universität
wien

MASTERARBEIT

Making Sense

An Integrative View on Understanding Action and Observation

Verfasser

Raphael Deimel, Bachelor of Science

angestrebter akademischer Grad

Master of Science (MSc)

Wien, 2011

Studienkennzahl lt. Studienblatt:	A0425354
Studienrichtung lt. Studienblatt:	Middle European interdisciplinary Master in Cognitive Science
Betreuerin / Betreuer:	a.o. Prof. Dr. Igor Farkas

Contents

Acknowledgements

I want to thank all the visible organizers and invisible backers of the *Middle European interdisciplinary Master in Cognitive Science* for creating such an adventurous and exciting study. The openness and support of the teachers, the kaleidoscope of fellow students, and the wealth of opportunities made it a memorable experience.

I especially want to thank Elisabeth, the administrative soul, for for her invaluable help with the intricacies of bureaucracy encountered, and Igor, for his constructive feedback to my ideas and giving me the opportunity to pursuit my personal interests.

I also want to thank my family, who supports me in my quest for knowledge, even though the destination is unknown and the path is anything but straight.

Thanks to my friends for countless hours of stimulating activities, adventures and challenges. Creativity and insights do not happen in a void.

And thank you Monika for filling my life with joy and love.

Foreword

“Tell me and I will forget.
Show me and I will remember.
Involve me and I will understand.
Step back and I will act.”

Chinese Proverb

How do humans intuitively tell the difference between the weights of two objects, e.g. a Rubick’s Cube, but their shape, colour and materials differ. I conducted this small ad hoc experiment in several locations, and also during my talk at the MEi:CogSci Conference 2010. It’s very interesting to see how similar most people try to solve it. But first, we’ll have a look at some possible hypotheses:

A researcher believing in classical AI will (should?) suspect, that humans perform a complicated form of object recognition, scale detection, assessment of probable material type, remembering it’s specific density and then inferring each object’s weight, compare it, and declare the truth value of the utterance “the cube is bigger than the ball” (or vice versa). After all, a picture of the two objects in question tells you everything you need to know, doesn’t it? Of course, there is a host of problems with this analytic approach, namely that it presupposes a vast and very specific knowledge base to infer the appropriate information from (for simply deciding on the weight of something!). It is also prone to small deviations and errors propagating through the logic inference. In light of this, one should easily see that this can hardly be a sensible description of what a human intuitively would do.

A typical practical engineering approach, on the other hand, would postulate a measuring device for the objects with a weight sensor (we generously assume the arm muscles and proprioceptive sensor system to resemble such a weight measurement device), and then computing the difference between the measured weights. According to a decision table, the engineer concludes then, whether the first object “is heavier”, “lighter” or “roughly the same”. Unfortunately, such an approach has a lot of flaws too, such as unreliable sensor data and especially the need for a dedicated, single-purpose device (and perhaps the need of a miraculous or evolutionary explanation for its very existence). This is the description of a dedicated special-purpose machine, but definitely not one of flexible human behaviour.

What usually happens when one gives people such a riddle to solve, is that they take the two objects into each hand and start moving them around, *to get a feel* for the weight difference.

They instantly start to perturb their environment (apply a certain force or acceleration pattern to the object), that gives them additional feedback for an increased accuracy of their judgement. This action is neither done consciously, nor is it planned. It is the skilled use of the *arms* and the accompanying sensor and motor circuits in the brain, that provide the illusion of an *immediate access* to the weights of the objects. There is no need for a complex planning process for a skilled task. Access to proximal features and process parameters of the environment are transparent. The weight in the hand is directly accessible, as if there was no complex system of motor cortex, spinal circuits, nerve fibres, muscles, tendons and arm kinematics in between.

How come we can access our bodies so easily? Why are all the sophisticated dynamics of the body transparent during performance of actions? What is the difference between using one's hand and using pliers? Is there any?

Introduction

This thesis tries to derive principles for an *autonomous cognitive apparatus* (being a physical part of an agent), to facilitate transparent access to features of the agent's physical body and environment. It regards action and observation to be aspects of fundamentally the same process that enables an agent to define its body. This process is shaped not only by the brain, but also by the sensorimotor contingencies of the somewhat arbitrary, "attached" environment.

This work also honors the agent's agency. By their very definition, they can act deliberately. Their actions are not necessarily predetermined by the environment, as supposed by Behaviourism. The prevalent view nowadays still is the one of the agent being like a clockwork (an algorithm) performing a certain computation according to some externally defined (e.g. evolutionary) goals. These oxymoronic "reactive agents" (reacting to environmental stimuli alone) already lost their agency by definition. Luckily though, this paradigm is being replaced in some research for an interactivist and Embodied Cognition perspective, where an agent can define itself by creating action opportunities, and being spread out over parts of the physical world (i.e. the embodiment of cognition), instead of being a separate mechanism.

A central part of this thesis is the understanding of the brain's environment in terms of sensorimotor contingencies. Much like the nature of a dynamical process is primarily defined by its fix points (attractors), the brain's environment can be described by its sensor invariances and contingencies with respect to effected action patterns, as proposed by David Philipona and Kevin O'Regan (Philipona *et al.* , 2003). These contingencies induce surfaces (subspaces) in the action-sensation space, and the agent can learn to deliberately move on these surfaces. These learnt surfaces of full control enable the agent to enact it's own laws and ideas there, much like using those surfaces as a scratch board to drool upon. We depart from the notion of a purely passive observer to an active agent changing it's environment.

Many ideas and explanations are formulated using a paradigm different to the one usually used in the hard sciences. Borrowing from philosophical Constructivism, some arguments are elaborated from an agent's point of view. This *first person perspective* can sensibly complement the usual, objective *third person perspective* of hard sciences (like Computer Science, or Psychology) in cases, where it is rendered incapable due to a self-referential observer. This is the case when the supposedly independent observer is the observed subject itself.

In the practical part, the theoretically derived model is put to the test in a series of exploratory simulations. Starting point are two well understood problems of Computer Science (Classification and Action Selection). Step by step, the models for the cognitive apparatus and it's environment are incrementally modified, until finally they implement the full *Cognitive Body* model derived in the theoretical work, able to learn to control and observe arbitrary Finite State Machines of limited size.

Chapter 1

An Overview of Popular Models of Perception

What Does Perception Mean?

Different scientific communities use the term *perception* in different ways. Especially in the realm of Computer Science and Artificial Intelligence Research (e.g.Russel & Norvig, 2002), perception is understood as the process of extracting *relevant* information from sensory input. The information is deemed relevant with respect to certain goals, and often are interpreted to represent certain features of the world, e.g. a class distinction between edible objects and inedible ones. This view is historically inherited by Information Theory, as it directly relates to the "input-processing-output" nature of algorithms, and thus advances in algorithms can be directly applied to problems of perception (or vice versa). An important aspect of this perspective is, that perception is conceptualized as a directed graph where vertices represent algorithms, and edges are the data flows between them. Often, these graphs are additionally structured into hierarchical layers. Even the popular but unconventional Subsumption Architecture (Brooks, 1991) can be fit to this explanation of perception - the main difference being its bottom-up parallel functional division instead of a hierarchical one.

Nevertheless, it is wrong to generalise this historically entrenched view to all contemporary research, as there are a growing number of people that question this simplistic view on perception (Clark, 1996; Noe, 2009; Philipona *et al.* , 2004). Also, there is a school of philosophers called constructivists (e.g. Heinz von Förster; von Förster & Pörksen, 1997), that propose a radically different approach to perception, with interactive loops as their basic concept. Here, the perceived environment is constructed by the brain, thus perception is not a process of "passively accessing" a reality, but rather perception means using a previously established understanding of the agent's interactions with said reality.

1.1 The Passive, Objective Observer

The most popular framework for understanding perception is undoubtedly the one of a passive, objective observer. This paradigm is deeply rooted in decades of Metaphysics and Epistemology regarding the way proper Science should be (and is) conducted. In any *hard* Science, the observer (by its very definition) does not interfere with the observed phenomenon. This is actually one of the central dogmas for conducting Science. There are certain distinct advantages to separate the observer, namely the observations can be reused by other observers *as if* they were done by themselves, enabling to share and merge observations to come to a common observer-independent (objective) conclusion. This objectivity also facilitates a rapid knowledge uptake (via teaching) and reliable conservation (e.g. via books or recordings). An illustration of this view on perception is shown in Figure 1.1.1.

Supposing the understanding of an agent as a passive, objective observer, most researchers equate perception to be synonymous to observation.

Understanding perception as an internal process of an observer necessitates a passive nature. Perceiving does not change the perceived. Therefore, the only causal dependence goes from the environment (cause) to the agent (effect in the observer).

Incidentally, this view is especially prevalent within the field of Computer Science, e.g. in Computer Vision (Marr *et al.* , 1980), but also in Neuroscience. Often, Researchers see perception simply as the process of extracting knowledge from the sensor's stream of information. Fundamental problems like the *Chinese Room Argument* (Searle, 1990), and *Symbol Grounding* (Harnad, 1990) have to nag the researcher, lest it simply is ignored.

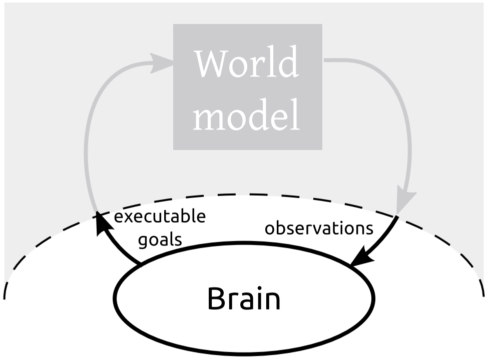


Figure 1.1.1: View of perception by representation-focused classical Artificial Intelligence. The objective world model is presumed.

1.2 Evolutionary and Other Developmental Adaptions

A different paradigm of perception is often employed by researchers with a background in Biology, Developmental Psychology and Robotics. Perception again is a passive process of identification, but the actual method to do so is either phylogenetically evolved or ontogenetically learned by a teacher (or by trial and error) using specially crafted reward signals. The actual internal structure of the agent is unimportant, only its effects to the environment (behavioural reaction to applied stimulus).

From an explanatory point of view, this is not very satisfying, as a reward signal has to be present for all adaptations, and the number of possible implementations for a certain behaviour prohibits predictions to genuinely novel stimuli. This does not make it a very practical framework for all but the simpler problems and organisms. It especially does not explain the wealth of structure in the brains of agents, when simple learning algorithms would suffice for associating stimuli and rewards.

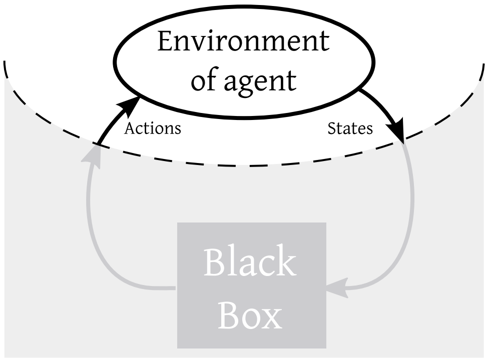


Figure 1.2.1: Behaviourism influenced view on perception. The agent's behaviour is predetermined by the environment, and thus its structure is unimportant.

1.3 Utility Focused Approaches

A more recent class of approaches are utility-centric. According to those, perception is a means to achieve certain goals of the agent in the environment.

Reinforcement Learning

An important example of utility-centric approaches comes from Machine Learning. The goal of Reinforcement Learning is to find an (optimal) policy for achieving certain tasks. The unique property here is, that it's not the sensory information, that gets processed, but the motoric information, acting on the world. The sensory information merely acts as a feedback channel, informing the Reinforcement Learning algorithm about the utility of it's performed action. The algorithm then constructs a (usually probabilistic) model of expected utility, such that it can compute the optimal policy to reach a goal, given such a probabilistic model.

Reinforcement Learning still relies on the existence of a predetermined sensory apparatus, that can at least sense the relevant states of the environment. It is often seen as a necessary preprocessing step to condense the wealth of sensory data into a manageable number of environmental dimensions or states.

An important difference to other algorithms in Machine Learning is, that the Reinforcement Learning idea places the learning agent into the position of choosing the next action. Thus, the agent's environment is not the sole source of causes.

Dynamic Systems

Dynamical Systems approaches have a long history in explaining coupled, recurrent systems, though their explanatory value often breaks down due to the high dimensionality of real system's phase spaces. One interesting approach is pursued by Karl Friston, called the *Free-Energy Principle* (Friston, 2010; Friston *et al.* , 2009). The idea is, that an agent tries to minimize the *Free Energy* measure of its bodily states, for the simple fact that it has to try to resist disorder (entropy) to perpetuate its structural identity. This equates to the agent trying to not be surprised by the environment, by choosing actions with better known outcomes and making better sensory predictions.

Embodied Cognition and Extended Mind

According to *Embodied Cognition*, perception is chiefly facilitated by the specific physical form of the agent's body, and may also include neural signal paths. Popular works like those of Rodney Brooks (Brooks, 1991) build upon this notion. It relies on evolution (and engineering), as *Embodied Cognition* draws its power from a cleverly constructed body, that already contains the necessary form to perform the intended way. Perception therefore is a property of the physical body, and not of the brain alone.

Embodied Cognition forces the scientist to think about the part, that often gets neglected by other paradigms, the physical body of the agent. It is seen as implementing parts of the intelligence of an agent ("embodying" it). This approach is illustrated in Figure 1.3.1.

Building upon Embodied Cognition is the field of Developmental Robotics, dealing with the problem of making sophisticated use of a pre-existing but unknown body by an autonomous robot. Borrowing ideas from developmental psychology (Smith & Gasser, 2005), the body can also change over time to facilitate learning objectives. The premise is, that the mind (in a classical AI sense, i.e. its information-processing structure) of the autonomous agent needs to co-evolve (epigenesis) with the body. Often, unsupervised learning methods are employed, or the problem is formulated as a search for mathematical structures, i.e. detecting manifolds in high-dimensional spaces, that limit possibilities, or express invariances (Philipona *et al.* , 2004).

Closely related to *Embodied Cognition* is the approach of the *Extended Mind* (Noe, 2009; O'Regan & Noë, 2001; Clark, 2008, 1996). According to this idea, Cognition is neither confined to the brain nor the agent's body. It can extend into the agent's environment, e.g. by manipulating artefacts (like using a sheet of paper and pencil to perform a complicated multiplication), or by externalising information (chemical trails of ants). It calls into question the strict physical boundaries of agency, be it the brain-body or the body-environment boundary.

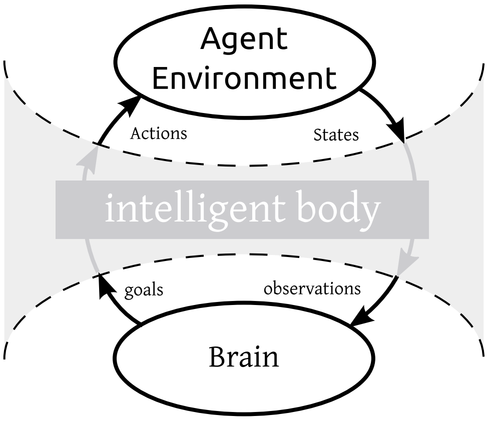


Figure 1.3.1: A holistic view on perception, encompassing both brain and environment.

Both approaches take a utilitarian, holistic approach to perception. It is the tool that enables an agent to enact its agency and separate itself from the environment. Perception is viewed as a dynamic interaction between the brain and its environment, (usually) facilitated by its body. For *Embodied Cognition*, this is the physical body of the agent, while for *Extended Mind*, it is the part of the environment, that is (currently) interacting with the brain, or even is under direct control.

The thesis was built upon the ideas of *Embodied Cognition* and *Extended Mind*,and can be related to Developmental Robotics. But it also utilizes classical symbolic explanations where deemed appropriate. The presented Cognitive Body Model builds its key design upon the philosophy of *Constructivism*, *Embodied Cognition* and *Extended Mind*, but the author also tries to relate as many components as possible to established knowledge and explanations in the domain of *Computer Science* and *Information Theory*, to show that this approach is not competitive, but complementary to the one presuming a passive, objective observer.

Chapter 2

The Cognitive Body Model

The analysis of approaches to perception in the previous chapter highlights the need for understanding the “glue” between the agent’s external environment, and its cognitive realm. In this chapter, the author proposes a genuine model for this glue, termed the *Cognitive Body model* [1] , whose inception was guided by the following questions:

- How can we use seemingly controversial paradigms (i.e. Constructivism, Autopoiesis) for complementing explanations in the research fields of perception and developmental robotics?
- Can we formulate a method to autonomously learn one’s body structure and even opportunistic tool use?
- Is there a way to bridge the divide between an autopoietic thought process and the surrounding, inherently inaccessible, environment?
- Can we formalize the processes responsible for constructing a self image of the body?

2.1 Causes, Effects and States as Projections of the Mind

Note, that in the previous section, the distinction between an agent’s body and its immediate environment was made on a quite arbitrary (albeit very useful) definition of a cell membrane or a skin. The boundary could also be defined on a histological level, e.g. around the central nervous system. This “brain in the vat” agency is a popular conception of western culture, and also deeply rooted in the self image of classical mid-20th century Artificial Intelligence research. This philosophic stance can be traced back to the works of Immanuel Kant and René Descartes. The AI paradigm implicitly claims that the mind is located in, or enacted by, the brain, and can be studied in isolation of the rest (the physical body, it’s environment). The problem we face with this paradigm is the classical *Homunculus fallacy*, i.e. that we try to explain an intelligent agent (e.g. a human individual) by postulating another “smaller” intelligent agent (the brain) controlling the surrounding mechanistic, clock-like machinery. This explanation can be iterated indefinitely without *actually* explaining the remaining homunculus itself. Daniell Dennett argues, that such a homunculus simply does not exist at all. It is a false belief, that can be uncovered by certain effects like change blindness (Dennett, 2004). An alternative proposition is promoted by Alva Noë (Noe, 2009). The *Extended Mind* does not stop at the agents brain, and is not a property of an object (body). Rather it is to be understood

[1] The term *Cognitive Body model* was chosen to reflect the focus on learning a control structure for the agent’s physical body. As this work shows in Section 2.9, the *Cognitive Body* is a more flexible concept, and need not be of identical (or even similar) extent as the physical body. Therefore both concepts must explicitly be distinguished. The proposed model has no direct connection to the likewise named *Cognitive Body* in the paper of Montebelli *et al.* , 2009.

as a process of interaction with, and thus extending into the environment. Noë uses the metaphor of the mind being the *dance of the agent with its environment.* There is no use in separating the two dancers into the agents mind and the agents subjective environment and trying to understand their movements separately.

At this point, I want to introduce the term *Umwelt* for a subjectively constructed environment, as formulated by Jakob Johann von Uexküll (von Uexküll, 1934). According to him, *the Umwelt* is that part of the complex and vast universe, which the agent is sensible for, and interacting with. Uexküll brings the example of a tick hunting for blood, whose Umwelt is extremely simple - comprising of things like a signal for presence of butyric acid molecules (sweat smell), a heat sense and simple feet, but is totally oblivious to what intricate reality (a forest's trees, a sweating deer passing by, its own cell machinery...) actually produces those interaction patterns. All the tick cares for, is its *Umwelt.*

As used in this thesis, *Umwelt* describes the subjective structure of the world the agent experiences. It is not to be mistaken for the objective, real *environment* of an agent. The structure of the *Umwelt* is constrained by the environment, but the *Umwelt* can likewise be restricted by the agent, through promoting or avoiding certain interaction patterns.

Coming back to Noë's proposition of an *Extended Mind*, the term *Umwelt* is very useful. The mind can be interpreted as a complex (but closed) dynamic system comprised of the *Umwelt* on one hand, and a *cognitive apparatus* (i.e. the abstract information processing capability of a human nervous system, or an embedded computer) on the other.

To describe the Umwelt, it seems plausible to use cause-effect relationships. Karl Friston's work gives us an evolutionary explanation of why an agent would create them – to be able to lower the entropy of its body states via those cause-effect relationships between actors and sensors.

2.2 The Two Views

This thesis relies on a dualist 1st person *and* 3rd person view on an agent model. The work presupposes, that there is a difference between the objective, real world, and what the agent *imagines* the world to behave like. Here, *imagining* is used to describe in a rather blunt way, that the agent's behaviour is rationally explainable in such an imagined world. This especially does not imply any claims of the agent's ability for experience and consciousness. A detailed graphical illustration of the complete Cognitive Body model is shown in Figure 2.5.1 on page 26.

The Third Person View

The (objective) *third person view* is the most prevalent in current research of Computer Science and Machine Learning (Russel & Norvig, 2002; Barber, 2011). The agent is separated from the Environment via some distinguished boundary, usually creating a continuous extent in space (e.g. a cell membrane, or a mobile robotic computer system), but sometimes this might not be that obvious (e.g. software agents). Interaction between agent and environment is organised into sending channels (actors, efferent signals), and receiving channels (sensors, afferent signals). Also, both agent and environment have states. In the case of the agent, states often are called memory. This model closely resembles the architecture of a computer. The agent's internal structure is thought to be complicated, but *mechanistically* (i.e. *algorithmically*) explainable. The explanation may include probabilistic elements.

