described by Pilarski, Dick & Sutton (2013) – either using divergence from homeostasis, reward, or motor commands related to changes in prediction magnitude – would be a nice demonstration and highlights one way that the present paper by Porr and Di Prodi may catalyze development in other constructivist settings.

« 10 » So, with the discussion above as background, could we reasonably phrase Porr and Di Prodi's experimental setup in heterostatic terms with the learned, temporally extended predictions of Modayil, White & Sutton (2014) and Sutton et al. (2011) in place of distal sensors? First, would there be any benefits to formulating this parallel view and further validating the authors' observations? Second, if we did so, would the results and subsystems formed by the interacting learners be similar? It stands to reason that they might, and this would be a potentially valuable link. In addition to this link extending the impact of Porr and Di Prodi's work into a wide and continually growing body of reinforcement learning literature, it may also be possible for a broader community of researchers to leverage modern developments from reinforcement learning that include representation learning, stable off-policy learning of predictions and control, continuous-state-and-action control learning, and planning that is grounded in sensorimotor experience.

[2] A second edition is in progress and can be accessed at http://webdocs.cs.ualberta.ca/~sutton/book/the-book.html; this new edition comprehensively describes the average reward setting.

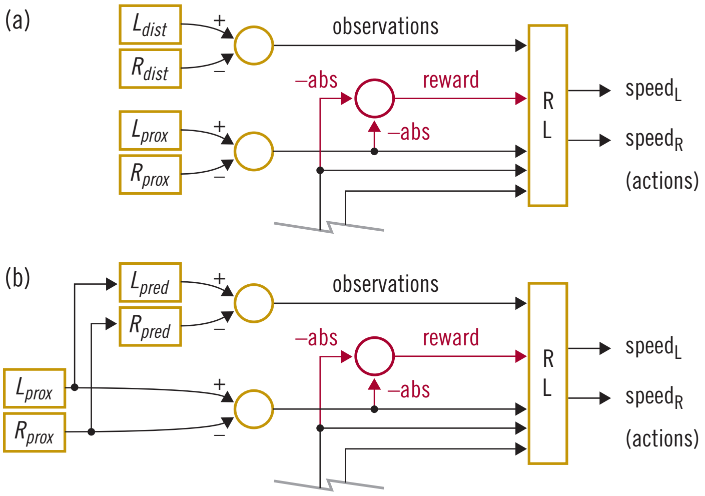

Figure 1: A reinforcement learning formulation for validation experiments using (a) proximal reflexive sensors combined with distal sensors and (b) proximal sensors combined with learned, temporally extended predictions. The generation of differential inputs (observations) to the learning system (denoted RL) is shown for one behaviour type only for clarity; an identical structure can be assumed for the complete system of food attraction and food stealing.

A thought experiment

« 11 » As a concrete suggestion: we could perform a simple experiment to begin investigating the parallel, heterostatic view on the authors' work. This would help us identify whether the subsystem results presented by the authors might indeed arise if the multiple agents in their simulated domain were instead basing their behavior on reward-driven learning and learned predictive representations of state. One case would be to implement a control learner that takes as input the differential signals from all predictive sensors and reflex sensors, outputs two continuous-valued motor commands (§21), and receives a negative reward proportional to divergence of the reflex sensors from their differential set points (Figure 1a).[3] To provide a starting behavior identical to that specified by Porr and Di Prodi, the learner's control policy should be initialized to generate reflexive actions as in §23–§28. As a further extension and alternative to pre-specified distal sensors, each predictive sensor could instead be implemented such that it reports situation-specific, long-range forecasts about the output of a single reflex sensor at one or more time scales (Figure 1b). As noted above, forecasts of this kind can be incrementally acquired during sensorimotor interaction, and can also be acquired even when an agent is not pursuing a behaviour specifically related to the prediction of interest (i.e., *off-policy learning* as in Sutton et al. 2011).

« 12 » Examining the case of two agents and a single food source, one agent would invariably contact the food first (as indicated in §56). In addition to inducing policy change, this interaction would prevent the second agent from experiencing the food and thus building up anticipatory knowledge related to food attraction (temporally extended predictions) and the related motor responses (linked in either a fixed or a learned way to the magnitude of its predictive signals). The second agent would, however, be able to build up anticipations regarding food stealing since the first agent is now carrying food. Building on the initial perturbations experienced by the two (or more) agents, the formation of subsystems could then follow as per Porr and Di Prodi's discussion. This hypothesis can and should be empirically tested for both the simple case of distal sensors (Figure 1a) and the case where predictive sensors are implemented as incremental prediction learners (Figure 1b).

Conclusion

« 13 » There are many good things to take away from Porr and Di Prodi's "Subsystem Formation Driven by Double Contingency." These contributions should be evident to the careful reader, so the aim of this commentary has been to bring out points that may be missed when focusing on the larger claims of the work. This commentary also aimed to emphasize points that may unify our thinking in small ways such that we can move forward more quickly in the understanding of learning and adapting multi-agent systems. Finally, we should ask whether the alignment of homeostatic and heterostatic constructivist viewpoints proposed in this commentary is useful as a default mindset. Likely not. But the quest for life-long, learner-driven representation, prediction, and control could involve a long and bumpy road; I suggest that we need patience and the ability to see both commonalities and differences to follow this road through to its fruitful conclusion.

Patrick M. Pilarski is an Adjunct Assistant Professor in the Department of Computing Science, University of Alberta. As a researcher with the Alberta Innovates Centre for Machine Learning and the Reinforcement Learning and Artificial Intelligence Laboratory, his work focuses on real-time machine learning and adaptive brain-body-machine interfaces for use with assistive robots.