An important detail for understanding this thesis is, that in this *third person view*, an agent only can be reactive, in the sense that the algorithm and its parameters (i.e. the goal) are ontogenetically predetermined. This behavioural "destiny" is either supported by an evolutionary argument or by some external omniscient entity. An evolutionary argument might be, that e.g. the goal of maximizing energy resources is highly advantageous with respect to not trying, and therefore we can expect an energy-maximizing algorithm to exist with high probability. The other option is, to just let an omniscient entity choose the right parameters/goals. We usually call those entities an engineer, designer, or scientific experimenter.

One can easily see, that a promise of understanding or designing a genuinely autonomous agent cannot be delivered in this view, as the agent is not in charge of its development or goals, or environmental observations in the first place. Nevertheless, the *third person view* has powerful properties, such as independence from the actual observer and causality of explanations. This especially means, that we (as experimenters) can deliberately cut any loops between Agent and Environment (by altering the environment appropriately) to greatly simplify our analysis of parts of the Agent-Environment system. The *third person view* fits very well with investigating and describing non-circular structures. That is, structures that can be cast into the causal *Input-Processing-Output* scheme of Computer Science.

The First Person View

The *first person view* of an agent model is different. We literally put ourselves into the position of the cognizing agent, i.e. we draw the boundary around the cognitive apparatus.The uniform nature of it's physical signals enable the agent to potentially access any internal state. In case of humans and mammals, this is the nervous system, where only electrochemically transmitted events exist. Humberto Maturana (Maturana & Pörksen, 2008) highlights this feature by calling those systems *closed*, in the sense that those signals cannot leave the nervous system, and signals are also exclusively generated within the nervous system. This feature of *autopoietic* systems excludes the possibility of directly sensing e.g. light, sound, newtonian forces, the agent's physical body, etc. Consequently, those notions necessarily have to be constructed by the cognitive apparatus itself. The way the agent *thinks* the environment behaves, is called the agent's *Umwelt (von Uexküll, 1934).*

For a scientific, objective description of reality, this view is not well suited, as every agent has its own private Umwelt, and worse, other agents have to be constructed in the Umwelt first, rendering the notion of an objective description negotiated with imagined agents whimsical. Nevertheless, from the point of understanding an autonomous agent, there is a powerful advantage. The Umwelt can be changed by the cognitive apparatus, thus giving us the possibility to describe phenomena, where exactly this enactment of restrictions on the environment (mediated by a certain imagined Umwelt) plays a crucial role. This includes having a mental body model, skilled tool use, improvised tool use, pretend play, and "what-if" simulation.

It is important to understand, that neither *first person view* nor *third person view* are the one and only right paradigm of description. Both perspectives have their strengths and their respective blind spot.

2.3 Accessibility of States: Mirroring the Umwelt

Representation by Complementary Reverse Causal Models

We previously established in this chapter the favourability of successfully hypothesizing cause-effect relations and accompanying environmental states outside of the cognitive apparatus. We can embody such a set of hypotheses by a representative cause-effect model (e.g. a belief network, or a simple logic function) within the agent.

Fundamentally, the autonomous agent wants the environment's structure to be mirrored by an internal model, to give it access to states not trivially accessible to the cognitive apparatus (i.e. native nerve signals). Usually, this goal is fulfilled by setting up a *simulation*. A simulation recreates the environmental structure as a copy (up to a certain accuracy). Though, with this approach the agent faces the need for constantly matching and correcting its full simulation state with the environment, to make sure the simulation state stays accurate. One can easily see that for any sophisticated model this requirement gets increasingly hard to satisfy due to the bottleneck of sensors and intrinsic observability, or simply because of the system's dynamics (chaotic attractors).

A different approach is to use a complementary *inverse model*. The agent is subjectively not interested in actually recreating a faithful structure of its environment, but just to have a method for measuring and effecting a certain state with a good accuracy or certainty. In mathematical terms, this can be expressed as applying an inverse function to the one performed by the environment. For effecting a certain environmental state, we can then "simply" set the corresponding agent state and apply the appropriate inverse function:

$$\begin{aligned}
F_{env} : &\quad x \rightarrow states \\
F_{model} : &\quad states \rightarrow x \\
state_{env} := &\quad F_{env}\left(F_{model}\left(state_{agent}\right)\right) \\
F_{model} \approx F_{env}^{-1} \quad \Rightarrow &\quad state_{agent} \approx state_{env}
\end{aligned}$$

Here, $state_{env}$ is a hidden environmental state, and $state_{agent}$ is its mirrored representation in the agent. Of course, this presupposes $F_{env}$to be bijective for at least most states x. The argument also works, when swapping F_{model} and F_{env}, so that:

$$\begin{aligned}
F_{model} : &\quad x \rightarrow states \\
F_{env} : &\quad states \rightarrow x \\
state_{agent} := &\quad F_{model}\left(F_{env}\left(state_{env}\right)\right) \\
F_{model} \approx F_{env}^{-1} \quad \Rightarrow &\quad state_{agent} \approx state_{env}
\end{aligned}$$

These two variants actually correspond to successful acting (first) and sensing (second).

F_{env} can be thought as a cause-effect relationship, while F_{model} consequently is an effect-cause relationship. In the rest of the thesis, F_{model} is called the *Reverse Causal Model* (R-Model), and is implemented and represented by the cognitive apparatus (i.e. the brain) of the agent. The inverse function of the R-Model is termed *Forward Causal Model* (F-Model) and denotes the cause-effect function of the environment, *as the agent thinks it is* (its *Umwelt*). This does not necessarily mean that the F-Model is identical, or even remotely close, to the (objective) F_{env}. The F-Model is not implemented anywhere, but is implicitly defined by the corresponding R-Model. This is the key difference to a Simulation-based approach (where the F-Model would be implemented).

An important key point is, that the R-Model (and especially the implicit F-Model) is constructed by the agent. The F-Model does not necessarily accurately reflect the structure of the environment, though this would be highly desirable. It rather represents the structures the agent thinks there are (its *Umwelt*).

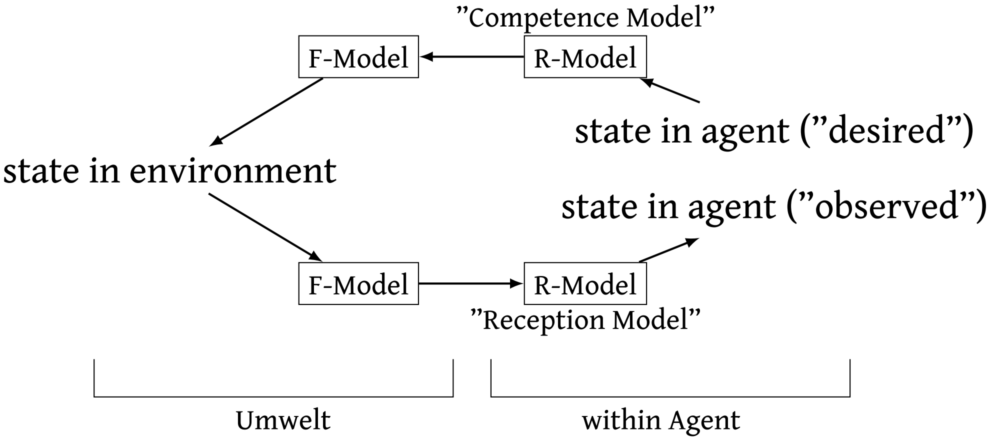


Figure 2.3.1: *Reverse Causal Model* and *Forward Causal Model* connecting the agent states with the constructed environmental state.

Competence and Reception Model

An illustration of the Umwelt as hypothesized by the agent is shown in Figure 2.3.1. It makes a distinction between two possible R-Models. One, that calculates what the (constructed) environmental state is (*Reception model*) and the other calculates actions necessary to bring the environmental state in line with the desired one (*Competence model*). These causal paths relate to observing and controlling an environmental state respectively.

Figure 2.3.1 shows a small inaccuracy though. By having two agent states, there should actually be two mirrored ones in the Umwelt, but there is only one shown.

The figure supposes, that both *Competence Model* and *Reception Model* relate to the same environmental state, but it leaves the possibility for the input of the *Competence Model* (desired state) and the output of the Reception Model (observed state) not to be equal, to account for errors along the causal chain, and not to form a loop, which would defeat the claim on causality.

Matching the F-Model with the Environment by Invariant Causal Chains

An F-Model is only good for the agent if it accurately represents the behaviour of (parts of) its environment. To match the F-Model to the behaviour of the environment, we can decompose the function performed by the environment into:

$$F_{env} = F_{unknown} \circ F^{-1}_{model}$$

or alternatively:

$$F_{env} = F^{-1}_{model} \circ F_{unknown}$$

where $F_{unknown}$ represents an arbitrary transformation, that conceptually is responsible for any deviations between $F^{-1}_{model}$and F_{env}. If we assume two of such transformations, one mapping efferent signals to an environmental state, and one from an environmental state to the afferent signals of our cognitive apparatus, we can construct a chain of cause-effect relationships spanning from efferent to afferent signals:

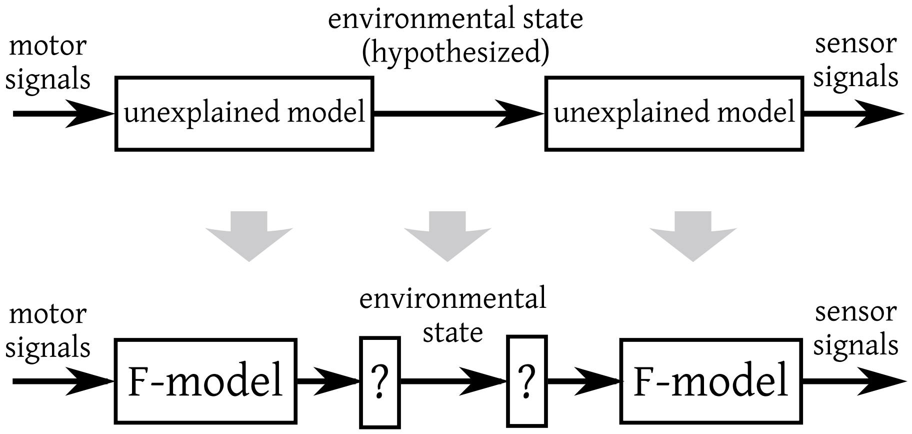


Figure 2.3.2: Conceptual substitution of the functions performed by the environment.

The agent's goal now is to find R-Models, so that the individual $F_{unknown}$ functions get trivial (Identity function). If the cognitive apparatus of the agent succeeds, it has found an *invariant causal chain*, as it is illustrated in Figure 2.3.3. The invariance refers to the fact, that the output (observed state) is highly correlated with the input (desired state).

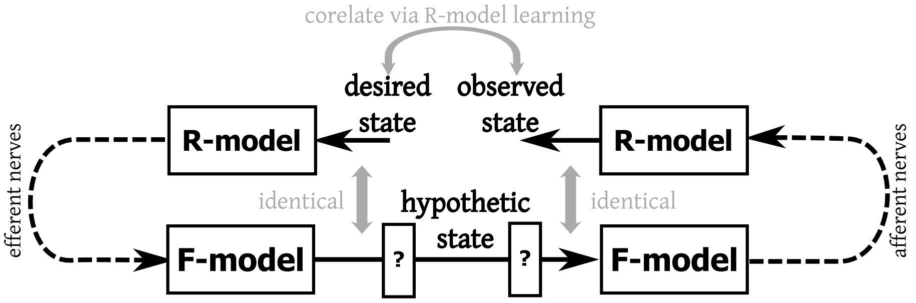


Figure 2.3.3: Illustration of an *Invariant Causal Chain*.

Once such an *invariant causal chain* passing through the real environment is established, the (indirectly) constructed Umwelt state is supposed to have a corresponding state in the environment. At least, the environment will behave, like the F-Models describe it.

By definition, the Umwelt state will also exhibit the same (or homomorph) state trajectory as the two internal states. This *mirroring* of the Umwelt state by agent-internal states makes it possible to transparently access the hitherto hidden Umwelt state. It can now be accessed like any other internal signals of the closed, autopoietic cognitive apparatus.

Some readers might be inclined to think that the environmental state is simply traced, or copied, by an internal representation, as proposed by Emulation Theory (Grush, 2004). This would be a wrong conclusion, as we are not talking about an objective environmental state, but of a state of the (subjectively constructed!) Umwelt. As actually the perceived state is constructed within the agent's cognitive apparatus, and the Umwelt state is implicitly assumed to exist, it is real in the same sense an ephemeral mirror image is real. Although, because of establishing the invariant causal chain by going through the external environment, it is a reasonable assumption. In a certain sense, we can imagine ourselves as the cognitive apparatus being in a closed room without windows to an outside world. Suppose we additionally have a slit to pass simple orders and receive factual reports from the other side of the walls. We could then put up mirrors on the wall and arrange some parts of the interior, so that the mirror image looks like an actual outside world to us. If we can find an interior arrangement and a set of rules to adapt it in accordance to the flow of orders to and reports from the outside world, we can then pretend the mirror *to actually be* a window to the outside world.

We could speak of a constructivist creation of the Umwelt by the agent. Though, I think the environmental involvement with the proposed method (the interaction with an objective environment via afferent/efferent signals) makes this explanation not a very accurate one. I want to stress, that the creation of the *invariant causal chains* is on one hand restricted by the behaviour of the objectively existing environment, and on the other hand by the subjectively constructed Umwelt. Of course, if the differences between Umwelt and environment vanish (as it should be in a working model), the distinction is all but theoretic.

How do we formulate a method to create those *invariant causal chains*, i.e. how to marginalize the contribution of $F_{unknown}$? This problem will be tackled next.

2.4 Constructing State Approximations

To find *invariant causal chains*, the cognitive apparatus first constructs two candidate Umwelt states, which are then adapted to approximate the supposed environmental one. Unfortunately, without direct access to the environmental state, the cognitive apparatus cannot simply calculate an information-theoretic distance measure directly for an optimization. But we can use the *Data Processing Inequality theorem* (see Section A.1), to calculate an upper bound on the *Information Distance* between the approximations and the environmental state, or equivalently, a lower bound on the *Mutual Information* . The theorem supposes a *Markovian Chain* though, which means that the Umwelt needs to follow this constraint.

A Lower Bound on Mutual Information with a Hidden State

Presuming the desired state S_d, environmental hidden state S_h, and observed state S_o to form a causal chain, we can apply the Data Processing Inequality theorem (Section A.1) to those states. This gives us a lower bound on the Mutual Information $I\left(S_h, S_d\right)$ based on entropies, which only depend on S_d and S_o, but not on the inaccessible S_h.

$$\begin{aligned} I\left(S_d; S_h\right) &\geq I\left(S_d; S_o\right) \\ &\geq H(S_d, S_o) - H\left(S_d|S_o\right) - H\left(S_o|S_d\right) \end{aligned}$$

where $H\left(\cdot|\cdot\right)$ denotes conditional entropies. The *joint entropy* $H\left(S_d, S_o\right)$ can be decomposed into *conditional entropies* $H\left(\cdot|\cdot\right)$ and (unconditional) *entropies* $H\left(\cdot\right)$*:*

$$\begin{aligned} H\left(S_d, S_o\right) + H\left(S_d, S_o\right) &= \left[H\left(S_d\right) + H\left(S_o|S_d\right)\right] + \left[H\left(S_o\right) + H\left(S_d|S_o\right)\right] \\ H\left(S_d, S_o\right) &= \frac{H\left(S_d\right) + H\left(S_o\right)}{2} + \frac{1}{2}H\left(S_o|S_d\right) + \frac{1}{2}H\left(S_d|S_o\right) \end{aligned}$$

Substituting this equivalence into the *Data Processing Inequality* theorem yields

$$\begin{aligned} I\left(S_d; S_h\right) &\geq H(S_d, S_o) - H\left(S_d|S_o\right) - H\left(S_o|S_d\right) \\ &\geq \frac{H\left(S_d\right) + H\left(S_o\right)}{2} + \frac{1}{2}H\left(S_o|S_d\right) + \frac{1}{2}H\left(S_d|S_o\right) \quad - H\left(S_d|S_o\right) - H\left(S_o|S_d\right) \\ &\geq \frac{H\left(S_d\right) + H\left(S_o\right)}{2} - \frac{1}{2}H\left(S_o|S_d\right) - \frac{1}{2}H\left(S_d|S_o\right) \end{aligned} \tag{2.4.1}$$

This lower bound on the Mutual Information between the desired state and the hidden environmental state can be maximized by the agent, without having direct access to S_h. The same lower bound can be derived for $S_h$and S_o by starting with a different version of the *Data Processing Inequality:*

$$\begin{aligned} & I\left(S_h; S_o\right) \geq I\left(S_d; S_o\right) \\ \Longrightarrow\quad & I\left(S_h; S_o\right) \geq \frac{1}{2}H\left(S_d\right) + \frac{1}{2}H\left(S_o\right) \quad - \frac{1}{2}H\left(S_o|S_d\right) - \frac{1}{2}H\left(S_d|S_o\right) \end{aligned} \tag{2.4.2}$$

Indirect Optimization of State Approximations

As the proxy states S_d and S_o are calculated by the agent, they can be changed to better fit the hidden state S_h. We can do this indirectly by maximizing the lower bound in Equation 2.4.1. We assume to optimize over two sets $\mathcal{M}_c, \mathcal{M}_r$ of possible *Reverse Causal Models,* that are available to define Umwelt states S_d and S_o respectively. Thanks to the universal property of entropies, $H(X) \geq 0$, we can easily split the optimization of the mutual information into four separate tasks: Minimizing $H(S_o|S_d)$and $H(S_d|S_o)$, and maximizing $H(S_d)$ and $H(S_o)$.

We therefore need to find a Competence model (or function) m_c and a Reception model m_r :

$$\begin{aligned} m_c^{-1} : &\quad \text{efferent signals} \rightarrow S_d \\ m_r : &\quad \text{afferent signals} \rightarrow S_o \\ m_c &= \operatorname*{arg\,min}_{m \in \mathcal{M}_c} H\left(m^{-1}\left(\text{efferent signals}\right)|S_o\right) &\quad (2.4.3) \\ m_r &= \operatorname*{arg\,min}_{m \in \mathcal{M}_r} H\left(m\left(\text{afferent signals}\right)|S_d\right) &\quad (2.4.4) \end{aligned}$$

Note that m_c^{-1} is, what was called a *Forward Causal Model* (Section 2.3) , whereas m_r is a *Reverse Causal Model* . This distinction is important when maximizing the entropies $H(S_d)$ and $H(S_o)$ of Equation 2.4.1.

A look at Figure 2.3.3 tells us, that the variable S_d can freely be generated by any stochastic process p out of a set $\mathcal{P}$ of possible processes, because it is then mapped to the efferent signals by m_c .

$$S_d = \operatorname*{arg\,max}_{p \in \mathcal{P}} H(p) \quad (2.4.5)$$

S_o, on the other hand, is computed by the *Reverse Causal Model* m_r. Therefore we need to maximize the entropy of the Reception model itself:

$$m_r = \operatorname*{arg\,max}_{m \in \mathcal{M}_\mathrm{r}} H\left(m\left(\text{afferent signals}\right)\right) \quad (2.4.6)$$

Note, that this equation might conflict with Equation 2.4.4.

The four separate optimization goals now need to be realized by an optimization strategy.

Optimizing the Competence Model

We can minimize the entropy in Equation 2.4.3 by changing S_d, computed by the *Forward Causal Model* (Section 2.3) m_c^{-1} in the Umwelt with the agent's actors as input, and S_d as output. As the model is actually defined by its inverse function m_c in the cognitive apparatus, we can minimize the Conditional Entropy by optimizing m_c. The learning goal is, to increase the similarity between S_d and S_o, i.e. to select models where $S_d = S_o$ is satisfied more often (learning of the motor side *Competence model*). For continuous state variables, the correlation $\sigma(S_d, S_o)$ can be maximized. If the *Reception model* is held constant during such an optimization, then this optimization equates to Reinforcement Learning.

In the Cognitive Body model, the learning algorithm of the Competence model is given the reward signal $r_{competence}(t) = Equality(S_{d,}(t), S_o(t + \Delta t))$.

Optimizing the Reception Model

We can minimize the entropy in Equation 2.4.4 by changing S_o, which means changing the afferent *Reverse Causal Model* m_r on the agent's sensor side. We do this in exactly the same fashion as with the efferent one, i.e. choose models that more often satisfy $S_o = S_d$ (Learning of the sensor side *Reception model*). For continuous state variables, the correlation $\sigma(S_d, S_o)$ can be maximized. If the *Competence model* is held constant during such an optimization, then this optimization equates to *Supervised Learning* (statistical Classification).

In the Cognitive Body model, the learning algorithm of the Reception model is given the reward signal $r_{reception}(t) = Equality(S_d(t - \Delta t), S_o(t))$ and seeks the model with the highest reward. This is convenient, as the structure of the Reception model becomes exactly like the Competence model . Of course, the Reception model can also be optimized by using a supervised algorithm. In that case, the training signal would be $y(t) = S_d(t - \Delta t)$.

Optimizing the State Distribution of S_d

Maximizing the entropy of the desired state, $H(S_d)$, is very easy when the situation is dedicated exclusively to learning. Then, the agent can simply sample S_d from a uniform distribution for maximum entropy. During performance (i.e. when the set of possible desired states is predefined), the agent can still support online learning or adaptation by maximizing the entropy within the restricted set of desirable states. This means, that intentional variations in action execution can be used to support online model adaptation, by keeping the bound on Mutual Information high.

Optimizing the State Distribution of S_o

At first glance, $H(S_o)$ cannot be changed as easily as $H(S_d)$ can be. But this is not necessary anyway, if the three previously defined optimizations are conducted. When the conditional entropies $H\left(S_d|S_o\right)$ and $H\left(S_o|S_d\right)$ successfully minimized close to zero, then $H(S_o)$ roughly equals $H\left(S_d\right)$:

$$\lim_{\substack{H(S_d|S_o)\to 0 \\ H(S_o|S_d)\to 0}} H\left(S_o\right) = H\left(S_d\right)$$

Thus, $H(S_o)$ approximately gets maximized simultaneously with $H(S_d)$.

2.5 The Cognitive Body Model

Figure 2.5.1 shows a big picture of the Cognitive Body model, incorporating the two distinct perspectives (1st and 3rd person view), where they overlap, and where they differ. Additionally, the figure hints at the relationships between the functions of the environment, the Umwelt and the cognitive apparatus. Finally it also shows, that the learning signals derived in the previous section, form loops when considering information flow in both 1st and 3rd person view. Without considering learning, the model simplifies to a chain of functions.

The mirror in between environment and brain is a metaphor for the intangible Umwelt, which is a "reflection" of the brain's internal organisation. Like a mirror image, it can be treated *as if* it was real (1st person view), or be ignored (3rd person view). Vice versa, the same distinction can be made with the physical environment (ignored in 1st person view, real in 3rd person view).

For a specific set of models in the cognitive apparatus to be (close to) optimal, they have to represent an Umwelt *behaving* similar to the environment.

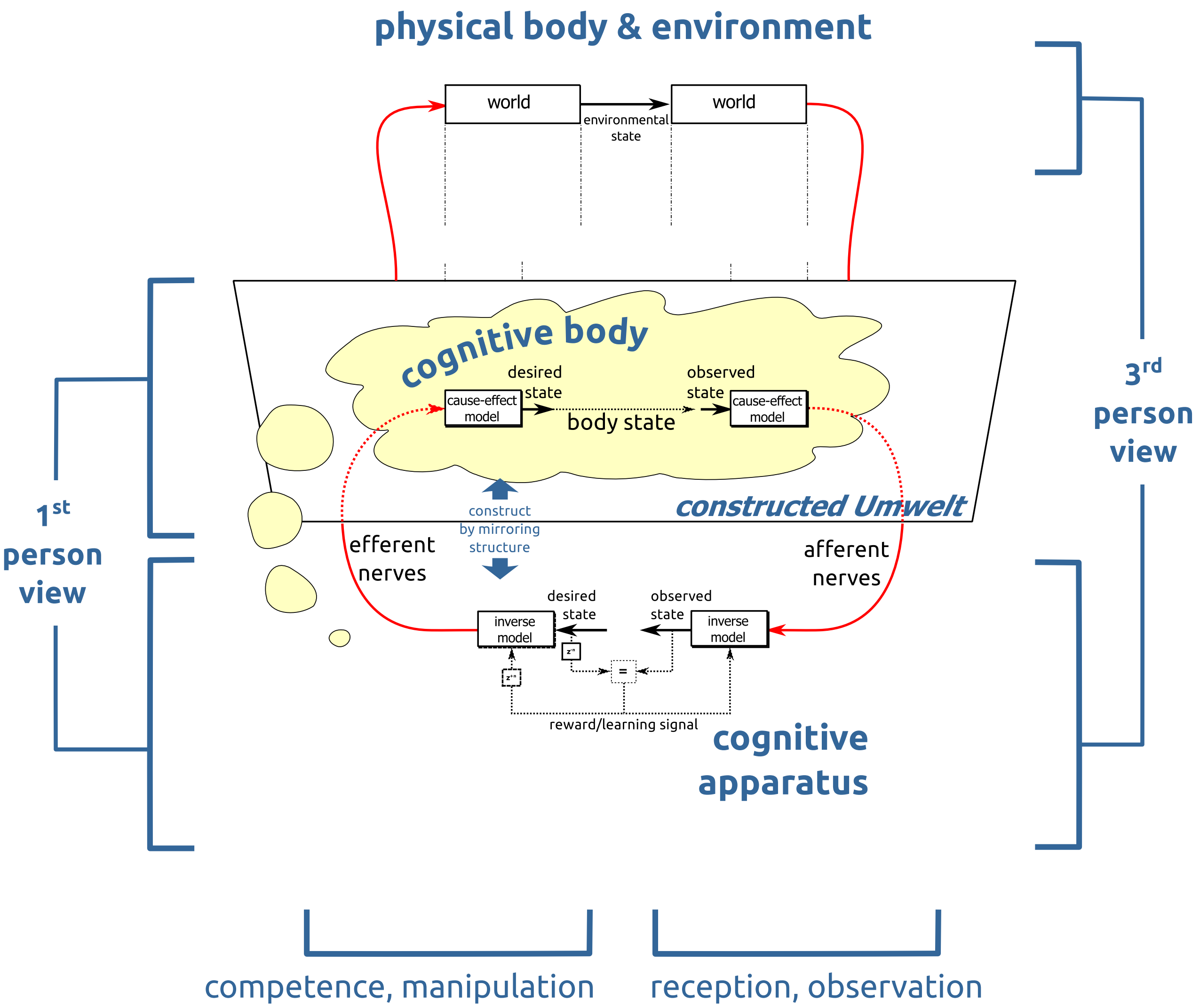
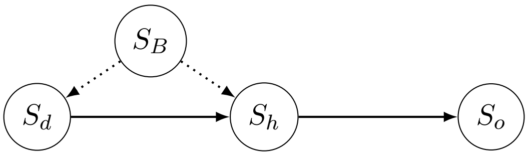


Figure 2.5.1: Overview of the complete Cognitive Body model, with the relations between the three distinct domains of *cognitive apparatus* (brain), *Umwelt*, and *physical environment*.

The Cognitive Body model can also be vertically disected for analysis into an action-oriented column, and a sensation-oriented column. The action-oriented column deals with phenomena such as tool use, skilled manipulation, and prediction, and represents acquired competences to change the physical body and environment (hence termed *Competence column* and *Competence models*). The sensation-oriented column deals with access to states, passive observation, and recognition, and represents the ability to passively perceive (parts of) physical body and environment. As the term “Perception” already has a very well defined (and different) meaning, it was named “Reception” instead (hence the terms *Reception column* and *Reception model*).

2.6 Ensuring Independent Environmental States

We started this section with the assumption, that the environmental state S_h and the two approximations form a Markovian Chain, i.e. that the two approximations are not dependent on anything else but S_h. If this assumption is violated, then there exists another state S_B, that influences our states:



In this case, we cannot guarantee that S_h will be approximated by the S_o/S_d pair. This is called a *Byzantine fault*, named after the *Byzantine Generals Problem. (Lamport* et al. *, 1982).*

The Byzantine Generals Problem prototypically exemplifies a failure mode of a model, where the environment is not behaving differently in a benign, that is, detectable way.

A Byzantine Fault can be illustrated by the following example. For deciding on a value for the amount of fuel in a car, we could take two separate measurements, and average it. A Byzantine State could now arbitrarily change the average to a certain value by influencing one value alone, based on the knowledge of all the other measurements. Without additional measurements (say, checking the fuel consumption while driving, or reading the gauge at the gas station), there is no possibility to detect, that the actual fuel level is different from the reported one.

Another example, that is more relevant to robotics and simulation, is that the Byzantine state is a neglected experimental bias, where the desired state and the hidden environmental one are (maybe only partly) predetermined by the setup of the experiment.

Taking Care of a Byzantine State

Luckily, there's a solution to avoid Byzantine States with high probability. By using a random stochastic process to generate desired states, we implicitly affirm the desired state's independence from any other possible state. When the agent exclusively is learning (and not performing), it is therefore best to use a random state trajectory. If a truly random source is not available, one can fall back on cryptographic methods, or even chaotic oscillators, to lower the chance of being accidentally correlated with other environmental states.

Of course, during actual performance (using the models), the state trajectories of the desired states cannot be selected arbitrarily. Here the danger of being misled by a Byzantine Fault can only be mitigated by exploiting as much randomness in the set of feasible state trajectories as possible.

2.7 Learning Reverse Causal Models

The key to a successful Cognitive Body model, is learning the appropriate Reverse Causal Models (R-Model). The targets for optimization were already derived in section 2.3. The R-Models need to optimize the conditional probabilities from desired to observed state and vice versa. This can be either done by correlative learning algorithms (Supervised Learning, Linear Regression), or by algorithms formulated to use reward signals (Reinforcement Learning). Figure 3.1.2 on page 32 shows an illustration of a setup with two R-Models.

Because we need to learn (at least) two R-Models concurrently, and they provide their reward signals for each other, we have a recursive feedback loop, making the evolution of such a system nontrivial. By introducing such a loop, the system in principle might converge to trivial models (lock ups), or exhibit oscillations in learning (circular attractors).

Compensating for Time Delays

An important detail not yet considered, is the time aspect of the states and reward signals. Due to the constraints of causality, there is always a nonzero positive time delay in F-Models. A truly inverse R-Model would therefore need to have a negative time delay to compensate for this fact. Unfortunately, this is impossible, as the R-Model itself has to be implemented in the real, causal world. Therefore, the time delay cannot be compensated by the R-Models, and thus the observed state is delayed relative to the desired state. This delay needs to be considered when comparing them for learning.

For calculating the incremental update of the *Reception model*, we only need to delay the desired state by the time delay accumulated along the causal chain. For the *Competence model*, things are more complicated, though. For comparing the two states, we need the observed state of the future, i.e. we need a negative delay on the observed state. We have to use a trick. By delaying the calculation of the update rules, we can implement negative delays (relative to other signals). As the learning is done incrementally in small steps, this general delay of learning does not significantly change the learning procedure.

2.8 Hierarchical Extension

Since a successful approximation of an Umwelt state both includes evolving the capability to observe and control it, we can also treat this newly emerged state as a new action or sensor. If we can manage to learn a larger set of stable Umwelt states, we can theoretically construct hierarchies of *Reverse causal models* on top of them. This possible layering is illustrated in Figure 2.8.1.

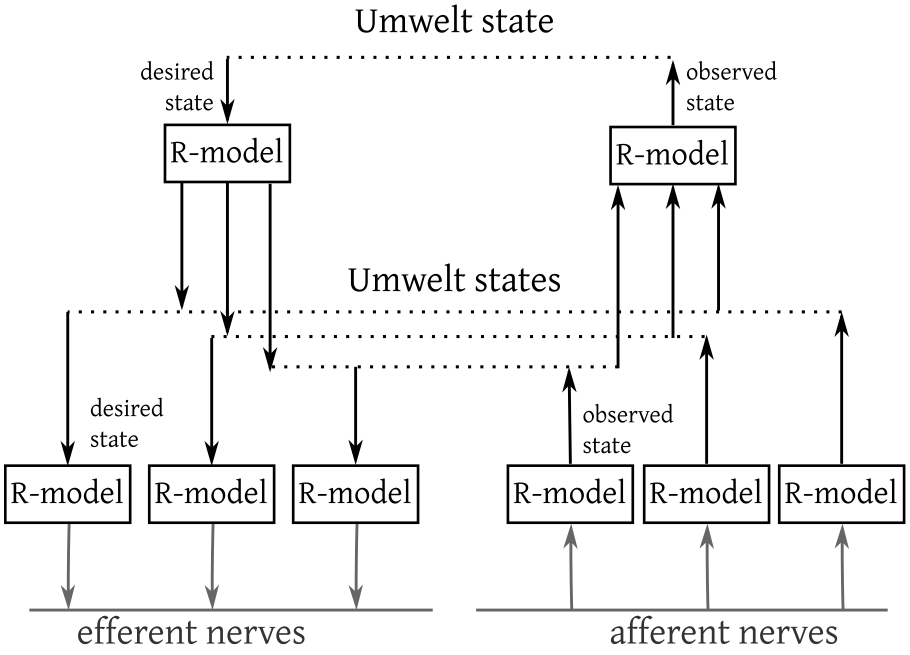


Figure 2.8.1: Illustration of the possible hierarchical construction of layers of Umwelt states

This hierarchical modelling especially would get interesting, when the lower-level R-Models would still be learning and adapting. They could then possibly optimize the semantics of their states to provide a better fit, given the non-random restricted state trajectories imposed by the upper layers, while the upper layers would be subject to those changes, too. Also, if more complex R-Models can learn longer state sequences, this would probably induce different time scales between layers (where lower layers take care of faster and easier to predict dynamics). Due to time constraints, hierarchical extension of the *Cognitive Body* model was not covered in the experiments.

2.9 Advanced Uses of the Cognitive Body Model

The Growing and Shrinking Cognitive Body

Another interesting consequence of the ability to create layered sets of Umwelt states (by grouping R-Models) is, that we can "switch" them on or off. Especially with respect to the ability of controlling Umwelt states, switching groups of R-Models on or off yields an interesting capability, which can be described as *growing* and *shrinking* of the *Cognitive Body*. If R-Models are actively used to control Umwelt states, then this Umwelt state can be considered to be part of the Cognitive Body of the agent. This way, an agent can opportunistically gain or release control of parts of the environment. The states within the Cognitive Body would usually (but not necessarily) contain the states of the physical body of the agent.

Having a growing and shrinking Cognitive Body also necessitates to comprehend perception as a dynamic process, where the agent altering its perception is a fundamental feature, and thus there is no single fixed "right" world model that can describe the world independently of the agent's stance.

Tool Use

With the *Cognitive Body* model, the agent could also temporally gain control over states of the environment, which are not part of its physical body, i.e. during skilled tool use. By activating the (previously learned) R-Models, the tool becomes a part of the Cognitive Body, and the agent gains transparent access to the environmental states manipulated by that tool. To give an example, we can introspect e.g. biking or car driving. During skilled tool use (biking or driving), we don't experience ourself stomping our feet, swinging our arms and scanning the area in front of us. We experience ourselves to *be* the bike or to *be* the car. We can instantly access and control states like driving direction, wheel positions, body dynamics, fuel consumption, the grip of the tyres, and so on. The manipulation is *effortless* because it is transparent to higher cognitive functions.

The Cognitive Body model also offers an explanation for improvised tool use. A previously learnt R-Model can get repurposed by being activated in a different situation that it was learnt in. If the environmental structure is similar enough to the learnt one (i.e. if the improvised tool provides a function similar to the originally learnt tool), the model will provide a sensible ad hoc approximation. At the same time, the possibility to use a certain skill could possibly be detected by monitoring the Reception side models for familiar patterns. Neurons performing this proposed function might resemble the behaviour of *Mirror Neurons*.

Simulation and Pretend Play

Though simulation does not play a crucial role for performance in the Cognitive Body model, it nevertheless is easy to incorporate. A simulation of effects on the Umwelt are only necessary, when the Reception side model can not (for whatever reason) provide observations, or when the agent only wants to act *as if,* without actually effecting the environment (e.g. during Pretend Play). For those cases, we can then simply replace the observed state by a copy of the (appropriately delayed) desired state.

Very closely related to Pretend Play, is the ability of enacting norms on the environment. When writing words on a paper (using the respective R-Models), there is no environmental constraint that *only* words can be written onto it. The constraint of the paper only showing words is actively enforced by the agent. For an independent observer of such a constrained environment, it would *look as if* the paper was physically only capable of containing words.

Learning by Observation

Learning by observation can also be nicely captured by the Cognitive Body model. Depending on the type of sensors, the Reception side models will also generate state sequences for Umwelt states even if they are not currently under control of the agent. By "replaying" observed state sequences using the matching Competence model, the agent can instantly reproduce observed behaviour. Albeit, the reproduction might be very crude, as it is not optimized yet.

2.10 Conclusion

The presented *Cognitive Body* model offers a framework for an autonomous agent to structure interaction with its own body and environment into states with causal relationships. The goal is to gain the ability to control those states if wanted. This constructed structure need not necessarily be congruent with the structure of the real, objective world, though they likely are due to learning mechanisms. Because of a circular information flow (due to mutual learning signals), it constitutes a dynamical system, and therefore is subject to the same analytical problems (stability, complexity, scalability).The Cognitive Body model also offers the possibility to implement a plethora of complex phenomena of perception, like tool use, mental simulation, and pretend play, within a single framework. It does specifically not deal with planning, which is considered to work on top of the Cognitive Body. The current work also does not cover highly adaptive environments (e.g. interaction with other adaptive agents) and its resulting dynamical properties.

Chapter 3

Exploratory Simulations

This Chapter describes a series of experiments conducted to explore the general feasibility of the Cognitive Body model. As the theoretic model first needed to be implemented and tested, the first experiments rather validate the basic simulation. Nevertheless, they also show important connections to well established learning problems (Classification and Reinforcement Learning).

To minimize the number of parameters (and thus experiments), a well defined simulation was chosen over a real-world robotic implementation. This also reduces the possibility of false interpretations, because it removes many sources of error, which would in turn reduce the significance of unexpected results. It also makes it possible to study the Cognitive Body model in its simplest form. For even a very simple real-world robotic experiment, one would need to presume several Umwelt states for a sensible application of the Cognitive Body model. On the other hand, an embodied robotic experiment would control for an inadvertent selection bias of the environment.

3.1 Simulation Setup

To explore the theoretically derived model, the author devised a series of exploratory simulations. The simulation series 1 establishes the correct implementation of the used algorithms, and also serves to connect the simulation structure with those commonly used in Artificial Intelligence and Machine Learning. The simulation series 2 was conducted to progressively relax the preconditions of the simulation into a full example of the Cognitive Body model. Simluation series 3 finally takes a look at the performance, when the number of available agent states is varied.

The simulation was implemented using the *Python* Programming Language and the *Scientific Python (Oliphant, 2007)* framework.

The simulation is split into a cognitive apparatus and its environment. The environment (detailed in Figure 3.1.1 on the following page) was selected to be as simple as possible, yet sophisticated enough to not be trivially learnable. It is a simple Moore FSM (Finite State Machine) with 5 distinct states, and an input of 10 different actions. The FSM implements both deterministic and nondeterministic (random) actions. The output of the FSM is the hidden environmental state, which is then encoded by a sensor matrix to yield 10 binary sensor signals, that are available to the cognitive apparatus. The sensor matrix represents the physical configuration of the agent's sensors, while the state transition matrix represents the physical configuration of the agents actuators. Importantly, actions may lead to different outcomes depending on the current environmental state (context-dependence). Also, the environmental states are hidden from the cognitive apparatus, and constitute an information (transmission) bottleneck.

The full top-down simulation diagram for the Cognitive Body model is shown in Figure 3.1.2 on the next page. Every simulation conducted only modifies the actual implementation of individual

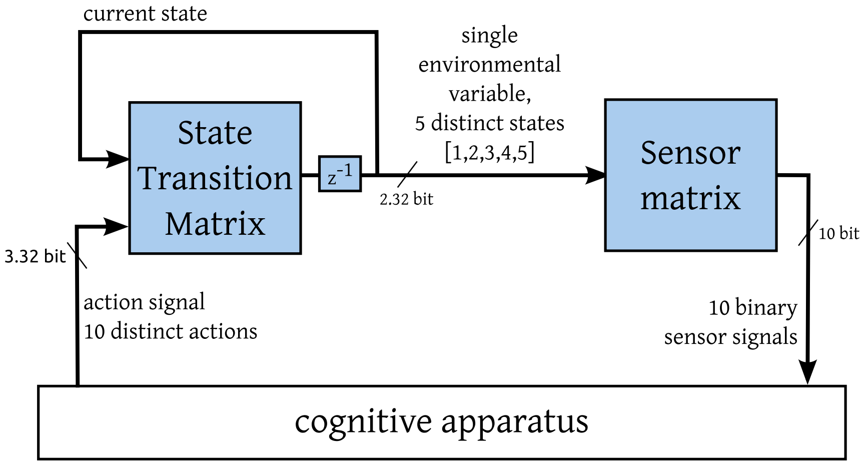


Figure 3.1.1: Emulated Environment of the cognitive apparatus.

boxes. The agent cannot access the environmental states directly under any circumstances. For convenience though, the agent's desired and perceived states in the simulation results are labelled according to the corresponding environmental state. The necessary mapping of agent state to environmental state was computed post hoc by using the maximum likelihood according to the reception model.