Received: 20 February 2014
Accepted: 24 February 2014

[3] Another plausible example is to examine the drive for agents to return to a desired or expected food carrying state, e.g., positive reward for carrying food (obtained by any means) or negative reward for dropping below an average baseline. This case can also be seen from multiple viewpoints, and might lead to the same type of subsystem formation and reflex-prediction shifts noted by the authors.

The Looping Problem

Fabien Hervouet
University of Montpellier, France
fabien.hervouet/at/lirmm.fr

> Upshot • By analyzing Porr and Di Prodi's model for addressing the double contingency problem, I try to take a step further by questioning the importance and implications of the loop concept in the constructivist approach.

« 1 » In their target article, Bernd Porr and Paolo Di Prodi present a simulation of a self-organizing, learning, multi-agent society. Agents are involved in a traditional constructivist dynamics. They actively perturb and adjust each other continuously, generating more and more unpredictability. This defines the double contingency problem. As pointed out by Niklas Luhmann (1984), a possible solution to this problem resides in a selection mechanism that removes some sensory-motor loops. This entails reducing the agent's awareness of its environment and thus the global unpredictability. This directly derives from the hypothesis that systems form subsystems when their composing entities only concentrate on one aspect of their environment, therefore ignoring all others.

« 2 » The authors' point is that anticipatory learning under external constraints leads to subsystem formation. The main constructivist part in the article comes from how they define their agents. Indeed, the main learning loop only processes information coming from the agent's own sensors and motors, considered as a black box. This is precisely why I would like to address a few aspects of what I call "the looping problem." In this commentary I will analyze and question loops from a constructivist perspective.

More behaviors

« 3 » In the target article, agents are provided with two predefined behaviors, namely *search food* and *steal food*. Each of them is defined by two loops: a reflexive one and a predictive one. The first one is called *inner loop* while the second one is called the *outer loop*. Thereby, the aim of learning process is gradually to reduce the influence of the *inner loop* in favor of the more powerful *outer loop*, in terms of anticipation. This comes from the particular sensory-motor embodiment that requires a delay (§6) between the moments when the distal sensor and the proximal sensor are activated.

« 4 » First of all, one could argue that there may be a bias in the way the predictive loop is coupled to a corresponding reflexive loop. It drives the outcome of the learning process during simulation too significantly. Moreover, demonstrating the emergence of subsystems in a society where only two types of behaviors are predefined also remains in some way biased. As experimental results indicate, the more food sources there are, the more it becomes attractive to search for food. Vice versa, the fewer food sources there are, the more it becomes attractive to steal food from others (§56). A partial solution to this drawback would be to implement other behaviors. But this is likely to require more external constraints in order to exhibit the same kind of differentiation.

« 5 » We could even think of constraints that have a genuine impact on the agent. Here, the food is simply a symbol for the concept of perturbation. Indeed for an agent, having its food reservoir full or empty does not really impacts its behavior.

Generating loops

« 6 » In the authors' model, the agent is provided with three sensory-motor behaviors, whose function is *a priori* defined. Thus, the role of the agent is to improve its adaptiveness using learning, seen as the correlation between proximal and distal sensors (§19). The agent's aim is to anticipate the trigger of the reflex with the help of distal sensors. In short, proximal sensors lead to hardwired reflexes while distal sensors lead to learned predictions. To this purpose, the agent is driven to maintain a desirable errorless state, decreasing the reflex loop entropy to 0 as a side-effect. After learning, steering must only be achieved by a predictive mechanism. This is what they called a transfer from reactive to proactive behavior, from uncertainty to predictability, or in other words, the transition from a reflexive to a predictive way of interacting (§65). The predictive interaction is preferred because of its faster reactivity to dangerous situations and its planning behavior.

« 7 » This underlines that the model only describes how agents can improve a particular aspect of pre-given sensory-motor loops. In this case, it means that an agent can only balance the prevalence of the predictive aspect over the reflexive one. In my opinion, the greatest challenge remains: how can the agent generate new sensory-motor loops on its own?

« 8 » Enacting new sensory-motor loops may possibly be tackled through the concept of schemas, developed by Jean Piaget. Schemas are sensory-motor structures emerging from the repetitive activity of sequences that crystallize. If so, it may constitute an argument for largely accepting and promoting self-exploration.

Latent learning within low-level exploration

« 9 » Would this especially make sense if there were no predefined behaviors but rather agents involved in some very low-level sensory-motor coordination learning from scratch? The target model proposes learning as being driven by predefined reflexes, which according to the authors, guarantee a minimal reaction from the agent to perturbations. They also explain that it is mandatory to create more predictability for specialization to emerge (§69). They argue that performing low-level exploratory behavior in the form of what they call "off-policy" in reinforcement learning would lead to more unwanted and unpredictable behaviors. But in fact, by not taking into account any kind of latent learning, it leads to restricting the life of agents to a more or less predefined area. This does not leave room for emergence.

« 10 » This aspect also comes into conflict with the constructivism view of cognition as an open-ended development through sense-making. As pointed out by Varela (1989), while heteronomous systems *create a representational relationship* with the environment (logic of *correspondence*), autonomous systems *enact a semantic relationship* with the environment (logic of *consistency*). Following Steels (2004), we believe that this sense-making development can be achieved through *autotelism*. This notion goes beyond the classical reinforcement learning framework initiated by behaviorist psychology, allowing an agent to construct grounded sensory-motor structures, continuously balancing challenge and skill to drive its very own development.

Contextual selection mechanism

« 11 » There is a very interesting point stressed by the target article: it clearly stresses how external resources can shape behavior differentiation in agent society. This is indicated through the *Predictive Utilisation* measure that the authors introduce. This may really be considered to apply or adapt to other implementations of learning agents. Still, even if the influence of the environment may be more easily observable with predefined behaviors, experimental results show the emergence of subsystem formation in the simulation.

« 12 » Nevertheless, following the previous questions, we could wonder whether the selection of loops may rather occur according to the ephemeral context? Enabling or disabling some loops may somehow be considered in a more dynamic way, depending on a specific on-the-fly experienced context. This would possibly remain consistent with the double contingency problem. If we imagine a scenario where the number of food sources varied over time, it might lead to dynamic subsystem formation under dynamic constraints.

Parallel co-constructed loops

« 13 » Besides, we could speculate whether specifying such a closed sensory-motor loop for each behavior may be in full agreement with radical constructivism. The closed loop remains a sequential concept originating from traditional causal thought. But if we follow the requirements promoted by enactivism (Varela, Thompson & Rosch 1991), for instance, I am not sure whether we can deduce that the closed-loop is the best possible implementation for modeling cognitive mechanisms.

« 14 » The traditional vision of the closed-loop in first-order cybernetics and control theory is architectured around the notion of feedback and anchored in a *sensing-thinking-acting* loop. There would then be, as the authors seem to agree and promote in their model, a new paradigm arguing that humans control their input/sensation rather than their output/behavior in a homeostatic fashion (Riegler 2007). But this leads to a model that is not flawless either.

« 15 » One can rather imagine multiple sensory and motor loops evolving in parallel on their own while interacting continuously. This parallelization may eventually approach a complex system vision of the construction of cognition. Within this framework, sensory and motor loops would become entities in themselves, reciprocally disrupting and adjusting each other. As evidence of this, in the sensory-motor contingencies theory, O'Regan, Myin & Noë (2005) insist on an intrinsic quality of the feeling of presence, which they call "grabbiness" (or alerting capacity). This is that sensory systems possess sudden change detectors and are able to interrupt the ongoing cognitive activities and cause an automatic orienting response. How could this mechanism be achieved if there were no dedicated sensory loop(s)?

Are loops sufficient?

« 16 » More radically, we could even wonder whether the notion of a loop is actually consistent with the cognitive phenomena? This is the credo of dynamic systems theory. For instance Tim van Gelder (1997) argues in favor of abandoning loops, by comparing the Turing machine and the Watt governor. He argues that we should preferably get inspiration from the latter for understanding and/or modeling cognition. He points out that the Turing machine a-contextually manipulates symbols, obeying rules given by the programmer, while the Watt governor only obeys physical laws applied to built-in constraints. In other words, he stresses that the Turing machine relies on mathematics while the Watt governor relies on mechanics and physics. In this context, the notion of the loop is considered to be a hidden way of monitoring the need to do something. But sensing and acting do not consist of waiting for something to do. Rather, cognition must be considered as constituted of continuous processes that are fundamentally proactive, compared to the passive waiting character of looping.

« 17 » The aspects covered in my commentary represent only a fragment of a deeper ongoing debate. This is probably where the frontiers are situated of what *in silico* simulation can offer to help build constructivist foundations of socio-cognitive understanding. So far, computational methods may lead us to understand complexity growth by using simulation. In the meantime, however, we must ask ourselves how these purely deterministic sequential methods can really help us simulate what we, as adaptive animals, are.

Fabien Hervouet is currently completing his Ph.D. in artificial intelligence at the University of Montpellier (France). His research first dealt with multi-agent architectures and simulations for language emergence and spreading. For the last three years he has been interested in cognitive sciences, intrinsic motivations for sensory-motor learning in developmental robotics, and enactive artificial intelligence and its philosophical and epistemological implications.

Received: 20 February 2014
Accepted: 25 February 2014

The Relevance of "Differentiation" and "Binary Code" for Simulating Luhmann

Gastón Becerra
Universidad de Buenos Aires, Argentina
gastonbecerra/at/sociales.uba.ar

> **Upshot** • I acknowledge the value of Porr & Di Prodi's piece for simulating Luhmann's key process of subsystem formation and exploring how the concepts of "differentiation" and "binary code" relate to their model.

« 1 » Bernd Porr & Paolo Di Prodi's target article does a great job of showing how the process of "subsystem formation," central to Niklas Luhmann's sociology, can be simulated. To assess the value of Porr & Di Prodi's work, we must first make explicit our interests. For this commentary, I take a sociological point of view. What interests me the most is how this simulation sheds some light on the intricate levels underlying the process of subsystem formation and how they relate to key concepts taken from Luhmann's work.

« 2 » The first level corresponds to neuronal systems with interrelated signals processed by agents. This model is based on the guidelines of radical constructivism, and although it could be said that Luhmann differs from RC in several aspects, Porr & Di Prodi's model seems to me in line with Luhmann's cognitive and systemic postulates. Systems' self-referent operations driving the learning process are stimulated by irritations (internal states) (Luhmann 2006: 627).

« 3 » The second level is perhaps even more important from a sociological perspective. It corresponds to a social system that is drawn upon the interaction of agents' behavior. To talk about communicative systems, at least in the way I understand Luhmann's model, it is not enough that agents are able to broadcast signals (§62). Contingency is the link between the two levels. Contingency arises when an agent faces another agent whose behavior cannot be predicted but only expected. Alterity (otherness) is the heart of a sociological perspective. Luhmann depicted double contingency as two black boxes that deal with one another under the constraints of their own capacity to observe and influence (Luhmann 1991: 118f). The authors suggest that double contingency raised by this interaction could motivate an agent to select a particular aspect of its environment as a complexity reduction strategy (§61). The agents then could specialize into a role or particular behavior (§57) by triggering a modification in their neuronal structures. The agents are not individually preprogrammed to adopt one particular role. They are just programmed to adapt their behaviors under the conditions of their environment (§64), which is only possible because they have been first programmed to observe the environment through distinctions (e.g., food sources). The emergent order (social system) is conditioned by the complexity of the individual systems, without depending on their capacity to coordinate or control it (Luhmann 1991: 119).