The model is designed to use algorithms that learn on a reward signal. The reward itself is computed by comparing desired and observed state for equality. For the Reception model, supervised learning algorithms may also be used by replacing Reception model and equality operator, directly using the (delayed) desired state as the learning signal.

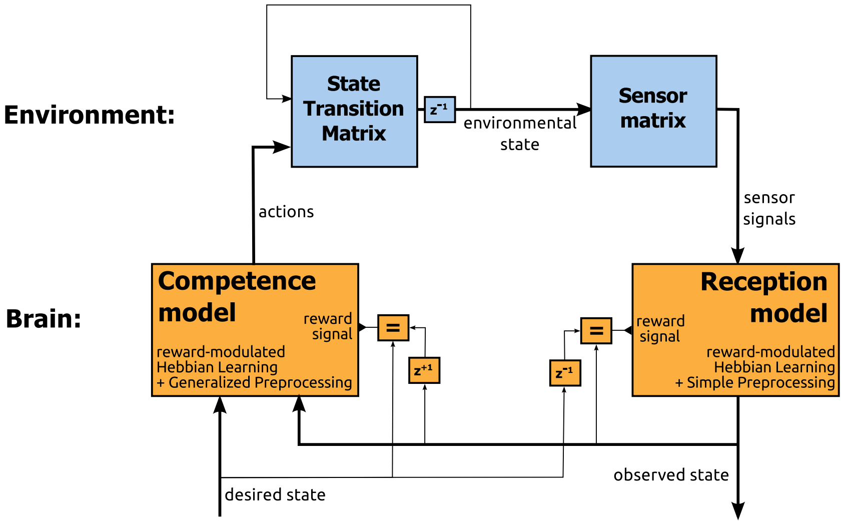


Figure 3.1.2: Complete diagram of the simulation components for the Cognitive Body model and environment. The reward signals are computed by comparing the input of the Competence model and the output of the Reception model for equality. The time delay filters (denoted z^{-1},z^{+1}) compensate the delay accumulated between desired and observed state.

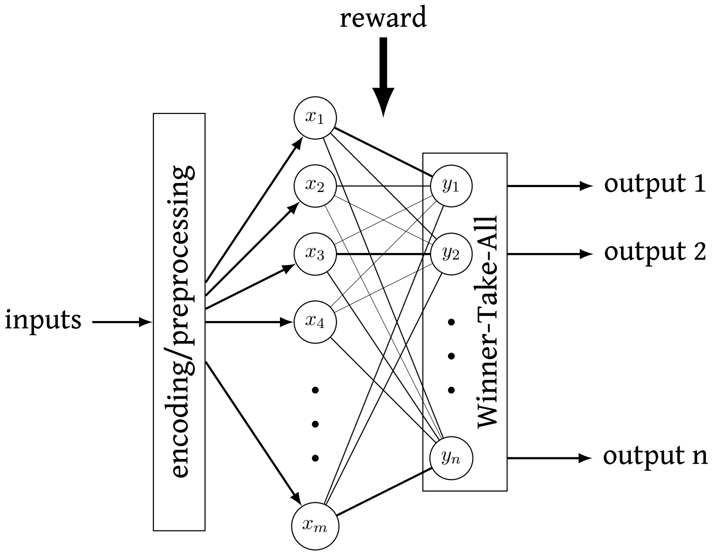
Reward-Modulated Hebbian Learning algorithm

For the implementation of the learning models, the *Reward-Modulated Hebbian Learning* algorithm *(Pfeiffer* et al. *, 2010)* was chosen. The algorithm is designed to learn policies in a typical Reinforcement Learning setup, i.e. it selects actions given a state and learns their utility with a (binary) reward signal. It operates on discrete, sparsely coded input and output neuron populations. The algorithm describes a close to Bayesian optimal weight update rule for a simple one-layer feed forward neural networks, operating on a fixed encoding of input combinations. Given a certain encoding (called *Generalized Preprocessing*), the algorithm can learn any utility function. Another feature is the locality of its learning rule. The computation of weight updates can be done independently for each synapse.

The Reward-modulated Hebbian Learning Algorithm was chosen because it explicitly operates on reward signals, is mathematically well supported, and quickly converges on optimal policies. Also, the algorithm can be used for both the Competence model and the Reception model. Possible alternatives for the Reception model would be *e.g. Support Vector Machines* (Cristianini & Shawe-Taylor, 2000), or even a supervised/unsupervised learning hybrid like the *contextual Slow Feature Analysis* algorithm (Deimel, 2009). For the Competence model, *Temporal Difference Learning (Sutton & Barto, 1998)* could be used. As the Cognitive Body model is formulated to be agnostic to the actual learning algorithm, any algorithm that can handle multiple interfering inputs and learns on a reward signal, can be used.

The main disadvantage of the Reward-modulated Hebbian Learning Algorithm is, that to be universal it requires a very resource-intensive encoding of $O(2^n)$ neurons for n binary input variables. The authors also present a less powerful and resource intensive encoding (*Simple Preprocessing*), which is used in the Reception model because of the number of input variables (10 sensor signals).

The structure of the Reward-modulated Hebbian Learning Algorithm is shown here:



The update rule for the neural network, with i indexing the input neurons, j indexing the output neurons, $x_i, y_j \in \{0, 1\}$ indicating a spike, η the learning rate, and w_{ij} being the connection weights from input neuron i to output neuron j, is:

$$\Delta w_{ij} = \begin{cases} x_i \cdot y_j \cdot \eta \cdot (1 + e^{-w_{ij}}) & \text{when rewarded} \\ x_i \cdot y_j \cdot \eta \cdot (-1 - e^{w_{ij}}) & \text{when not rewarded} \end{cases}$$

This update rule closely resembles the Hebb learning rule with a regularization term. The weights converge to the log odd ratio of the reward probability. Learning only takes place when both neuron i and j fire simultaneously, and due to the Winner-Take-All (WTA) stage, only one output neuron fires at a time (sparse coding). The amount and direction of weight change is modulated by the reward. Unexpected reward outcomes result in bigger weight changes than expected ones. The WTA stage is "soft" in the sense that not only the output neuron with the highest activation can win, which would be a deterministic "hard" WTA .

The preceding combinatorial expansion of inputs by the Generalized Preprocessing step ensures, that all possible input variable interactions can be modelled by a simple linear combination. Simple Preprocessing does not provide for modelling interactions between input variables (but is computationally more efficient).

The output of the algorithm is determined by a stochastic Winner-Take-All stage, operating on the values of the feed forward network. The selection probability of each output is calculated from the values with a positive, monotonic function, providing a parameter for the inherent Exploration/Exploitation trade off. As in the original paper (Pfeiffer *et al.* , 2010), the weighing function was chosen to be a sigmoid:

$$p(y) = \frac{1}{1 + e^{-\tau y}}$$

For all simulations, the parameter was set to $\tau = 5$, and the learning rate to $\eta = 0.1$, unless otherwise noted.

For the Competence Model, the Generalized Preprocessing flavour of the algorithm was chosen, as it enables modelling the action selection depending on both current and desired state. For the Reception model, the more scalable Simple Preprocessing was used, as it requires only $2 \cdot 10$ instead of 2^{10}neurons for encoding the 10 binary sensors in the simulation.

Modifications

The actual learning rule of the Reward-Modulated Hebbian Learning algorithm was slightly modified to be able to compensate negative time delays when calculating reward. The weight update calculation was delayed by a small, fixed number of iterations, to enable comparison with rewards from the future (positive time delays). This was implemented by replacing the pre- and post-synaptic firing coincidences ($x_i y_j$) with a *synaptic trace* :

$$\Delta w_{ij} = \begin{cases} trace\,(x_i \cdot y_j) \cdot \eta \cdot (1 + e^{-w_{ij}}) & \text{when rewarded} \\ trace\,(x_i \cdot y_j) \cdot \eta \cdot (-1 - e^{w_{ij}}) & \text{when not rewarded} \end{cases}$$

The synaptic trace is implemented as a Finite Impulse Response (FIR) filter. It can be formed locally at the synapse between input and output neurons, keeping the weight update fully local as in the original paper. A delay introduced by the added FIR filter causes the reward to be ahead of time relative to the neuron firing coincidence. By this trick, the (otherwise non-causal, i.e. non implementable) negative time delay (z^{+1} filter) needed for the reward calculation of the Competence model, can be honored.

This trick is only possible because the learning is delayed, but not the calculation of the model itself (which would also add to the time delay of the causal chain). Finally, the delay in weight updates has a negligible influence on the algorithm due to much slower dynamics (when $\eta \ll 1$).

For the Competence model, the simulations used the coefficients $[0.0, 1.0, 0.0]$ in the FIR filter (making it a z^{-1} filter). This delay is equal to the total delay of the whole causal chain in the simulations.

For the Reception model, the coefficients were set to $[1.0, 0.0, 0.0]$ (i.e. a pass-through filter), effectively reverting to the algorithm's orignal implementation. Informal simulations indicate, that when the FIR filter implements averaging over several iterations (i.e. coefficients $[0.25, 0.25, 0.25, 0.25]$), the algorithm still learns the optimal policy, but less fast. The desired state has to be held constant during at least as many iterations as the FIR filter is long, though. The results suggest that the overall time delay of the causal chain (shown in Figure 2.3.3 on page 22) can be learnt too, though this capability was not further investigated.

3.2 Simple Baseline Simulations

Goal of the Baseline Simulations was to establish tests for the implementation of the simulation, especially for the newly implemented learning algorithm.

For the Baseline Simulations, the simulated world was adapted to resemble two well known problems, Classification and Reinforcement Learning. problem. Both setups can easily be achieved by presetting either of the two models to an optimal solution.

For Classification, the Competence model is fixed to select the right actions, so that (previous) desired agent state and environmental state are always equal. One can explain this setup as the agent being its own teacher, and setting itself a class label for each training point.

For Reinforcement Learning, the Reception Model is fixed to a solution, so that the environmental state and the observed state are always equal. For the Competence Model, the learning setup then looks like calculating a reward on reaching an environmental goal state. Though, in contrast to a classical Reinforcement Learning setup (with a single fixed goal state and a related learnt policy π), we actually learn a set of (mutually independent) policies, one for each value of the *desired state* variable. The goal state is randomly switched between the individual episodes.

Simulation 1.1: Classification with the Reward-Modulated Hebbian Learning Algorithm

The first simulation pitched the Reward-Modulated Hebbian Learning algorithm to learn the reverse mapping of a random sensor matrix. The simulation setup resembles a simple classification problem. The diagram of the simulated world is shown in Figure 3.2.1. The learning rate of the Reward-Modulated Hebbian Learning algorithm was set to $\eta = 0.1$.

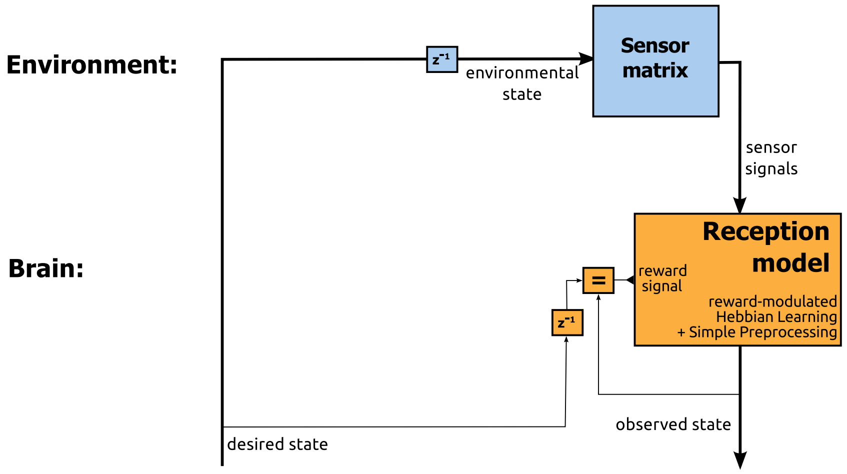


Figure 3.2.1: Simulation 1.1: Diagram of agent and environment. This resembles a Classification problem setup

In the actual implementation, the Competence model and state transition matrix were hand-crafted to yield an identity mapping from desired to environmental state, and learning was disabled by setting the learning rate $\lambda = 0$.

As expected, the Reception model quickly learned the appropriate weights. The reward average (Figure 3.2.2) directly relates to the error rate of the learnt classifier. The average reward frequency of virtually all episodes quickly approaches 1.0. Computing a deterministic policy from the learnt stochastic one (i.e. replacing the soft Winner-Take-All stage of the algorithm with a deterministic WTA) would show, that the Reception model can perfectly distinguish all states far before reaching maximum reward.

Though, there were still a few episodes, where reward probability does not converge to 1.0. This happened, because the sensor matrix was drawn from random values for each episode. There's a very small chance (about 1%) that some environmental states evoke identical sensor signals. In those cases it is impossible for any classifier to distinguish, and thus correctly reconstruct the environmental state (i.e. to find an inverse function).

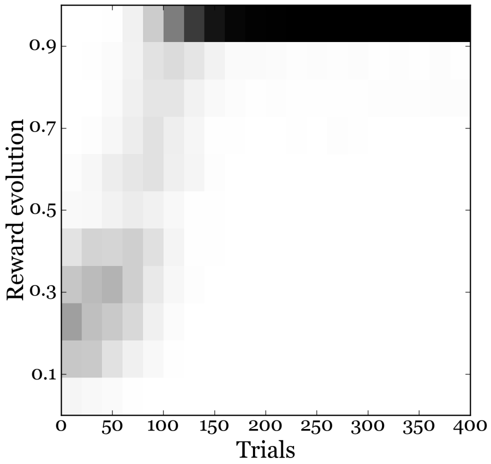


Figure 3.2.2: Simulation 1.1:Average reward frequency histogram over time (500 episodes).

Simulation 1.2: Reinforcement Learning of Context-free Actions with Simple Preprocessing

Next, the algorithm was put to work like in the original paper, doing a classical model-free Reinforcement Learning task. In the implementation, the sensor matrix was set to an Identity matrix, and the reception model was fixed to a corresponding Identity matrix to yield an identity mapping form environmental to perceived state. The setup is illustrated in Figure 3.2.3

The State Transition matrix (Table 3.1) was crafted to contain context-free actions. That is, a single action always yields the same future state, no matter what the current state is (i.e. "go to state x" actions). The rest of the states are highly context dependent.

The results shown in Figure 3.2.4 and 3.2.5 are, as to be expected. As the actions always yield the same result no matter which is the current state, the model is able to easily learn the right weights for an optimal action selection policy. But it also completely ignores the actions that change their effect depending on the current environment state, and would be perfectly fine for a certain context.

Results of the baseline simulation in the Reinforcement Learning setup with 5000 iterations and 500 episodes are shown in Figure 3.2.4 and 3.2.5. The learning rate of the models were set to η =0.1.

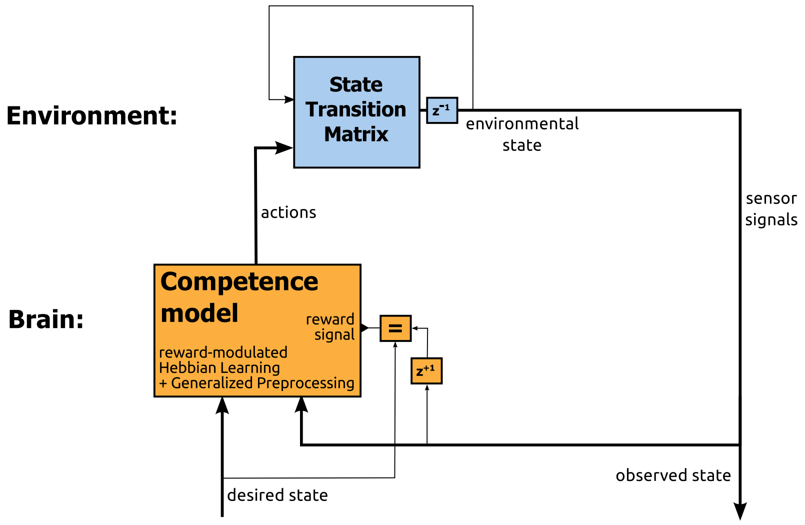


Figure 3.2.3: Simulation 1.2: Diagram of agent and environment. This resembles a Reinforcement Learning setup.

	state 1	state 2	state 3	state 4	state 5
to 1	1	1	1	1	1
to 2	2	2	2	2	2
to 3	3	3	3	3	3
to 4	4	4	4	4	4
to 5	5	5	5	5	5
+1	2	3	4	5	1
-1	5	1	2	3	4
permutation #1	1	2	5	4	3
permutation #2	5	4	1	2	3
permutation #3	5	3	2	4	1

Table 3.1: Simulation 1.2: Transition matrix of the environmental states.

Increasing the learning rate would greatly increase the speed of convergence, but also increase the variance of the connection weight fluctuation. The reward settles to a maximum of approximately 0.9 due to the stochastic action selection of the Winner-Take-All stage. Chance level probability of reward is at 0.16[1].

Figure 3.2.5 shows a Hinton diagram (see Section A.2 on page 68 for an explanation) of the final connection weight matrix. The network successfully learnt the correlation between desired state and appropriate action. The two lowest rows are biases, and are always +1 and -1 respectively. The Greyish color indicates a high variance over different episodes, but the weights always cancel each other out in specific weight matrices.

[1] The chance level is approximated by marginalizing the probability of getting into a desired state over all actions and current states (i.e. the transition matrix), and averaging over all desired states. This naive calculation does assume a uniform state distributions, and thus is not exact. For an exact calculation, a limit distribution using a Markov chain, would need to be calculated.

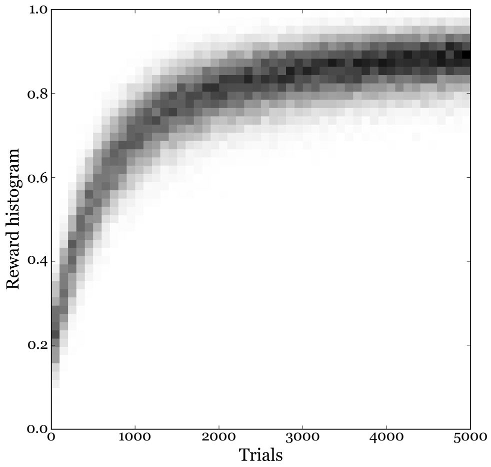


Figure 3.2.4: Simulation 1.2: Reward evolution histogram over time.

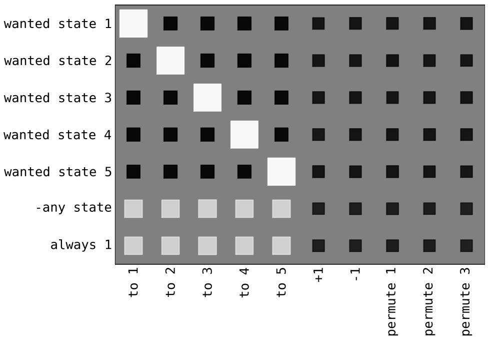


Figure 3.2.5: Simulation 1.2: Learnt synaptic weights.

The figure also shows, that context-dependent actions (i.e, the *+1*, *-1*, and *permutation #1* to *permutation #3* actions) are learnt not to be used at all. Selecting those actions indifferent to the current state, there is only an 0.2 chance to yield a reward, whereas the context-independent actions (*to 1* to *to 5*) have a 1.0 chance of reward.

Simulation 1.3: Reinforcement Learning of Context-dependent Actions with Simple Preprocessing

In this simulation, the state transition matrix was changed to only contain actions, whose end states depend on the starting state (relative movements).

The the reward evolution shown in Figure 3.2.6 approaches the maximum theoretical value 0.6, because each (learnt) action leads to the desired state in 3 out of 5 states at most.

Results of the Simulation is shown in Figure 3.2.7 reveals, that out of the possible actions to reach a state, the more context independent ones are preferred (white squares). The “snap to” actions yield the desired result 3 out of 5 times (when marginalized over the current state), which is much better than the marginal probability of 0.2 of the “+N” actions. The size of the squares (weight strengths) reveal, that the model did not find a perfect, deterministic action. The weight differences in the “-any state” and “always 1” have to cancel each other out, and can thus be ignored in the analysis.