« 4 » Does this process illustrate the emergence of social subsystems in accordance with Luhmann's theory? To answer this question, I think two concepts should be analyzed: "differentiation" and "binary code." The authors clearly take this direction in their conclusion. And I agree with their remarks, as pointed out in what follows.

« 5 » According to Luhmann (2006), the process of system differentiation can be triggered spontaneously as a result of evolution and can induce structural transformations. Differentiation takes place when a system/environment distinction is drawn within an existing system; the latter is seen as global in the eyes of the just-differentiated system (Luhmann 1991: 42). The authors illustrate the simplest and most trivial form of this differentiation, namely, one that carries no reference to the society as a whole and produces no subsequent formations within the system.

« 6 » A binary code can be established when the system is able to distinguish-and-select without losing the reference that created the distinction in the first place. Then there is always the possibility to turn to a (preexisting) negative value. This type of selection differs from basal operations that can only refer to identical elements and are blind to what is different (Esposito 1996). The distinction driven by binary code does not correspond to difference between system and environment but it refers to internal, accessible and contingent states of the system that duplicate the reality. A communication can accept or deny a certain value (e.g., a paper suggesting that an entire theory is false) but it cannot deny the importance of this distinction (Luhmann 1996: 222). In my opinion, this is the most substantial part of Luhmann's sociological work: it enables the treatment of a multi-contextural significant construction of reality within modern society.

« 7 » Porr & Di Prodi's model for simulation looks to be far from the conditions for generating a binary code. The systems they describe rely more on actions than on communications, and have no place for understanding. Without communication and language, is not that clear how acceptance/denial could play a part in this model (Luhmann 2006: 170). However, having a binary code is not a *sine qua non* condition for illustrating how social subsystems emerge. It is important to keep this in mind to avoid raising false expectations while evaluating the authors' contribution and its relevance to sociology. They are not simulating the emergence of functional systems, but showing that double contingency can lead to subsystem differentiation at the basal operations level.

Gastón Becerra is an assistant professor at the Universidad de Buenos Aires. He is pursuing a Ph.D. on philosophy of science and epistemology.

Received: 20 February 2014
Accepted: 22 February 2014

Authors' Response

What to Do Next: Applying Flexible Learning Algorithms to Develop Constructivist Communication

Bernd Porr
University of Glasgow, UK
bernd.porr/at/glasgow.ac.uk

Paolo Di Prodi
University of Glasgow, UK
robomotic/at/gmail.com

> **Upshot** • We acknowledge that our model can be implemented with different reinforcement learning algorithms. Subsystem formation has been successfully demonstrated on the basal level, and in order to show full subsystem formation in the communication system at least both intentional utterances and acceptance/rejection need to be implemented. The comments about intrinsic vs extrinsic rewards made clear that this distinction is not helpful in the context of the constructivist paradigm but rather needs to be replaced by a critical reflection on whether one has truly created autopoietic agents or just an engineering system.

« 1 » **Olivier Georgeon**'s commentary shows that there is a deep mistrust between two communities: on the one hand the reinforcement community, which has emerged from both engineering (e.g., optimal control) and economics, and on the other hand the community that has emerged from behaviour-based robotics, now branding itself as "enactive cognition." As **Patrick Pilarski** (§6) rightfully emphasized, the trouble comes from introducing the notion of reward. While in reinforcement learning the reward plays a central role, in enactive cognition it is very fashionable to claim that rewards are not part of the equation (**Georgeon** §9). However, rewards are then introduced through the back door in the form of "enjoyment," "dislike" or "intrinsic satisfaction." Even "sensor loops" (**Fabien Hervouet** §15), which learn to detect novelty, essentially reward themselves to be able to predict sensor states. As pointed out by Pilarski throughout his commentary, the boundary between intrinsic and extrinsic reward is blurred (§4) and rewards are and always will play a role in an experimental design in a more or less explicit manner. For us as authors writing a paper, using a reinforcement learning algorithm is treading on thin ice, being essentially halfway in the one camp and halfway in the other. However, bringing these two opposing views back together is probably easier than it appears to be (§13): the enactive cognition community needs to accept that rewards are part of the equation, for example as average rewards, and that it is politically correct to call them rewards (§7).

« 2 » At the same time, the reinforcement learning community needs to reflect on why the constructivist community and, more generally, the cognitive robotics community has such reservations against any form of rewards. This brings us back to its roots, namely engineering and economics. It comes as no surprise that engineering in particular is focused on results, namely *outputs*. However, from a constructivist perspective, agents control their *inputs,* which is the opposite of what an engineer wants to achieve. In Porr & Wörgötter (2005), we called this "the second chicken/egg problem": while the farmer wants to have the egg, the chicken wants to keep it. In other words, it is not so much an issue that we call a reward a "reward" or "satisfaction" but rather a problem of perspective. For example, a car should drive a passenger from A to B and not suddenly decide to make a detour to the cinema. However, for an autonomous agent that would be alright as long as it maintains its autopoiesis, which translates into the requirement that rewards need to be beneficial for the agent and not for the designer. This means that the use of the "R"-word in enactive cognition should be perfectly reasonable as long as the rewards help to maintain the agents' autopoiesis, usually through learning, as explained next.