The employed algorithm of the Competence model is Simple Preprocessing, and thus cannot learn the optimal policy. This happens, because the Simple Preprocessing step does not provide for capturing interactions between the input variables, and only can operate on marginal probabilities.

	state 1	state 2	state 3	state 4	state 5
nop	1	2	3	4	5
+1	2	3	4	5	1
+2	3	4	5	1	2
−2	4	5	1	2	3
−1	5	1	2	3	4
snap to 1	1	1	2	5	1
snap to 2	2	2	2	3	1
snap to 3	2	3	3	3	4
snap to 4	5	3	4	4	4
snap to 5	5	1	4	5	5

Table 3.2: Simulation 1.3: Transition matrix of the environmental states.

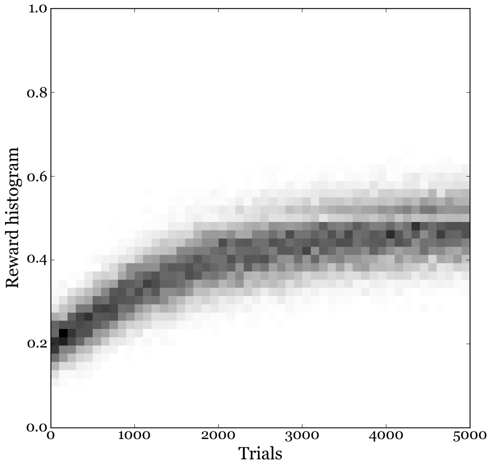


Figure 3.2.6: Simulation 1.3: Reward histogram over learning time.

Next to showing the limitations, it provides an argument for using the arguably more complicated and costly *Generalized Preprocessing* method for the Competence model.

The simulation also shows, that the model tries the best guess, and favours the “snap to X” group of actions (which have a marginal probability of 0.6 of succeeding) over the theoretically optimal “+N” group of actions. They have a marginal probability of only 0.2 and thus are not selected by the model, although there exists a deterministic, optimal action for each pair of (current state, desired state).

Note that neither the knowledge (i.e. their marginal distributions) of current state nor of the desired state alone are enough to discriminate the optimal actions if they are context-dependent. In extreme cases (such as the constructed one), there might not be any action that can be meaningfully learnt without context of the current state at all. Figure 3.2.8 shows such a case.

Since all actions then yield a similarly bad marginal probability (0.2), all weights (representing learnt log odd ratios) stay negative (shown in Figure 3.2.11). Due to the inability of the model to discriminate between certain contexts (the current state of the environment), the model settles to perform completely random action selection.

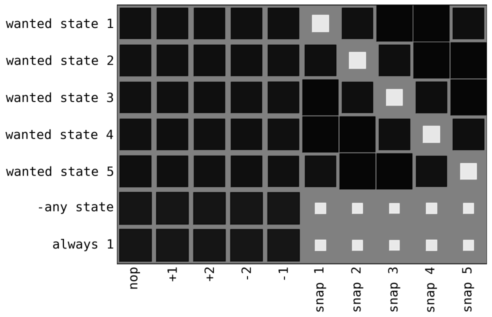


Figure 3.2.7: Simulation 1.3: Learnt Synaptic Weights (as a Hinton diagram A.2)

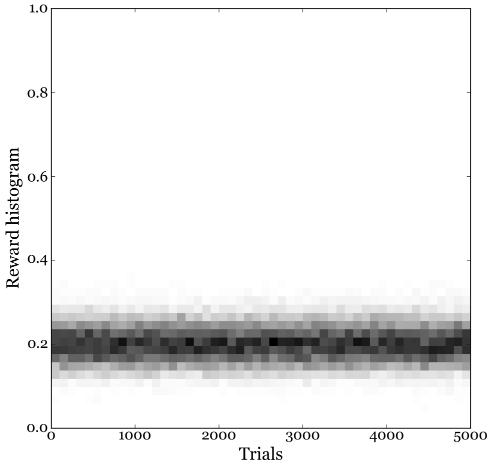


Figure 3.2.8: Simulation 1.3: reward evolution with only context-dependent action.The the reward probability stays at the chance level of 0.2

Simulation 1.4: Reinforcement Learning of Context Dependent Actions with Generalized Preprocessing

As the algorithm with Simple Preprocessing cannot capture the essential interactions between current state (the context) and desired future state for context dependent actions, we need to change to the Generalized Preprocessing flavour of the algorithm. This algorithm can learn any conditional dependencies between actions and the inputs, by simply expanding them into a sufficiently big set of conditional probabilities. Of course, this comes at the price of model complexity and increased computation demand, and a less favourable scaling behaviour.

Simulation 1.4 again uses the same transition matrix as simulation 1.2, as shown in Figure 3.1.

Running the experiment with the same environment as before shows, that the Competence model can now easily learn the optimal policy. The reward evolution in Figure 3.2.10 shows a fast increase early on. The Hinton diagram of the weight matrix (Figure 3.2.11) shows a memorable “staircase” pattern, which is due to the specific state transition matrix used in the experiment (and thus, the semantics of the actions). The whiteness/darkness of the squares represents the empirical variance over several episodes (500). Grey squares indicate high variance (logarithmically scaled), whereas pure white/black squares indicate zero variance. As shown, most squares, even the ones with small values, are very similar (i.e. predictable) over different episodes.

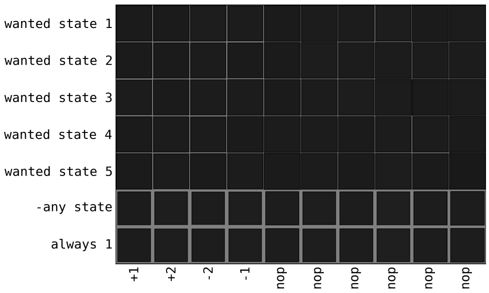


Figure 3.2.9: Simulation 1.3: learned weights

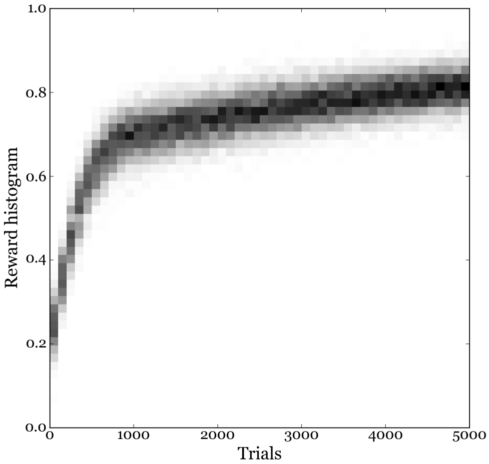


Figure 3.2.10: Reward average histogram of simulation with only context-dependent action

Conclusions to Simulation Series 1

Simulation series 1 was designed to test the simulation framework and the implementation of the employed algorithms on known tasks, and assert their proper function. This was done by splitting the Cognitive Body model into two separate parts matching the orthodox Classification and Reinforcement Learning paradigms. It also demonstrated the reasons for choosing two different flavours of the Reward-modulated Hebbian Learning algorithm for Reception and Competence models.

The simulations showed, that the algorithm is well suited in principle to reliably and quickly infer the correct inverse function to the environment for the Reception model and Competence model when they are separately learned .

3.3 Closing the Loop: Stability of Concurrent Learning

So far, the simulations only established well known behaviours, and were meant to set a baseline for subsequent experiments. In the next step, the simulation was reconfigured into its intended setup - both the Reception model and the Competence Model are learning at the same time. To introduce the changes incrementally, the first simulation is run with a manually preset weights of the Reception model, and a matching trivial sensor matrix (Identity matrix). Subsequent simulations remove the presets one by one. By doing this gradually, we are adding potentially destabilizing factors separately, and we gain a fine-grained empirical indication of the models performance. The final simulation removes any presumptions about the environmental state machine (in the models), and implements a randomly generated environment.

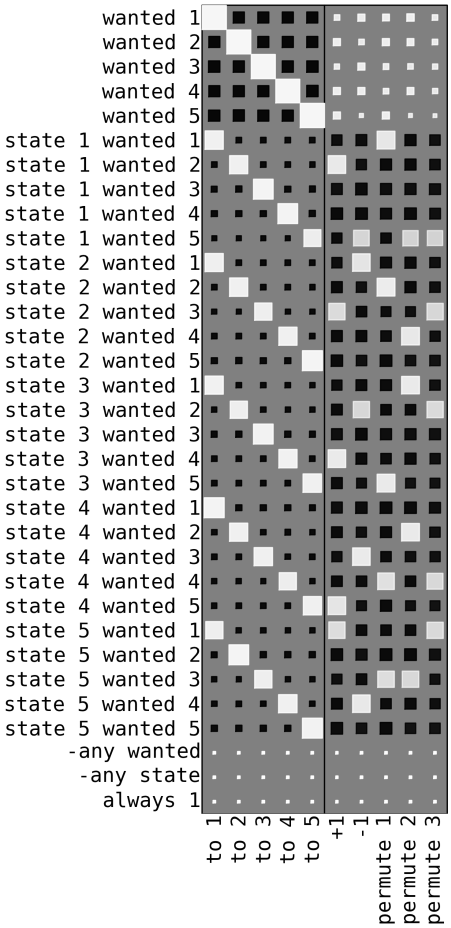


Figure 3.2.11: Simulation 1.3: learned synaptic weights.

Analytic tools

Injectivity and Surjectivity of the Implied Deterministic Map

To ease the analysis of the performance, a new type of measure was calculated. For any given simulation step, one can compute the optimal deterministic policy for the (probabilistic) Reception model. Together with the sensor matrix, it constitutes a function from environment state to agent state:

$$\begin{aligned} F: &\quad states_{environment} \rightarrow states_{agent} \\ F(e) &= \max_{x \in states_{agent}} p(x|e) \end{aligned}$$

where $p(x|e)$ denotes the conditional probability of agent state x being selected by the Reception model, given the environment state e.

This function (or map) can be characterized via the mathematical properties of surjectivity and injectivity (Section A.3 on page 68). It tells us the relation between the two realms, and can be understood as binding them together, effectively giving the agent states a *grounded meaning*.

An injective map implies, that every world state is uniquely distinguished within the agent, and thus no information is lost. A surjective map implies, that all agent states are utilized, and therefore no more states can be learnt. Either situation indicates an optimal map, in the sense that as much information as possible is transferred from environment to the agent states. For simulation series 2, an injective map is also a surjective map, because of the equal number of world and agent states.

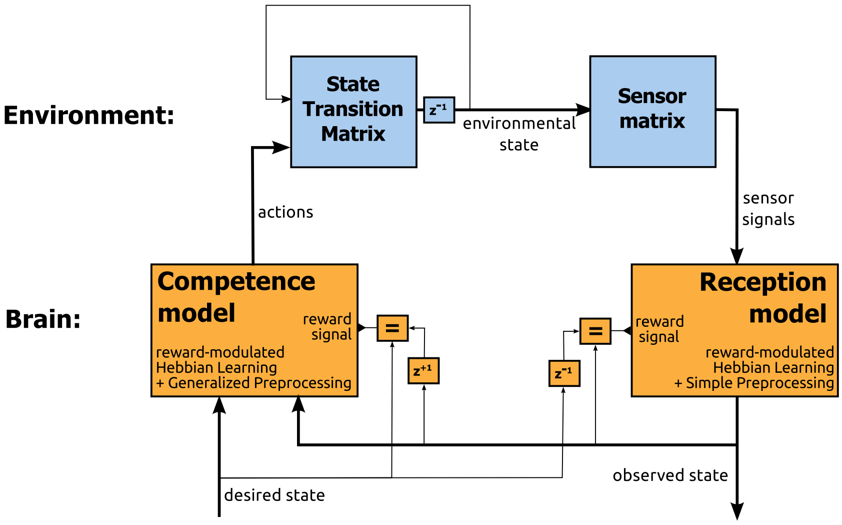


Figure 3.3.1: Concurrent Learning Setup. Both the Reception model and the Competence model are learnt concurrently, and provide each other an approximation of the theoretically optimal reward signal.

For performance analysis, only the deterministic maps calculated from the Reception model were used. A similar map could be calculated with the Competence model too, but is slightly more difficult due to the context-dependent nature, necessitating an additional marginalization over the input of observed state. When the Reception side learns to distinguish all environment states reliably, learning the Competence side simplifies to a Reinforcement Learning task. Both simulation series 1 and the original paper (Pfeiffer *et al.* , 2010) show, that this happens predictably and fast. The same is true, when the Competence model develops an injective deterministic map before the Reception model does, simplifying learning to a Classification problem. In either case, both models will quickly develop injectivity, and therefore the Reception model's map properties are a good estimator for the Competence model's map properties, and vice versa.

Unassociated States

A second, less strict measure of success was computed by counting the number of *unassociated states*. Those are agent states, that never are most likely to be selected by the Reception model, given any possible environment state. When computing the deterministic map, these states are not related to any environment state. They can be understood as having no grounded meaning (yet). The minimum possible number of unassociated states happens, when the deterministic map either is injective or surjective. The maximum occurs, when a single agent state is most likely for all environment states. So the number of unassociated states is in the interval:

$$n_{agent} > n_{unassociated} \geq max(0, n_{agent} - n_{environment})$$

where n_{agent} and $n_{environment}$ are the number of possible states of the agents models and the environment respectively. The number of unassociated states tells us, whether a model can distinguish almost all states ($n_{unassociated}$ is small), or it does not distinguish many environment states ($n_{unassociated}$ is big).

Simulation 2.1: Baseline Recreation

In the first simulation setup, the Reception model was manually preset to an optimal solution. It was set to bijectively map the environmental state to the agent's perceived state, and the learning rate was set to $\eta = 0$ to switch off learning. The Competence model had to be learnt, and was preset to assume equal expectations of reward for every action (uniform distribution). The state machine's transition matrix of the environment was set to the same context-dependent actions as the previous simulations, except that five were replaced by nondeterministic actions. These actions effect a random change of state, and thus are maximally useless for the control of state. This was done to provide for potential "false positives", as there's a chance of $p = 0.2$ that those random actions still yield the right state. Including 5 "useless" actions also decreases the likelihood of selecting the correct action by chance, thus giving a better interpretability of the reward histogram. The sensor matrix (mapping environmental state to sensor signals) was preset to the identity matrix. The tables in 3.3 show the experimental setup.

	state 1	state 2	state 3	state 4	state 5
nop	1	2	3	4	5
+1	2	3	4	5	1
+2	3	4	5	1	2
−2	4	5	1	2	3
−1	5	1	2	3	4
random	U(1,5)	U(1,5)	U(1,5)	U(1,5)	U(1,5)
random	U(1,5)	U(1,5)	U(1,5)	U(1,5)	U(1,5)
random	U(1,5)	U(1,5)	U(1,5)	U(1,5)	U(1,5)
random	U(1,5)	U(1,5)	U(1,5)	U(1,5)	U(1,5)
random	U(1,5)	U(1,5)	U(1,5)	U(1,5)	U(1,5)

	state 1	state 2	state 3	state 4	state 5
Sensor 0	1	0	0	0	0
Sensor 1	0	1	0	0	0
Sensor 2	0	0	1	0	0
Sensor 3	0	0	0	1	0
Sensor 4	0	0	0	0	1
Sensor 5	0	0	0	0	0
Sensor 6	0	0	0	0	0
Sensor 7	0	0	0	0	0
Sensor 8	0	0	0	0	0
Sensor 9	0	0	0	0	0

Table 3.3: Simulation 2.1: Transition table (left) and sensor matrix (right). U(1,5) denotes the (discrete) uniform distribution. The columns relate to environmental states, the rows to either actions or sensor vector elements.

Figure 3.3.2 shows the learnt conditional probabilities of a single episode. The names of the agent states are assigned post simulation by calculating the most likely agent state corresponding to a given environmental one (deterministic policy). Due to the (in principle) arbitrary mapping, a statistic of the synaptic weights over several episodes is not meaningful.

The results are comparable to the ones obtained in Simulation 1.4. As the implementation now employs two models, there are also two soft Winner-Take-All stages at work. This reduces the ultimately attainable reward (shown in figure 3.3.3) compared to Simulation 1.4.

Simulation 2.2: Concurrent Learning of Reception and Competence Model

The goal of this simulation was to corroborate the hypothesis that the loop, formed by environment and learning algorithms, creates a stable, stationary dynamic system, i.e. it has a static limit case once a locally optimal solution is found. Ideally, a stability analysis would be conducted by mathematical analysis and proof for certain classes of environments. Although the model (and thus implicitly the Umwelt) is set up as an analytically tractable causal chain, the model's outputs influence each other indirectly via the learning signals. Thus, it might be possible for the two models to converge to trivial, self-sufficient solutions, regardless the environment structure.

In this experiment, both models within the agent were initialized to uniform weights (corresponding to a minimal prior expectation), and learning rates were set to $\eta = 0.1$. Statistics were computed

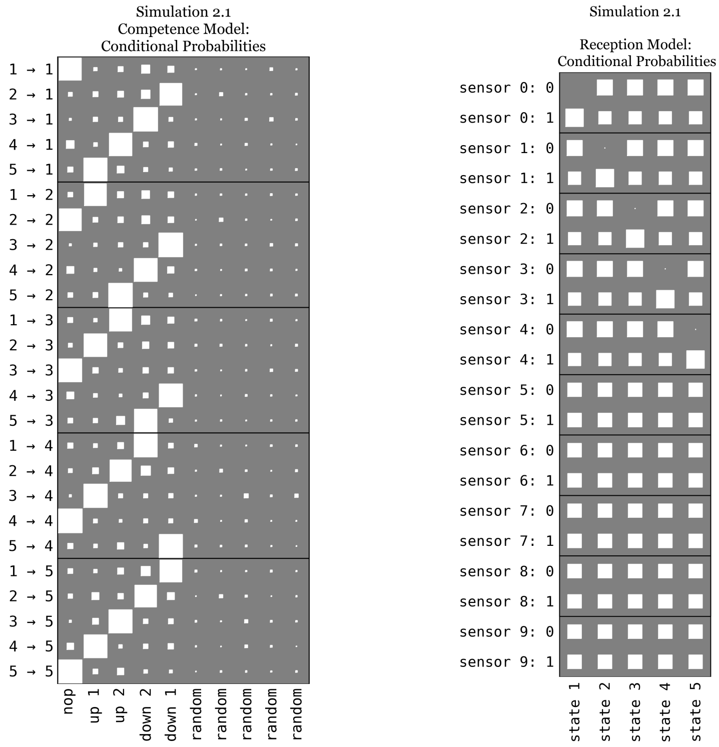


Figure 3.3.2: Simulation 2.1: Hinton diagram A.2 of the calculated conditional probabilities of the Competence and Reception model. The left diagram shows $P(\text{action} \mid \text{observed state, desired state})$, the probabilities of selecting a certain action (column), given a certain concurrence of observed and desired state (row), depicted as a desired state transition. Rows always add up to a marginal probability of $p = 1.0$. As this diagram is easier to interpret, it was used in subsequent simulation in favour of visualising weight matrices. The right diagram shows the conditional probabilities learnt by the Reception model

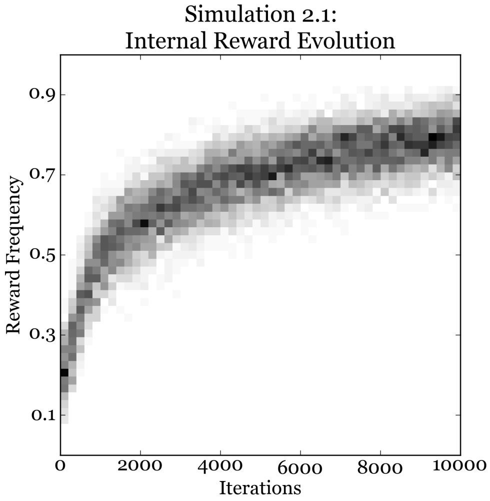


Figure 3.3.3: Simulation 2.1: Reward histogram. A high reward average implies a congruency between the state trajectories of (delayed) desired and observed state. As a difference to the previous Simulations, there are two soft Winner-Take-All stages operating, which explains the roughly twice as high rate of reward miss in the settled state.

over 50 episodes.

Note, that both models within the agent use a stochastic algorithm. Though one can easily change a parameter in the soft Winner-Take-All stage of the used algorithm to create a deterministic policy (similar to Simulated Annealing), if this is needed. In this experiment, the algorithms selects the wrong actions with a probability of roughly 10% . Thus, the (continuously learning) models are constantly perturbed by wrong model decisions (and thus faulty reward signals). They can in principle adapt to any other environmental structure as fast as if the hadn't learned anything at all. In this experiment series, the learning rate η was deliberately set very high and kept at this level to surface latent instabilities. In an application setting, of course, one would decrease the learning rate significantly, once a high reward ratio was reached for some time.

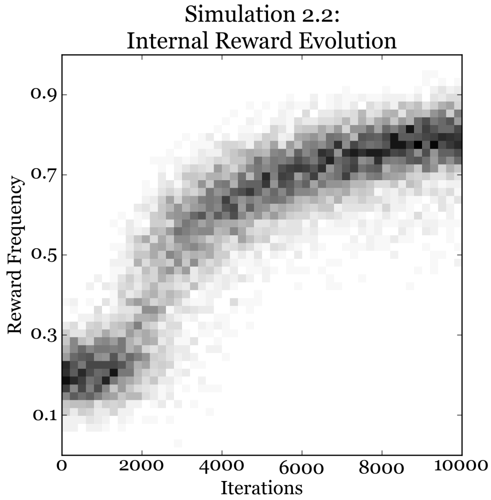
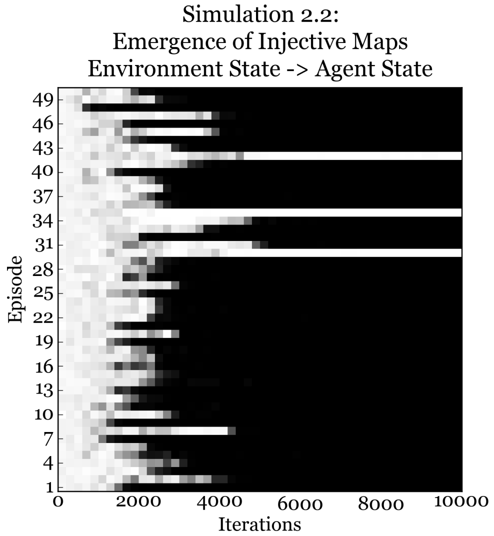


Figure 3.3.4: Simulation 2.2: Reward histogram. Different to the setup with learning only a single model, the reward at first (up to ca. iteration 1500) does not move away from chance level (0.2). In this stage, the behaviour of the system can be described as a stochastic search, trying to find a combination of Competence and Reception model firings, that yield an above average (above chance) reward. Once a stable causal chain is found, the model quickly converges to a stable model.

The evolution of the average reward in Figure 3.3.4 shows, that after an initially low value (at chance level), the reward quickly rises, as the two models find mutually rewarding mappings for certain combinations of desired and observed agent states . In the beginning, the learning very much works like a stochastic optimization, because the *desired state / action / perceived state* combinations are tried randomly (due to the stochastic nature of the employed learning algorithm). Figure 3.3.5 shows the evolution of the Reception model, according to a certain global feature. When the Cognitive Body model successfully approximates each environment state with (at most one) agent state , then this relation constitutes an injective map from the former to the latter. We can calculate such a (deterministic) map from environment to agent state by chaining the sensor matrix with the deterministic policy of the Reception model. This map tells us, which environmental state each agent state (most likely) corresponds to. In the optimal case, this map is injective, i.e. all environmental states map to different agent states and are thus distinguishable within the agent. This property was used in subsequent experiments to quantify the success of an episode, as the direct assessment of the learnt conditional probabilities get very difficult with increasingly complex environments.

The left diagram in Figure 3.3.5 shows, that once a simulation develops an injective map between world and agent states (black pixel), it stays injective. An injective map between world and agent states constitutes an optimal solution, as all hidden states can be perfectly distinguished by the agent. Though, there are few episodes that do not develop stable injective maps. The right diagram shows a histogram over the marginal probability of injectivity. An episode either develops a constantly injective map (frequency=1.0), or it develops no injectivity at all (frequency=0.0).

Figure 3.3.6 shows a histogram of the number of agent states unassociated with an environmental one. Even in those few cases, where the map does not develop injectivity, it still can distinguish 3 states, and mangle the remaining two into one. This leaves one agent state unassociated. This



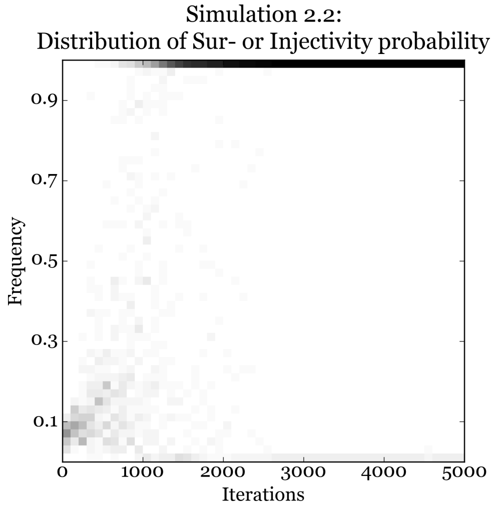


Figure 3.3.5: Simulation 2.2: Histograms of Injectivity probability (black=1.0 probability).
For each simulation step, the most likely map (deterministic policy) from environmental state to agent state (as implied by the Reception) model is calculated. The left diagram shows the frequency of injectivity as pixel darkness for each simulated episode (y axis) and simulation time (x axis). White denotes no occurrence, and black denotes constant occurence over the binned interval of iterations. The right diagram shows a histogram of the marginal frequency over all episodes with respect to simulation time. Frequency is calculated by binning and averaging the binary property.

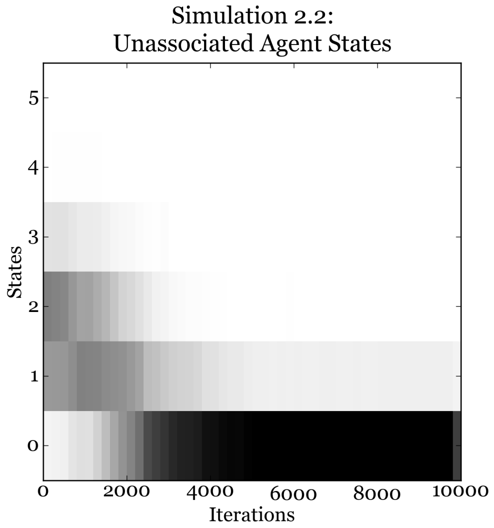


Figure 3.3.6: Simulation 2.2: Histogram of the number of “free” unassociated agent states.
If all environmental states can be distinguished, then the number of unassociated agent states is 0.

suggests, that the model learns *most* of the environmental structure, even in those cases where it does not learn it perfectly.

The statistics over multiple episodes show, that the two concurrent learning algorithms robustly develop a locally stable set of models (Figures 3.3.4, 3.3.5). Although the environmental structure is fixed in this simulation, there are several optimal solutions. They are homomorph, as they can be transformed into each other by permuting the model states, and the model’s synaptic weights accordingly. For analysis, the mapping of environmental state to the most likely agent state (using sensor matrix and Reception model) was extracted post simulation, to provide understandable labels of the model states in the figures.

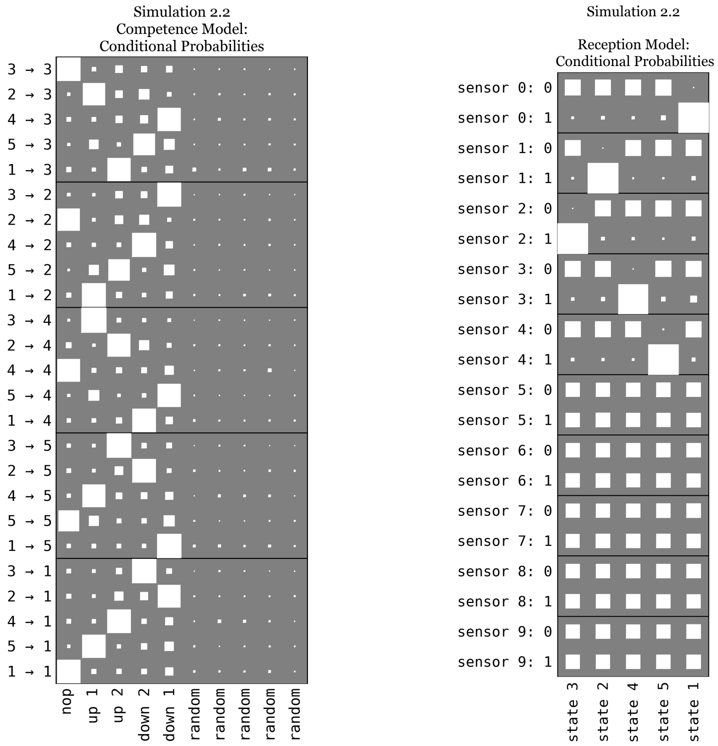


Figure 3.3.7: Simulation 2.2: Example for learnt conditional probabilities for selecting actions (left) and selecting observations(right) of a single episode.

The sensor matrix of the environment (Table 3.3) is trivial, so the reverse mapping of the Reception model (Figure 3.3.7) likewise is. The mapping of agent-internal Umwelt states (= model outputs) to the corresponding environmental states can arbitrarily be permuted. In the case of the simulated world, there are $5! = 120$ distinct models that are equally optimal given a matching permutation of the mapping of Umwelt states to actions. An interesting detail is the fact, that learning both models together increases the number of discoverable optimal models by n! (n being the number of Umwelt states), countering the necessary increase of dimensionality in model space (Reception models × Competence models) .

This simulation also shows an interesting behaviour in the convergence. Reward evolution (Figure 3.3.4) can be divided into two distinct stages. In first stage (iteration 0 to roughly 1500), the models do not show progress and stay close to the chance level of 0.2. Then, in the second stage, the average reward suddenly rises rapidly to the maximum average reward of about 0.8 (limited due to the model's probabilistic nature). A possible interpretation is, that In the first stage the models cannot easily converge. They initially have to rely on the lucky coincidence of a matching state observation and action selection to achieve an above-chance reward. In the second stage, the focus shifts on rather filling the holes of the models, e.g. to reliably distinguish states from others and select efficient actions in different contexts (desired states). This partition of behaviour might occur, because of a progressing reduction of search space. When an agent state gets associated with an environment state, stochastic search concentrates on a model subspace, reducing the effective dimensionality of the remaining search.

	state 1	state 2	state 3	state 4	state 5
Sensor 0	1	0	0	0	0
Sensor 1	0	1	0	0	0
Sensor 2	0	0	1	0	0
Sensor 3	0	0	0	1	0
Sensor 4	0	0	0	0	1
Sensor 5	1	0	0	0	0
Sensor 6	1	1	0	0	0
Sensor 7	1	1	1	0	0
Sensor 8	1	1	1	1	0
Sensor 9	1	1	1	1	1

Table 3.4: Simulation 2.3: Sensor activation w.r.t. environmental states. Different to simulation 2.2, sensor 5 to 9 also include some information about the environmental states.

Simulation 2.3: Sanity Check

In this simulation, the goal was to test, whether the performance of the previous simulation was only a lucky strike with respect to the chosen environment, and therefore replaced the environment with a more complicated version. The agent side is not changed, i.e. the full Cognitive Body model is used, with learning rate $\eta = 0.1$. The statistics are calculated over 50 episodes.

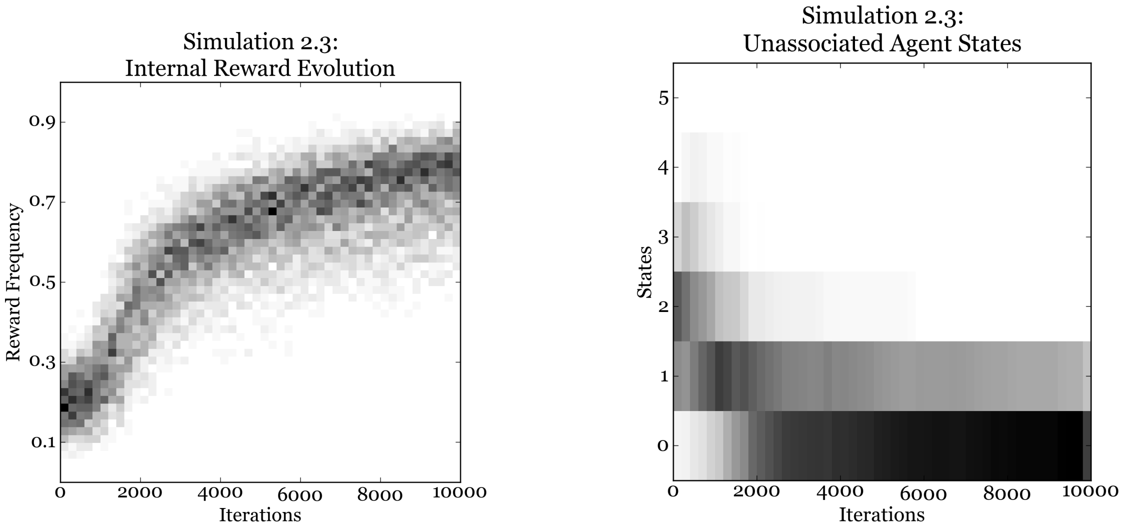


Figure 3.3.8: Simulation 2.3: Histogram of reward and unassociated states. The system converges quicker to the maximum reward than simulation 2.2 (Figure3.3.4), but the final distribution shows a fatter lower tail due to more episodes having unassociated states.

The results show a better reward histogram (Figure 3.3.8) than the previous simulation. As the sensors provide more (redundant) information about the environmental state, the reception model's updates lead to bigger cumulated weight changes. The histograms in Figure 3.3.9 tell a different story, more episodes failed to develop an injective map.

Figure 3.3.10 quantifies this difference. Overall, the differences to the previous simulation are modest. Inspecting the conditional probabilities of a learnt Reception model (Figure 3.3.11) tell us, that the model learnt to use all available information, and not just a subset of the sensors data.

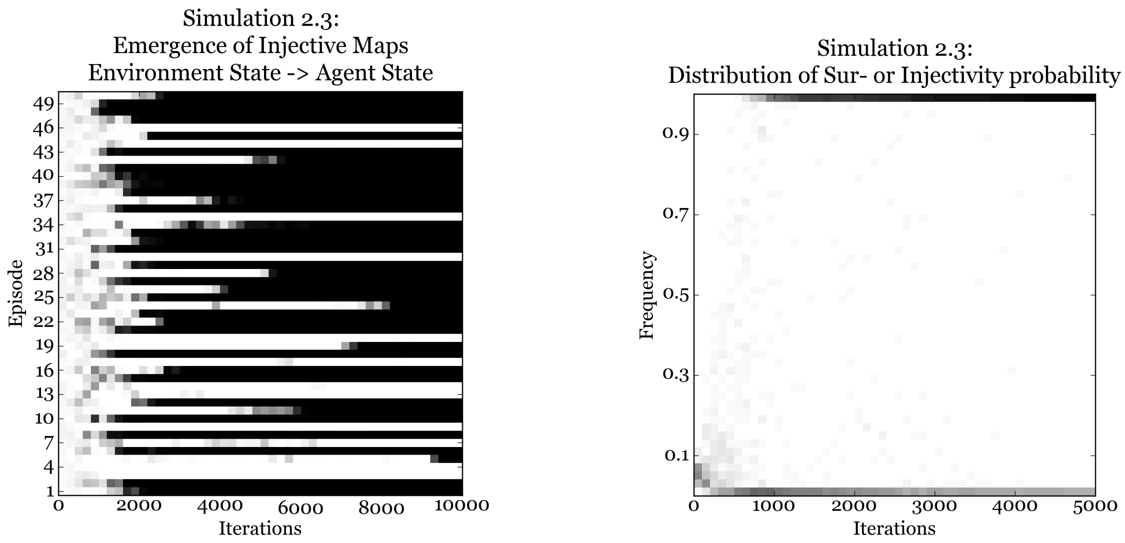


Figure 3.3.9: Simulation 2.3: Histograms of Injectivity probability.

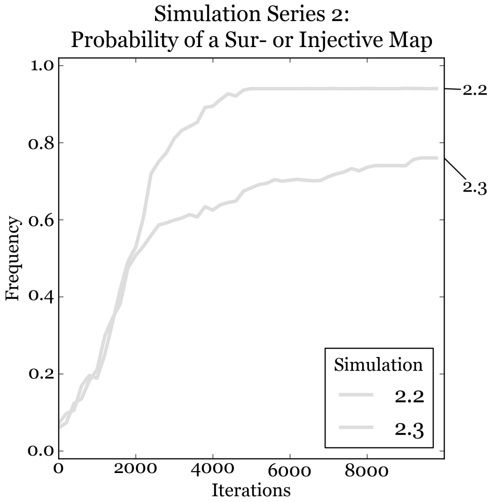


Figure 3.3.10: Simulation 2.3: marginal probability of Injectivity in comparison to simulation 2.2.

Simulation 2.4: Stability with a Random Sensor Matrix

One of the claims of the Cognitive Body model also is, to be able to cope with unknown (thus arbitrary) sensor and actor mappings. In this experiment, the sensor matrix was replaced by a randomly generated matrix. A random replacement of the state transition table was not yet included, as to make informal interpretation (and error checking) of the learnt behaviour tractable, enabling sanity-checking the simulation. Also, from a probabilistic point of view, doing two randomizations consecutively instead of one does not change the probability distribution in model (search) space.

Of course, the model can only learn an optimal (bijective) mapping, if it is at all possible to distinguish every environmental state based on the generated sensor signals. Such a degenerated sensor matrix would change the maximum attainable reward of the agent, and thus not be directly comparable to the previous experiments. To avoid this, generated matrices of Rank less than 5 (the number of environmental states) were discarded. The rank was calculated numerically using Singular Value Decomposition. Matrices containing close to zero eigenvalues ($\lambda_i < 0.1$) where discarded. This effectively ensures that the rank of the sensor matrix equals the number of environmental states. Thanks to the intentionally sparse encoding of 5 environmental states into 1024 distinct sensor state vectors, only very few randomly generated matrices actually have to be discarded.

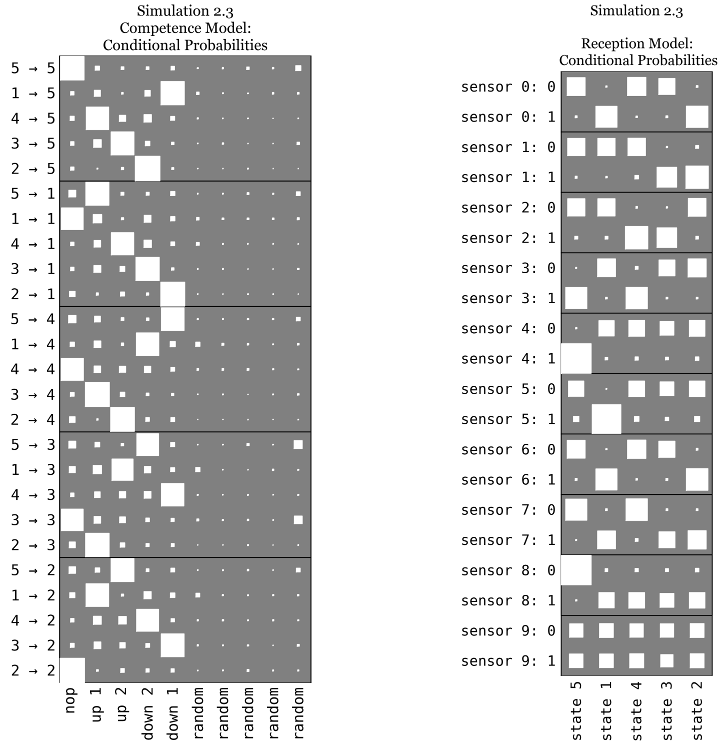


Figure 3.3.11: Simulation 2.3: Example for learnt conditional probabilities for selecting actions (left) and selecting observations(right) of a single episode.

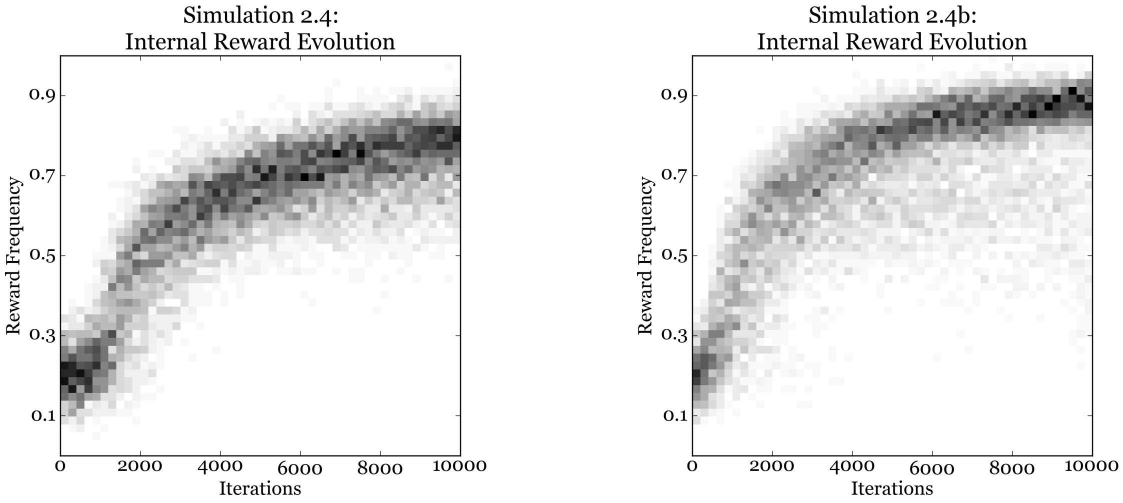


Figure 3.3.12: Simulation 2.4 and 2.4b: Reward evolution , with a randomly chosen sensor matrix of rank 5. (left: $\eta = 0.1$, right: $\eta = 0.25$)

Figure 3.3.12 shows the reward evolution of simulation 2.4. The left diagram shows no surprises, and behaves similar to previous simulations. This validates the hypothesis, that the Cognitive Body model can in principle learn any sensorimotor contingencies, if they have the same algorithmic complexity as the class of Finite State Machines.