« 3 » Based on these general remarks about the concept of reward, we can re-visit ICO learning, which has been used in this article. It is a biologically-inspired machine learning algorithm and belongs to the class of reinforcement learners where the actor and critic have been merged into one unit (Wörgötter & Porr 2005). We used it because of its very fast convergence, which shortened the execution of experiments considerably – at the expense of flexibility. Clearly, other learning algorithms of the class of reinforcement learning (Pilarski §10; Georgeon §11; Hervouet §4) would allow more flexible loops, the creation of new loops and also the generation of long-range predictions (Pilarski §8, §12; Hervouet, §§6f). This also addresses the criticism by Hervouet (§§6f, 13-15), who pointed out that ICO learning does not start off from random loops. This is certainly possible in any actor/critic scenario where the actor is allowed the freedom to create/remove any sensor/motor loop so that emergence is possible, in particular when employing the architectures suggested by Pilarski. We are in complete agreement that the learning suggested in his thought experiment (§§11f) would certainly lead to subsystem formation and would allow much more sophisticated behaviours. We also agree that an active exploration mechanism would greatly improve the agent's learning (Hervouet §9); however, this has been already discussed in our target article. In any case, all these alternative algorithms would have drastically increased the complexity of the experiment, which we wanted to keep as simple as possible to make our point about both the role of disturbances and subsystem formation.

« 4 » Hervouet (§16) pointed out that loops should become proactive, which is achieved with ICO learning and also with other reinforcement learning algorithms. However, it is debatable that an agent needs to become proactive at all to survive. One could argue that reactive behaviour is sufficient as long as it has the requisite variety and that this could be improved (Nakanishi & Schaal 2004). For example, a rabbit will always just run away from a fox but will never try to poison the fox in a proactive way to prevent further attacks. So being proactive is a huge evolutionary advantage and for that reason is used in our paper. However, one could argue that even humans could get by with being largely reactive.

« 5 » Hervouet (§§16f) also comments on the actual physical instantiation of the system. Do we need a physical system such as the Watt governor because its digital implementation in silico would not be appropriate? Any standard textbook on digital signal processing can easily provide an answer to that. The Watt governor is essentially a linear control system and performs its computations mechanically. A computer processes numbers that, in the case of navigating a real agent, arise from quantising analogue values, performing computations and transforming them back into analogue motor outputs. However, as long as the quantisation is smaller than the noise in the sensors (and the actuators), the digital processing will not change the dynamics of the system. It was shown *analytically* that one can perfectly replace parts of the Watt Governor with a digital controller without altering the dynamics of the system at all. Again, the main problem is not the implementation. This brings us back to the thorny issue of who defines the "reward" functions. Looking into the mechanical computations the Watt governor performs, it can easily be seen that it can also be described in terms of rewards because it has a setpoint (i.e., target value). For virtually all organisms, including humans, rewards have been provided by evolution (see the next paragraph). For artificial agents, we either need to simulate evolution or we can be inspired by the neural networks of animals and observe/measure how they have managed to stay alive in terms of rewards and punishments.

« 6 » Looking at real brains, primary rewards are usually coded, no matter if mouse or human brains, in the lateral hypothalamus (Nakamura & Ono 1986). These reward signals are then processed in the limbic system, where the most famous signal is the reward prediction error encoded by dopaminergic activity (Schultz 1998). More recently, it has been discovered that dopamine has two modes of operations: a phasic prediction error and slow tonic changes, where the latter seems to be encoding average rewards (Niv 2007). However, while the average reward plays an important role in the limbic (i.e., emotional) system, higher-level (conscious) decision making, especially in the basal ganglia, might operate reward-free and is probably driven by habits and/or novelty (Redgrave, Gurney & Reynolds 2008; Hervouet §9). This shows again that one needs to be careful when dealing with rewards: primary rewards certainly play an important role but at the same time one needs to acknowledge that they are not the sole drivers of action selection.

« 7 » Regardless of whether it is implemented in the traditional formulation as TD learning or in the advanced versions using average rewards, reinforcement learning employs closed loop processing. In his OPC, Georgeon stresses the fact that learning needs to be closed loop (§§1f) and that this kind of learning creates new loops, which he calls "schemas" in the spirit of Piaget. Our target article acknowledges the fact that ICO learning, though limited, is able to create new loops. However, one needs to be clear about why agents employ closed loop learning. Certainly, it has not been implemented to create "interesting" behaviours (Georgeon §§4-6). Such a requirement would create two problems.

1 | The first problem arises from the fact that the experimenter essentially performs output control by evaluating the agent's behaviour and then concludes that closed loop control should be preferred over open loop because it creates more "interesting" behaviour. However, while the agent might provide an output that is "interesting" for the observer, it might not be very beneficial for its own survival. For example, in the worst case, an external observer finds it highly "interesting" to observe an animal starving to death.

2 | To make things worse, Georgeon's writing suggests that one can freely choose between open and closed loop control when modelling autonomous agents. However, as outlined in the previous paragraphs, closed loop processing is not just a nice feature for creating "interesting" behaviours but it is fundamental for the constructivist paradigm because it performs input control, which guarantees that an agent can maintain its autopoiesis.

« 8 » Having now discussed input and output control extensively and concluded that agents perform input control, one needs to be careful when stating that the agent cannot access the environment's state (Georgeon §3). Again, it is a matter of perspective: experimenter vs agent. If one could ask the agent: "Are you aware of all of the environment's states, agent?," it would say: "Yes, totally." From the agent's perspective, its loops that act against the corresponding disturbances represent its *entire* environment, and the loops and disturbances represent the agent's "reality" of the world (Georgeon §7). Only when we observe the agent from the outside do we realise that its environment is infinite and thus the number of both accessible and hidden states is infinite as well. To have all states of the environment finite and observable to the agent can only be created artificially in very simple environments, which is not the case for actual organisms. For artificial agents this can be overcome – for example, by using embodied agents that have to deal with a complex environment.

« 9 » We are grateful that Hervouet comments on the environment (§§11f) and that altering it will impact on the subsystem formation. Indeed, a changing environment over time should cause a re-organisation of the subsystems and thus the predictive value readings of the individual agents. It would be an interesting follow-up experiment to change the number of food sources in a dynamic way. In a more general context, this also means that the makeup of the environment has a strong impact on the agent because (a) the disturbances force the agent to create appropriate loops and (b) these loops need to be operational with sufficient requisite variety.

« 10 » Finally, Gastón Becerra's excellent commentary is more an outlook than a criticism. It points out exactly how to develop our model further. Double contingency in our model is only established on the level of behaviour, where agents aim to predict each other. However, as already outlined in the target article, the level of communication has only been partially implemented. So far, our agents broadcast their states into the world continuously (Becerra §3), which leads to complete predictability in terms of the amount of food an agent carries. In a more advanced version of our model, the agents should be able to decide whether they want to release their states to the world in the form of utterances and to which agent. This may sound simple, but it requires loops and learning algorithms on the level of utterances and not just on the level of actions. In other words, we need to have both "motor" loops that generate actions (as done in our model) and loops that control the release of utterances. By creating separate loops for actions and utterances, we effectively create two subsystems. Consequently, the question arises of how they interpenetrate each other (to use Luhmann's words). The communication system would have an impact on the action system and vice versa. To enable a rich selection of utterances, one needs to allow the agent to release not only internal states but also sensor information and how sensor states lead to behaviour (by using weights, etc.).

« 11 » So far, we have just dealt with the release of utterances, which means that we have covered ego but not alter. In particular, alter observes an utterance by ego and can integrate it into its loops or just ignore it. However, this requires ultimately second-order cybernetics in the communication system so that agents can accept or reject certain communications (Becerra §§6f). This would then lead to a binary code that could accept or reject an utterance. As such, it is part of our on-going research.

Received: 3 March 2014
Accepted: 5 March 2014

Combined References

Ashby W. R. (1956) An introduction to cybernetics. Methnen, London.

Barber M., Blanchard P., Buchinger E., Cessac B. & Streit L. (2006) Expectation-driven interaction: A model based on Luhmann's contingency approach. Journal of Artificial Societies and Social Simulation 9(4). Available at http://jasss.soc.surrey.ac.uk/9/4/5.html

Beekman M., Nicolis S., Meyer B. & Dussutour A. (2009) Noise improves collective decision-making by ants in dynamic environments. Proceedings of the Royal Society London B 276: 4353–4361.

Braitenberg V. (1984) Vehicles: Experiments in synthetic psychology. MIT Press, Cambridge MA.

Camazine S., Franks N. R., Sneyd J., Bonabeau E., Deneubourg J.-L. & Theraula G. (2001) Self-organization in biological systems. Princeton University Press, Princeton NJ.

Cangelosi A., Tikhanoff V., Fontanari J. & Hourdakis E. (2007) Integrating language and cognition: A cognitive robotics approach. IEEE Computational Intelligence Magazine 2(3): 65–70.

Degris T., Pilarski P. M. & Sutton R. S. (2012) Model-free reinforcement learning with continuous action in practice. In: Proceedings of the 2012 American Control Conference, 27–29 June 2012, Montreal, Canada. IEEE Press, Piscataway NJ: 2177–2182.

Di Paolo E. A. (2005) Autopoiesis, adaptivity, teleology, agency. Phenomenology and the Cognitive Sciences 4(4): 429–452.

Dittrich P., Kron T. & Banzhaf W. (2003) On the scalability of social order. Modeling the problem of double and multi contingency following Luhmann. Journal of Artificial Societies and Social Simulation 6(1). Available at http://jasss.soc.surrey.ac.uk/6/1/3.html

Drescher G. L. (1991) Made-up minds: A constructivist approach to artificial intelligence. MIT Press, Cambridge MA.

Emmeche C., Koppe S. & Stjernfelt F. (2000) Levels, emergence, and three versions of downward causation. In: Andersen P. B., Emmeche C., Finnemann N. O. & Christiansen P. V. (eds.) Downward causation. University of Aarhus Press, Århus DK: 13–34.

Esposito E. (1996) From self-reference to autology: How to operationalize a circular approach. Social Science Information 35(2): 269–281.

Foerster H. von (2003) Understanding understanding: Essays on cybernetics and cognition. Spinger, New York.

Franks N. R., Mallon E. B., Bray H. E., Hamilton M. J. & Mischler T. C. (2003) Strategies for choosing between alternatives with different attributes: Exemplified by house-hunting ants. Animal Behaviour 65(1): 215–223.

Froese T. & Ziemke T. (2009) Enactive artificial intelligence: Investigating the systemic organization of life and mind. Artificial Intelligence 173(3–4): 466–500.

Gadenne V. (2010) Why radical constructivism has not become a paradigm? Constructivist Foundations 6(1): 77–83. Available at http://www.univie.ac.at/constructivism/journal/6/1/077.gadenne

Georgeon O. & Hassas S. (2013) Single agents can be constructivist too. Constructivist Foundations 9(1): 40–42. Available at http://www.univie.ac.at/constructivism/journal/9/1/040.georgeon

Georgeon O. & Marshall J. (2013) Demonstrating sensemaking emergence in artificial agents: A method and an example. International Journal of Machine Consciousness 5(2): 131–144.