The right diagram shows a the statistics for an even more aggressive learning rate. It speeds up the establishment of mutual reward in the beginning, but the wide lower tail of the histogram in later stages of the simulations indicate, that the models are not as stable. Though, the right diagram in Figure 3.3.14 (frequencies of injective maps) tells a different story. Learning performance is only increased in the beginning, and the advantage diminishes with further iterations. This indicates,

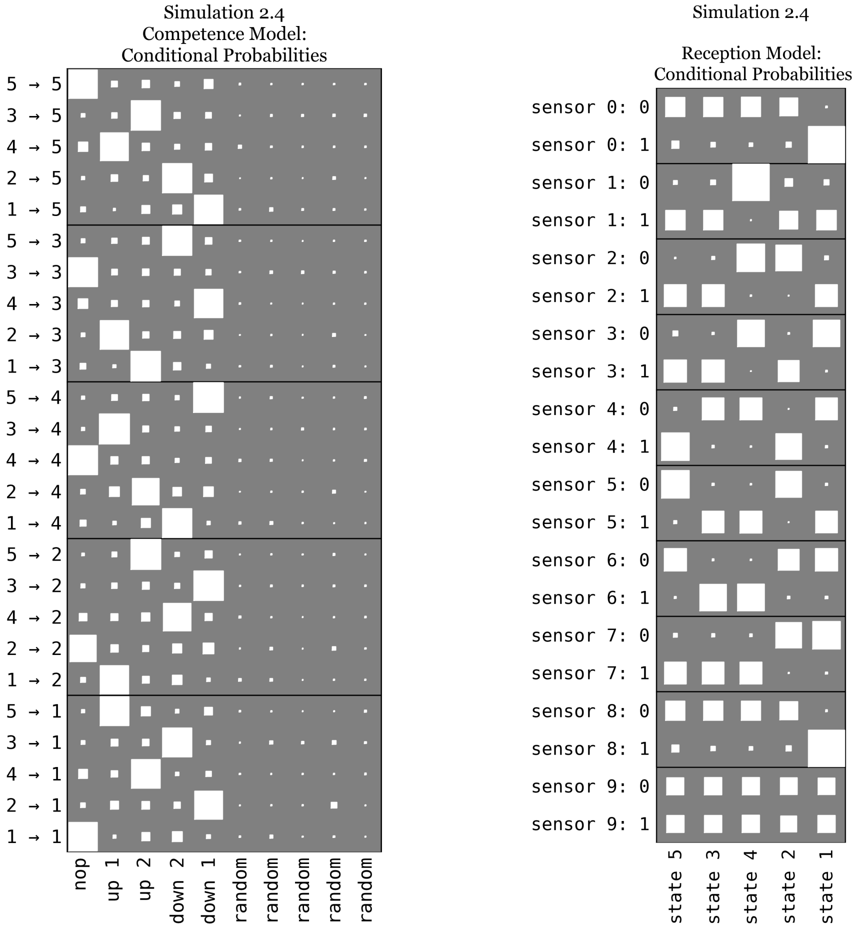


Figure 3.3.13: Simulation 2.4: Example of learnt conditional probabilities of the Competence and Reception model of a single simulation instance.

that the number of non-converging episodes of Simulation 2.4 cannot be reduced by increasing the learning rate (as would be expected with an exploration-exploitation trade off).

Figure 3.3.13 shows the outcome of one simulated episode. The agent successfully learns the mapping of the observed/desired Umwelt state combination to the facilitating action(s).

Due to the arbitrary map of environmental to agent states, the labels of the agent states have to be calculated on the fly, and can (and indeed do) change during a simulation. To create the labels, an optimal deterministic policy was computed from the Reception model. The agent states labels denote the most likely environmental states that the specific agent state represents (both perceived and desired). A label of “?” was used for agent states that never are most likely selected given any certain environmental state, whereas e.g. “2,3” signifies an agent state that is most likely selected given environmental state 2 or 3.

Due to the random nature of the sensor matrix, the learnt probabilities of the Reception model in Figure 3.3.13 (right diagram) is not intuitively interpretable. Therefore, the analysis has to rely on the previously introduced implicit properties of unassociated states and injectivity, shown in Figure 3.3.14

Overall, the results of simulation 2.4 revealed, that quite a few episodes do not converge to an optimal solution (thus not yielding a surjective map). Though, the distribution of unassociated states (Figure 3.3.14) shows that in all those episodes, there are only 2 states mangled (leaving one unassociated state). The agent still learnt a model, that mirrors most of the environment. Nevertheless, it is an interesting observation, that the model does not necessarily converge to an

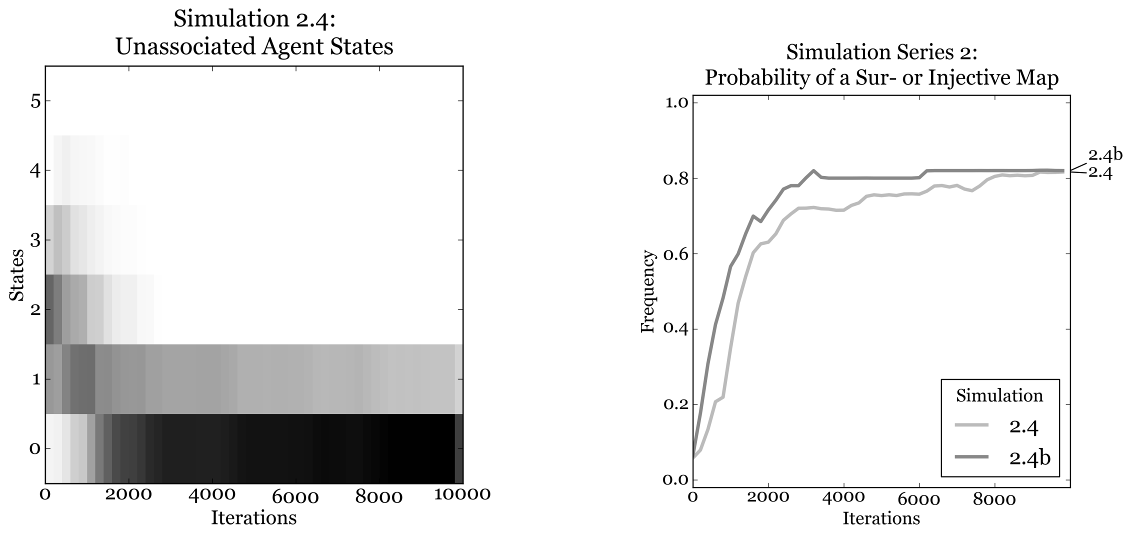


Figure 3.3.14: Simulation 2.4: Histogram of unassociated agent states, and the marginal probability of injectivity with respect to. different learning rates (2.4:$\eta = 0.1$, 2.4b: $\eta = 0.25$)

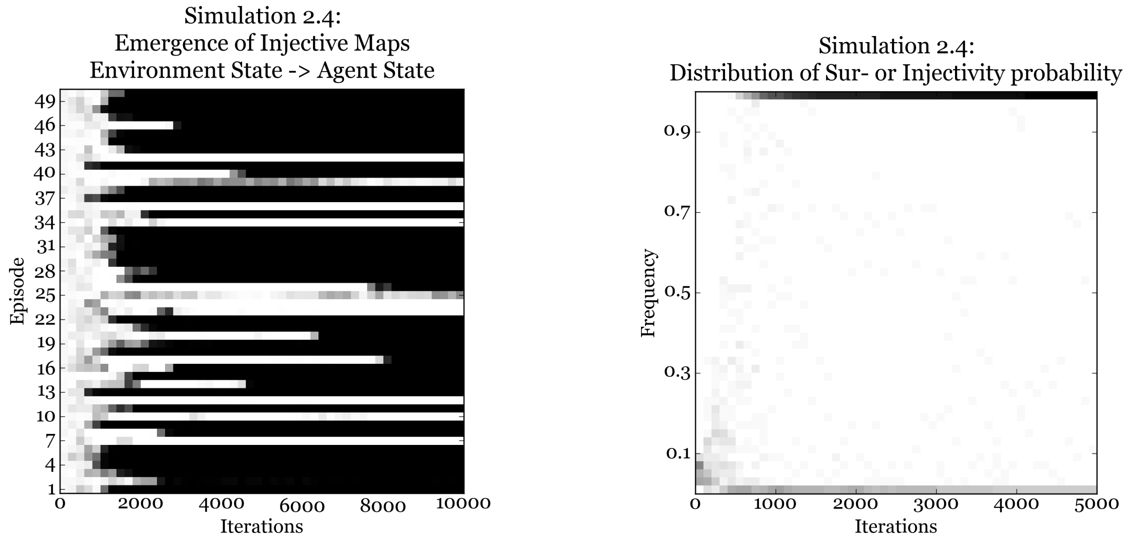


Figure 3.3.15: Simulation 2.4: Evolution of injectivity. The majority of episodes develop an injective map after about 2000 iterations.

optimal solution, i.e. the search space is all but free from local suboptimal minima. Further research could focus on events during learning that cause the Cognitive Body model to succeed developing a surjective map or not.

Simulation 2.5: Stability with Random Sensor and State Transition Matrix

In this experiment, we both randomly initialize the sensor matrix and the state transition matrix of the environment, thereby relaxing the environment to be an arbitrary 5-state Finite State Machine. The same considerations as in simulation 2.3 apply. Though, because of the completely random nature, we cannot intuitively assess the performance of the Competence model any more (e.g. by checking the sensibility of learnt preferred actions). Therefore we have to rely on computed mathematical properties of the (most likely) maps implied by the probabilistic models, and the reward evolution.

As with simulation 2.4, one constraint was put on the randomly generated environments. Both the transition and the sensor matrix were required to keep all world states separable in principle, i.e. no essential information about the environmental states was lost.

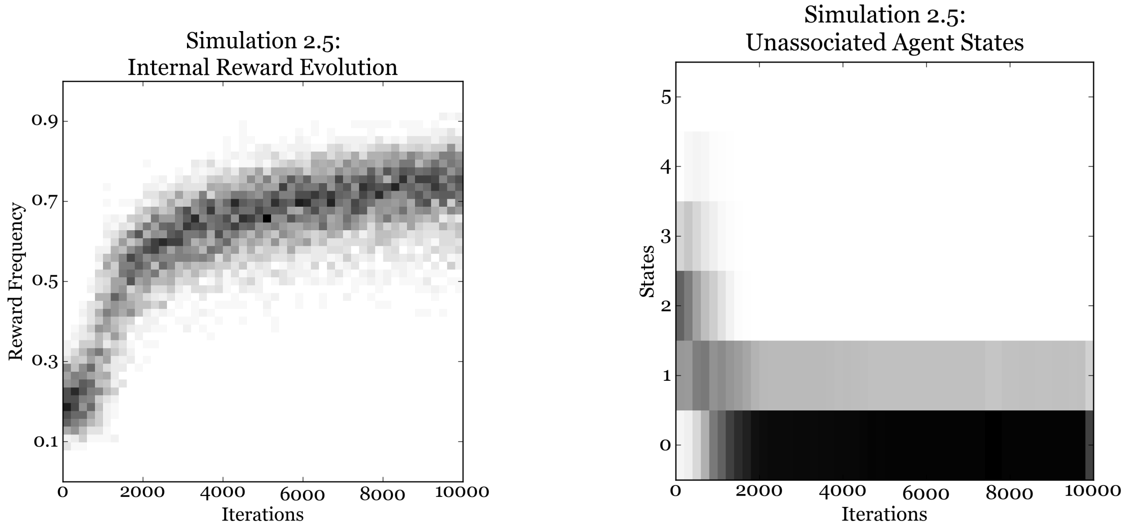


Figure 3.3.16: Simulation 2.5: Histogram of reward and unassociated states.The performance is similar to simulation 2.4, confirming the expectation, that randomization of the transition function additional to the sensor matrix does not change the environment's complexity.

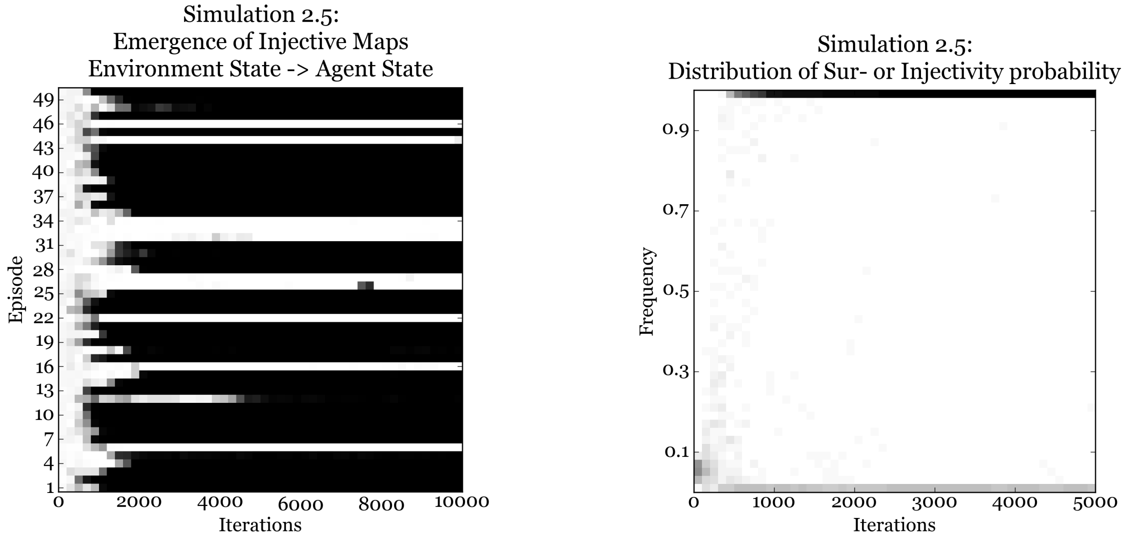


Figure 3.3.17: Simulation 2.5: Evolution of injectivity. The data are very similar to the ones obtained in simulation 2.4.

The results of simulation 2.5 are almost identical to the ones in simulation 2.4. This implies, that the Cognitive Body model relies neither on specially crafted actions nor on special sensor configurations in its environment (as long as they provide enough information to make the states separable). As with the simpler environments of previous simulations, most (roughly 4 out of 5) episodes quickly converge, while the rest take an exceptionally long time. This failed convergence on an injective map does not happen because of certain "bad" environments, as the results of simulation 2.2 and 2.3 show. This happens also, when a single environment is simulated repeatedly.

To investigate this phenomenon of non-convergence, the experiment was run for an extended period (200'000 iterations) with an increased number of episodes (1000). The results are shown in Figure 3.3.18 on the next page. The dotted line shows a fitted power law function $p(i) = 32.0 * i^{-0.54}$. The diagram suggests, that the convergence behaviour can be modelled with a Pareto distribution. This distribution is heavy-tailed, and we can not calculate an expectation value for the time until

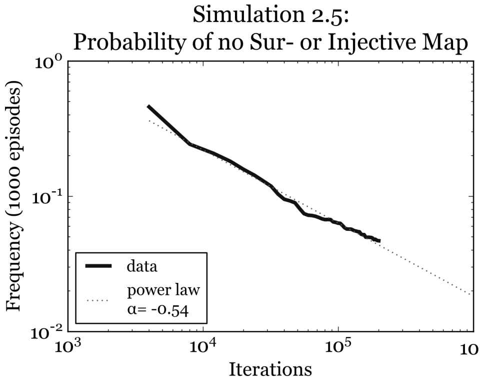


Figure 3.3.18: Simulation 2.5, long run: Probability of an episode *not* developing injectivity w.r.t. learning time. (1000 episodes, 200'000 iterations)

an episode develops injectivity (as $\alpha > -1$).

If calculation of the injectivity property was possible by the agent (it isn't), we could easily improve the convergence speed by *resetting* unsuccessful episodes after a fixed amount of iterations. This would lead to a geometric series with a finite expectation value. Unfortunately, the agent cannot decide, which model has developed an injective map, because it cannot access the environment state. To approximate the reset trick, the agent could operate a*n ensemble* of models, and periodically reset all but the best performing one, as measured by the internally available reward signals. This way, an already optimal solution will be preserved during resets, as it will yield the highest possible reward. This Ensemble Learning could be stopped as soon the reward does not increase for several resets, making further increases highly improbable.

Conclusions to Simulation Series 2

The Simulation series 2 aimed to construct step by step an example application of the Cognitive Body model. By successively removing constraints, the experiments show, that an Agent can learn to fully control and observe an arbitrary 5-state Finite State Machine.

A model was defined to be optimal, when each agent state is most likely to be selected by the Reception model in at most one environment state. This definition implies, that the calculable deterministic policy of the Reception model results in an injective map from environment state to agent state.

Figure 3.3.19 on the following page shows the success of all simulation according to the probability of developing an injective map. The two hand selected environments (simulation 2.2 and 2.3) over- and underperform respectively the generic case (simulation 2.5). Simulation 2.4, 2.4b and 2.5 show a similar evolution.

The simulations revealed, that in the majority of cases, the Cognitive Body model develops an implicit injective map. This only takes a moderate amount of iterations, usually around 2000-3000, which translates to roughly testing every state-transition/action pair ($5 \cdot 5 \cdot 10$ combinations) 10 times.

The typical evolution of the reward can be narrated to the Cognitive Body model first randomly trying (reward at chance level), but then having an "Aha moment", where reward quickly rises. The (average) development then reaches a point, where all states of the environment have a distinct and unique corresponding state pair within the agent. Still, the experiments also pose questions, such as why a substantial amount of episodes (roughly 10-20%) only come close to the optimum, but do

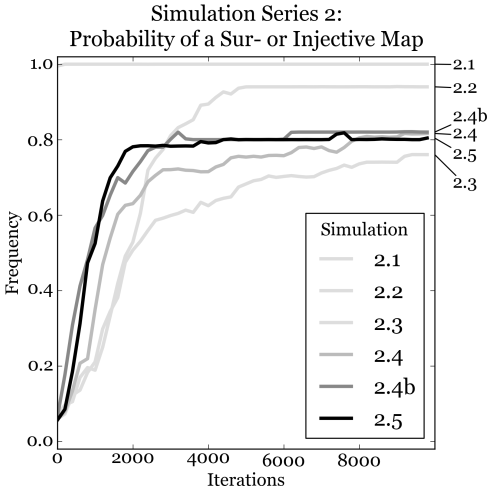


Figure 3.3.19: Simulation series 2: quantitative comparison of marginal probabilities of all simulations.

not actually reach it. The cumulative distributions of the mathematical properties of the (model-implied) deterministic decision policies clearly show that this is not an artefact of an asymptotic (stochastic) behaviour, but rather indicates a "fat tail" distribution.

Once an optimal (injective) map between environment and agent states is established, the system keeps this property, even with a quite high learning rate of $\eta = 0.25$ and substantial stochastic noise, due to the Soft Winner-Take-All stages of the used algorithms and the "random" actions available in the environment. An important aspect is, that this stability is reached even without an online adaption of the learning rate. Thus, the agent is not locked in to a certain environment, but will adapt its models, if it changes.

A weak point of the Cognitive Body model is revealed by the probability distribution for an episode not developing an injective map. A long term simulation suggests, that it likely follows a Pareto distribution. Its coefficient roughly is $\alpha = 0.5$, and therefore no sensible expectation value can be computed. This is an undesirable behaviour, and suggests that improvements can be made to the learning strategy, to yield an exponentially fast convergence .

An important constraint was put on the type of environment. The agents models capability to represent 5 states was matched to the environment's 5 states beforehand. Thus, this simulation series explicitly does not address behaviours and problems related to differing numbers of states. This is addressed in the next simulation series.

3.4 Exploring Incongruities Between Umwelt and Environment

One remaining crutch of the previous exploratory simulations is, to predefine the amount of states for the agent to model. Of course, in a realistic setting, knowing the right amount of states is a near impossibility, and there are disadvantages arising from both having too few or too many states. Simulation series 3 was conducted to explore the behaviour of the Cognitive Body model under such conditions. Another possible source for incongruities between Umwelt and environment are different mathematical frameworks. It is quite conceivable, that the environment follows a set of differential equations, while the agent tries to explain them by difference equations, or via a Finite State Machine.

In the case of the model having less states than the environment, one would expect the model to "lump together" states that are marginally different, and simply not differentiate between those.