Georgeon O., Marshall J. & Manzotti R. (2013) ECA: An enactivist cognitive architecture based on sensorimotor modeling. Biologically Inspired Cognitive Architectures 6: 46–57.

Georgeon O. & Sakellariou I. (2012) Designing environment-agnostic agents. In: Howley E., Vrancx P. & Knudson M. (eds.) Proceedings of the Adaptive and Learning Agents Workshop, 4–5 June 2012, Valencia, Spain: 25–32. Available at http://ai.vub.ac.be/ALA2012/

Georgeon O., Wolf C. & Gay S. (2013) An enactive approach to autonomous agent and robot learning. Proceedings of the IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (EPIROB2013) Osaka, Japan. 1–6.

Glasersfeld E. von (1984) An introduction to radical constructivism. In: Watzlawick P. (ed.) The invented reality: How do we know what we believe we know? W. W. Norton, New York: 17–40. Available at http://www.vonglasersfeld.com/070.1

Grant C. B. (2002) Complexities of self and social communication. In: Grant C. B., (ed.) Radical communication: Rethinking interaction and dialogue. John Benjamins, Amsterdam: 101–125.

Grondman I., Busoniu L., Lopes G. A. D. & Babuska R. (2012) A survey of actor-critic reinforcement learning: Standard and natural policy gradients. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42(6): 1291–1307.

Heath B., Hill R. & Ciarallo F. (2009) A survey of agent-based modeling practices (January 1998 to July 2008) Journal of Artificial Societies and Social Simulation 12(4). Available at http://jasss.soc.surrey.ac.uk/12/4/9.html.

Hülse M. & Pasemann F. (2002) Dynamical neural Schmitt trigger for robot control. In: Dorronsoro J. R. (ed.) Artificial Neural Networks ICANN 2002: 783–788.

Jost J., Bertschinger N., Olbrich E., Ay N. & Fraenkel S. (2007) An information theoretic approach to system differentiation on the basis of statistical dependencies between subsystems. Physica A 378: 1–10.

Kaelbling L., Littman M. & Cassandra A. (1998) Planning and acting in partially observable stochastic domains. Artificial Intelligence 101: 99–134.

Kenny V. (2009) "There's nothing like the real thing." Revisiting the need for a third-order cybernetics. Constructivist Foundations 4(2): 100–111. Available at http://www.univie.ac.at/constructivism/journal/4/2/100.kenny

Kernbach S., Thenius R., Kornienko O. & Schmickl T. (2009) Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic swarm. Adaptive Behavior 17: 237–259.

Klopf A. H. (1982) The hedonistic neuron: A theory of memory, learning, and intelligence. Hemisphere, Washington DC.

Leydesdorff L. (2005) Anticipatory systems and the processing of meaning: A simulation inspired by Luhmann's theory of social systems. Journal of Artificial Societies and Social Simulation 8(2). Available at http://jasss.soc.surrey.ac.uk/8/2/7.html.

Linden D. J. (2003) From molecules to memory in the cerebellum. Science 301: 1682–1685.

Littman M. L., Sutton R. S. & Singh S. (2002) Predictive representations of state. In: Dietterich T. G., Becker S. & Ghahramani Z. (eds.) Advances in Neural Information Processing Systems 14. Proceedings of the 2001 Conference. MIT Press, Cambridge MA: 1555–1561.

Luhmann N. (1984) Soziale Systeme. Suhrkamp, Frankfurt am Main.

Luhmann N. (1991) Sistemas sociales. Lineamientos para una teoría general. Anthropos, Barcelona. German original published in 1985.

Luhmann N. (1996) Die Realität der Massenmedien. Westdeutscher Verlag, Opladen.

Luhmann N. (1996) La ciencia de la sociedad. Anthropos, México City. German original published in 1990.

Luhmann N. (2006) La sociedad de la sociedad. Herder, México City. German original published in 1997.

Matarić M. J. (1997) Reinforcement learning in the multi-robot domain. Autonomous Robots 4(1): 73–83.

Maturana H. R. & Varela F. J. (1980) Autopoiesis and cognition: The realization of the living. Reidel, Dordrecht.

McCallum A. (1996) Learning to use selective attention and short-term memory in sequential tasks. In: Maes P., Mataric M. J., Meyer J.-A., Pollack J. & Wilson S. W. (eds.) From animals to animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior. MIT Press, Cambridge MA: 315–324.

Meyer B., Beekman M. & Dussutour A. (2008) Noise-induced adaptive decision-making in ant-foraging. Lecture Notes in Computer Science 5040: 415–425.

Modayil J., White A. & Sutton R. S. (2014) Multi-timescale nexting in a reinforcement learning robot. Adaptive Behavior, published online 7 February 2014.

Nakamura K. & Ono T. (1986) Lateral hypothalamus neuron involvement in integration of natural and artificial rewards and cue signals. Journal of Neurophysiology 55(1): 163–181.

Nakanishi J. & Schaal S. (2004) Feedback error learning and nonlinear adaptive control. Neural Networks 17(10): 1453–1465.

Niv Y. (2007) Cost, benefit, tonic, phasic: What do response rates tell us about dopamine and motivation? Annals of the New York Academy of Sciences 1104: 357–376.

O'Regan J. K., Myin E. & Noë A. (2005) Phenomenal consciousness explained (better) in terms of bodiliness and grabbiness. Phenomenology and the Cognitive Sciences 4(4): 369–387.

O'Regan J. K. & Noë A. (2001) A sensorimotor account of vision and visual consciousness. Behavioral and Brain Sciences 24(5): 939–1031.

Oudeyer P.-Y., Kaplan F. & Hafner V. (2007) Intrinsic motivation systems for autonomous mental development. IEEE Transactions on Evolutionary Computation 11(2): 265–286.

Parsons T. (1968) Social interaction. In: Sills D. L. (ed.) International encyclopedia of the social sciences. Volume 12. Macmillan/Free Press, New York: 429–440.

Pfeifer R. & Scheier C. (1994) From perception to action: The right direction? In: Gaussier P. & Nicoud J.-D. (eds.) From perception to action. IEEE Computer Society Press, Los Alamitos CA: 1–11.

Pilarski P. M., Dick T. B. & Sutton R. S. (2013) Real-time prediction learning for the simultaneous actuation of multiple prosthetic joints. In: Proceedings of the 2013 IEEE International Conference on Rehabilitation Robotics, Seattle, USA, 24–26 June 2013. IEEE Press, Piscataway NJ: 1–8.

Porr B., Egerton A. & Wörgötter F. (2006) Towards closed loop information: Predictive information. Constructivist Foundations 1(2): 83–90. Available at http://www.univie.ac.at/constructivism/journal/1/2/083.porr

Porr B., Ferber C. von & Wörgötter F. (2003) ISO-learning approximates a solution to the inverse-controller problem in an unsupervised behavioural paradigm. Neural Computation 15: 865–884.

Porr B. & Wörgötter F. (2002) Isotropic sequence order learning using a novel linear algorithm in a closed loop behavioural system. Biosystems 67(1–3):195–202.

Porr B. & Wörgötter F. (2003) Isotropic sequence order learning. Neural Computation 15: 831–864.

Porr B. & Wörgötter F. (2005) What means embodiment for radical constructivists? Kybernetes 34(1/2): 105–117.

Porr B. & Wörgötter F. (2006) Strongly improved stability and faster convergence of temporal sequence learning by utilising input correlations only. Neural Computation 18(6): 1380–1412.

Reading N. C. & Sperandio V. (2006) Quorum sensing: The many languages of bacteria. FEMS Microbiology Letters 254(1): 1–11.

Redgrave P., Gurney K. & Reynolds J. (2008) What is reinforced by phasic dopamine signals? Brain research reviews 58(2): 322–339.

Redish A. D. (2013) The mind within the brain: How we make decisions and how those decisions go wrong. Oxford University Press, New York.

Riegler A. (2007) The radical constructivist dynamics of cognition. In: Wallace B. (ed.) The mind, the body and the world: Psychology after cognitivism? Imprint, London: 91–115. Available at http://www.univie.ac.at/constructivism/riegler/44

Roesch E., Spencer M., Nasuto S., Tanay T. & Bishop J.-M. (2013) Exploration of the functional properties of interaction: Computer models and pointers for theory. Constructivist Foundations 9(1): 26–32. Available at http://www.univie.ac.at/constructivism/journal/9/1/026.roesch

Salgado M. & Gilbert N. (2008) Emergence and communication: Overcoming some epistemological drawbacks in computational sociology. In: Proceedings of the Third Edition of Epistemological Perspectives on Simulation. Lisbon: 105–124.

Schmidhuber J. (1991) Curious model-building control systems. In: Proceedings of the International Joint Conference on Neural Networks, Singapore, Volume 2. IEEE Press, Piscataway NJ: 1458–1463.

Schultz W. (1998) Predictive reward signal of dopamine neurons. Journal of Neurophysiology 80: 1–27.

Singh S., Barto A. & Chentanez N. (2005) Intrinsically motivated reinforcement learning. In: Saul L. K., Weiss Y. & Bottou L. (eds.) Advances in Neural Information Processing Systems. MIT Press, Cambridge MA: 1281–1288.

Steels L. (2004) The autotelic principle. In: Fumiya I., Pfeifer R., Steels L. & Kunyoshi K. (eds.) Embodied artificial intelligence. Lecture Notes in AI 3139. Springer, Berlin: 231–242.

Sutton R. S. & Barto A. G. (1998) Reinforcement learning: An introduction. MIT Press, Cambridge MA.

Sutton R. S., Modayil J., Delp M., Degris T., Pilarski P. M., White A. & Precup D. (2011) Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In: Tumer K., Yolum P., Sonenberg L. & Stone P. (eds.) Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems, 2–6 May 2011, Taipei, Taiwan. International Foundation for Autonomous Agents and Multiagent Systems, Richland SC: 761–768.

Tani J. & Nolfi S. (1999) Learning to perceive the world as articulated: An approach for hierarchical learning in sensory-motor systems. Neural Networks 12: 1131–1141.

Touchette H. & Lloyd S. (2004) Information-theoretic approach to the study of control systems. Physica A Statistical Mechanics and its Applications, 331: 140–172.

van Gelder T. (1997) Dynamics and cognition. In: Haugland J. (ed.) Mind design II: Philosophy, psychology, artificial intelligence. MIT Press, Cambridge MA: 421–450.

Varela F. J. (1989) Autonomie et connaissance: Essai sur le vivant. Edition Seuil, Paris.

Varela F. J., Thompson E. & Rosch E. (1991) The embodied mind: Cognitive science and human experience. MIT Press, Cambridge MA.

Weisbuch G. & Stauffer D. (2000) Hits and flops dynamics. Working Papers 00–07–036. Santa Fe Institute, Santa Fe NM.

Whitehead S. D. & Ballard D. H. (1991) Learning to perceive and act by trial and error. Machine Learning 7(1): 45–83.

Wischman S., Pasemann F. & Hülse M. (2004) Structure and function of evolved neurocontrollers for autonomous robots. Connections Science 16(4): 249–266.

Wörgötter F. & Porr B. (2005) Temporal sequence learning, prediction and control. A review of different models and their relation to biological mechanisms. Neural Computation 17(2): 245–319.