Depending on the exact structure of the environment, this might be associated with a loss of predictability of action outcomes.

In the case of too many agent states for a given environment, one would expect an unstable mapping of states, as several agent states would compete for representing the same environmental one, and their probability distributions within the models would converge. A sophisticated algorithm might leverage this fact for pruning unneeded states (and thus, computational complexity) from both the Competence and Reception model.

The simulations use the same experimental setup as simulation 2.5. Statistics are computed over 50 episodes, learning rate $\eta = 0.1$, and the environment is a randomly drawn 5-state Finite State Machine (with all states distinguishable/reachable). The simulation was run with 4, 5, 6, 7, 8, 12 and 20 agent states available to the agent. The simulations are numbered 3.1 to 3.7 respectively.

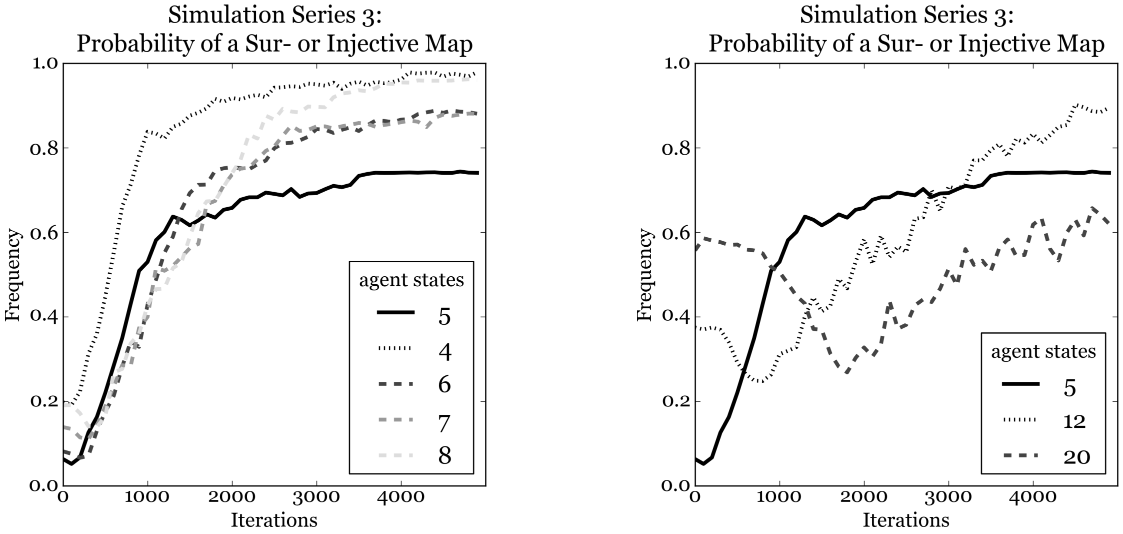


Figure 3.4.1: Simulation series 3: Evolution of injectivity with respect to number of used agent states (surjectivity for 4 agent states).

Figure 3.4.1 shows a comparison of the evolution of in- or surjectivity for each simulated number of agent states. the 5-state curve resembles the base case of the previous experiments.

The 4-state model has one less state than needed to fully mirror the structure of the environment, so, it can also not develop an injective map. Nevertheless, we can use the surjectivity property to assess, whether the model associated all its states with environmental states, which is the best possible outcome. Not surprisingly, a model with too few states converges faster to a surjective map, as there are fewer combinations to learn, and hard to discern states can stay mangled together.

The simulations with 6, 7, and 8 agent states show interesting behaviour. Figure 3.4.1 shows, that models having spare states develop injectivity roughly as fast as the 5-state base case. So, the number of agent states does not greatly influence the speed of mirroring the environment. Additionally, significantly more episodes develop injectivity in the long run if there are spare states available in the agent. Though, the right diagram shows, that for models that use many more states than necessary, this advantage breaks down.

Inspection of the evolution of single episodes (Figure 3.4.2, left diagram) reveals, that the Reception models do develop injective maps (darkness of pixels), but these maps are not stable, because it does not fit to the states learnt by the Competence model (indicated by the low mutual reward). Simulation 3.7 with 20 states behaves similarly to 3.6 with 12 states. Overall, these observations suggest, that it is better to have a slightly bigger model within the agent than strictly necessary.

When dealing with spare agent states, what happens with them in a model with a developed injectivity?. One hypothesis is, that the conditional probabilities for such a spare state are rather

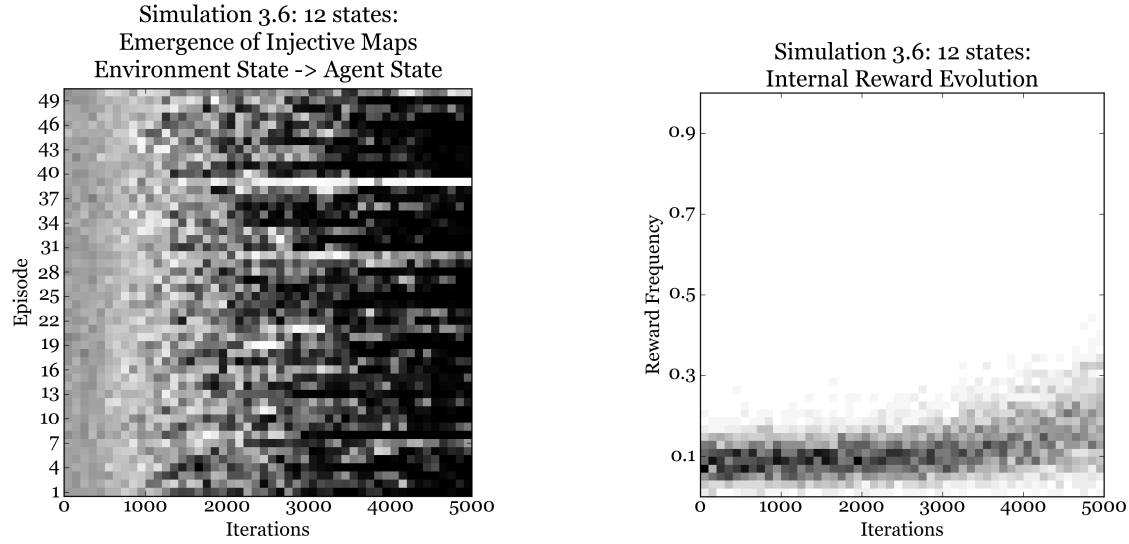


Figure 3.4.2: Simulation 3.6: Evolution of injectivity, and histogram of reward. The reward stays at chance level ($\frac{1}{12}$) for many times longer than compared to the simulations with less states.

uniform, due to the states not having any correspondence and thus the occasional occurrence is totally random. Another hypothesis is, that the models develop conditional probabilities similar to those of another (associated) state. This would be bad, because then, two or more agent states would compete with each other to represent the same environment state - and the most likely state would perpetually fluctuate.

Figure 3.4.3 shows the conditional probabilities of one episode each from simulation 3.3 and 3.4. The columns with the label "state ?" denote the spare states. It is easy to see, that all conditional probabilities are very small, indicating that the state is not likely to be selected with any sensor input. The Competence model shows a similar, but less pronounced behaviour. Figure 3.4.4 shows the learnt conditional probabilities of one episode each of simulation 3.3 and 3.4.

We can draw the conclusion, that the spare states are not duplicates of other associated states. Because of their distinct pattern of conditional probabilities, these states could easily be pruned from a model.

Conclusions to Simulation Series 3

Simulation series 3 explored the behaviour of the Cognitive Body model when the number of agent states differs from the number of states in the environment. Models with a few more states than necessary show a higher probability of optimally mirroring the environment states than the base model with an equal number of states. The advantage breaks down, when there are many more agent states available, as shown in simulation 3.6 (12 states) and 3.7 (20 states).

Cognitive Body models that have fewer states than needed, still exhibit favourable dynamics, and quickly associate all agent states with a corresponding environmental one. This observation suggests, that a Cognitive Body model can start with too few states, and then gradually add agent states to the Competence and Reception models, until some states exhibit the behaviour of spare states.

The simulation series only explored cases where both environment and the models within the agent operate on the same mathematical framework (Finite State Machines, in this case). It is quite conceivable, that e.g. the environment follows a set of differential equations, while the agent tries to explain them by difference equations. It is unknown, whether the Cognitive Body model leads to sensible approximations in those cases.

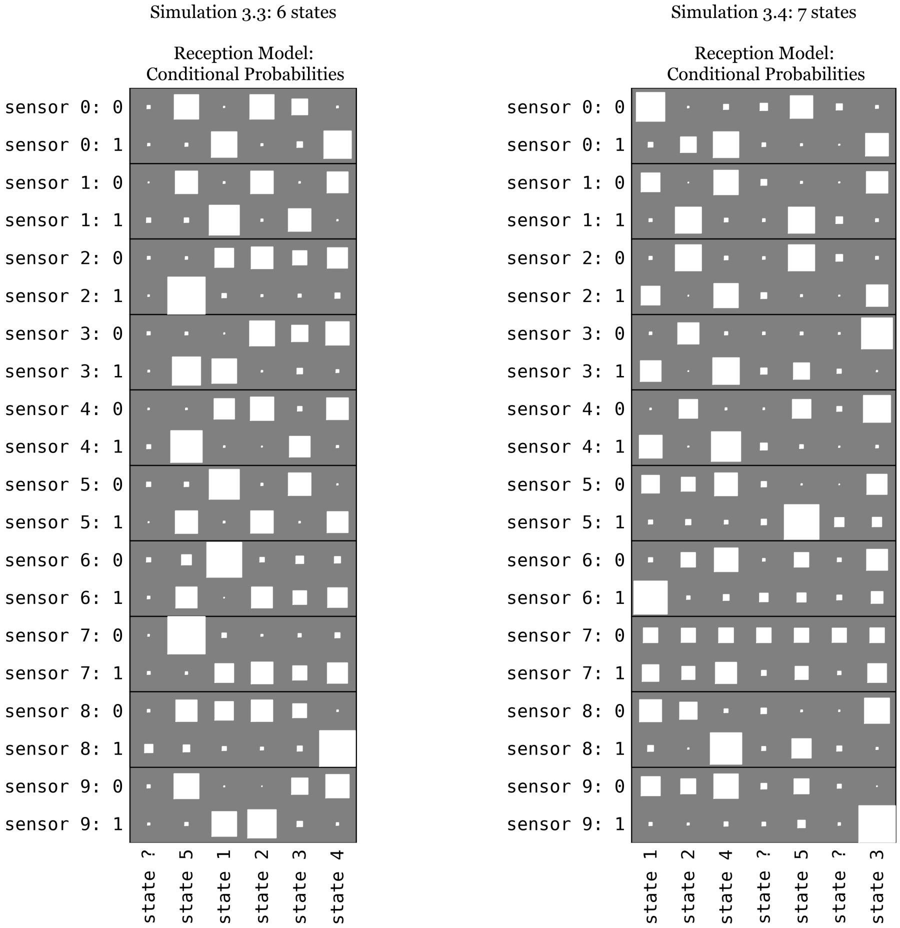


Figure 3.4.3: Simulation 3.3 and 3.4: Examples of conditional probabilities of the Reception model for 6 and 7 states.

3.5 Conclusions and Open Questions

The conducted experiments explored the application of the Cognitive Body model on autonomously learning a restricted but nontrivial simulated environment with its intrinsic states hidden to the agent. The simulations showed, that the Cognitive Body model is viable, and can match internal models to an environmental structure for the purpose of transparent access and control. The use of aggregate mathematical properties (injectivity and surjectivity) of maps proved to be efficient and a selective criterion to evaluate the approximation process. Though there is a significant number of cases, where the hidden environmental states are not perfectly separated within the agent, they are at least close to an optimal solution, quantifiable by the number of unassociated agent states.

The experiments in simulation series 2 showed, that an agent implementing the Cognitive Body model can autonomously learn to access the hidden states of any arbitrary 5 state Finite State Machine (iff information-theoretically possible) within a modest amount of iterations, including probabilistic actions. The histograms of the reward evolution can be interpreted to show “Aha” moments, after which reward, starting from the initial chance level, suddenly increases. A weak point is the low (ca. 0.7-0.8) probability of finding an optimal solution (injective map from environment to agent state) within a limited amount of time, when there is an equal number of agent and environment states. This can be remedied by providing “spare” agent states, and possibly by Ensemble Learning.

The agent states also exhibited stability with respect to their semantic meaning, which are grounded by their co-occurrence with the environmental states. This was observed with a relatively high and constant learning rate of $\eta = 0.1$. Because learning is not stopped, the models would still be able

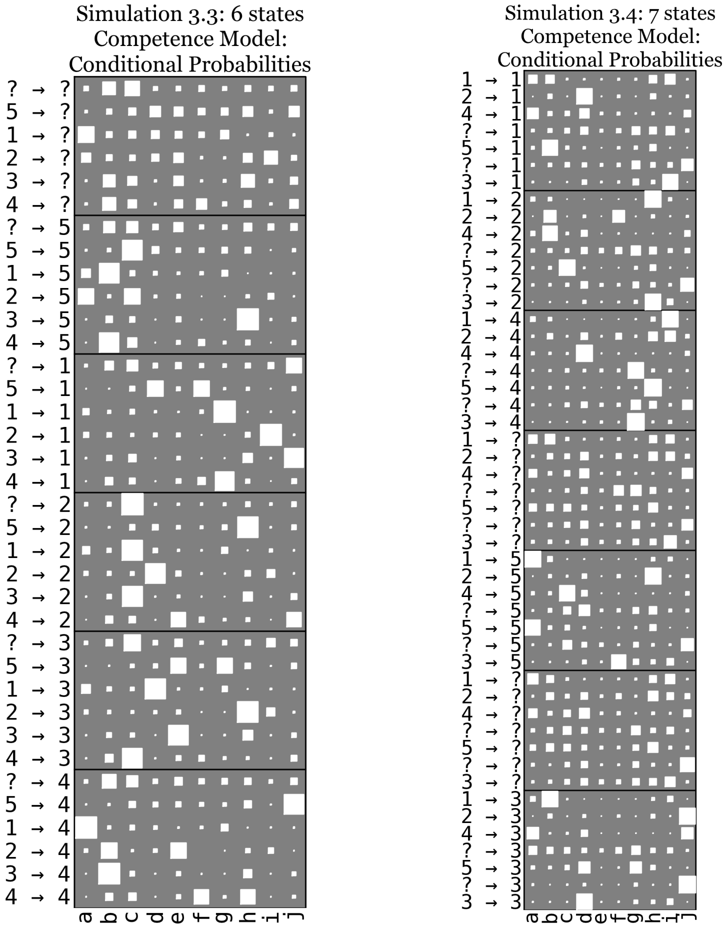


Figure 3.4.4: Simulation 3.3 and 3.4: Examples of conditional probabilities of the Competence model for 6 and 7 states.

to quickly reorganise to fit a changed environment, e.g. in case of an injury.

An influence difficult to predict was the number of agent states with respect to the number of environment states. Simulation series 3 showed, that the Cognitive Body model works even better when the agent model is slightly bigger in terms of states, than the environment. Also, using a model smaller than the environment still led to a favourable behaviour (associating as many states as possible) of the Cognitive Body model. Performance degraded badly, when the agent models grew over twice as big as the environment. The cause of this deterioration in performance was not investigated.

The experiments also focused on the application of a single learning algorithm, Reward-modulated Hebbian Learning (Pfeiffer *et al.* , 2010), primarily because of reducing the possibility of bugs. The Cognitive Body model itself is agnostic to the kind of learning algorithm used, and even to the model space which is searched in, as long as the algorithm optimizes the criteria formulated in Section 2.4. Especially, the Competence models could also learn action sequences instead of single actions. Suitable implementations could e.g. be Recurrent Neural Networks, or methods of Reservoir Computing (Jaeger, 2010; Maass *et al.* , 2002; Lukoševičius & Jaeger, 2009). Other promising venues of research would be to integrate methods optimizing the scalability by adding and pruning states, introducing competition between states, or adding elements of unsupervised learning methods to bootstrap the Cognitive Body model. A candidate hybrid algorithm is the *contextual Slow Feature Analysis* algorithm (Deimel, 2009).

Following the empirical research on stability of the Cognitive Body model, it is unclear whether a hierarchical, stacked model as proposed in Section 2.8 is itself stable. The stacking of another model on top an already learnt (and stable) one can influence the lower level model by changing the probability distribution of desired states, and the change of the lower level models can in turn influence the structure of the upper models. It is unknown whether this interaction is beneficial or disadvantageous to the performance of the Cognitive Body model.

Due to time constraints, only simulations of virtual environments were conducted. It would also be prudent to also run experiments with real robots, to see whether the Cognitive Body model also yields sensible results in really unknown environments.

Bibliography

Barber, David. 2011. *Bayesian Reasoning and Machine Learning. by David Barber.* Cambridge University Press. 18

Brooks, Rodney A. 1991. Intelligence without representation. Artificial Intelligence, no. 47. 11, 14

Clark, Andy. 1996. *Being There: Putting Brain, Body, and World Together Again.* Cambridge, MA, USA: MIT Press. 11, 14

Clark, Andy. 2008. *Supersizing the Mind: Embodiment, Action, and Cognitive Extension (Philosophy of Mind Series).* OUP USA. 14

Cristianini, Nello, & Shawe-Taylor, John. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* 1 edn. Cambridge University Press. 33

Deimel, Raphael. 2009 (June). *Contextual Slow Feature Extraction Framework.* Tech. rept. Österreichisches Forschungsinstitut für Artificial Intelligence, Wien. 33, 60

Dennett, Daniel C. 2004. *Consciousness Explained.* Gardners Books. 17

Friston, Karl. 2010. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, **11**(2), 127–138. 13

Friston, Karl J., Daunizeau, Jean, & Kiebel, Stefan J. 2009. Reinforcement learning or active inference? *PloS one*, **4**(7), e6421+. 13

Grush, Rick. 2004. The emulation theory of representation: motor control, imagery, and perception. *Behavioral and Brain Sciences*, **27**(3). 22

Harnad, S. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, **42**(1-3), 335–346. 12

Jaeger, H. 2010. *The "echo state" approach to analysing and training recurrent neural networks - with an Erratum note.* Tech. rept. German National Research Center for Information Technology. 60

Kotz, S., Balakrishnan, N., Read, C. B., & Vidakovic, B. 2006. Multivariate symmetry and asymmetry. *Pages 5338–5345 of: Encyclopedia of Statistical Sciences, Second Edition*, vol. 8. Wiley. 67

Lamport, Leslie, Shostak, Robert, & Pease, Marshall. 1982. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, **4**, 382–401. 27

Lukoševičius, Mantas, & Jaeger, Herbert. 2009. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, **3**(3), 127–149. 60

Maass, Wolfgang, Natschläger, Thomas, & Markram, Henry. 2002. Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*, **14**(11), 2531–2560. 60

Marr, D., Lal, S., & Barlow, H. B. 1980. Visual Information Processing: The Structure and Creation of Visual Representations [and Discussion]. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, **290**(1038), 199–218. 12

Maturana, Humberto, & Pörksen, Bernhard. 2008. *Vom Sein zum Tun. Die Ursprünge der Biologie des Erkennens.* Heidelberg: Carl-Auer-Syteme Verlag. 19

Montebelli, Alberto, Lowe, Robert, & Ziemke, Tom. 2009. The Cognitive Body: From Dynamic Modulation to Anticipation. *Chap. 8, pages 132–151 of:* Pezzulo, Giovanni, Butz, Martin, Sigaud, Olivier, & Baldassarre, Gianluca (eds), *Anticipatory Behavior in Adaptive Learning Systems.* Lecture Notes in Computer Science, vol. 5499. Berlin, Heidelberg: Springer Berlin / Heidelberg. 17

Noe, Alva. 2009. *Out of Our Heads: Why You Are Not Your Brain, and Other Lessons from the Biology of Consciousness.* First edn. Hill and Wang. 11, 14, 17

Oliphant, Travis E. 2007. Python for Scientific Computing. *Computing in Science and Engineering*, **9**(3), 10–20. 31

O'Regan, J. K., & Noë, A. 2001. A sensorimotor account of vision and visual consciousness (with commentary). *Behavioral and Brain Sciences*, **24**(5). 14

Pfeiffer, Michael, Nessler, Bernhard, Douglas, Rodney J., & Maass, Wolfgang. 2010. Reward-Modulated Hebbian Learning of Decision Making. *Neural Computation*, **22**(6), 1399–1444. 33, 34, 43, 60

Philipona, D., O'Regan, J. K., & Nadal, J. P. 2003. Is There Something Out There? Inferring Space from Sensorimotor Dependencies. *Neural Computation*, **15**(9), 2029–2049. 9, 71, 72

Philipona, D., O'Regan, J. K., P, M, O. J., & Coenen, D. 2004. Perception of the structure of the physical world using unknown multimodal sensors and effectors. *Pages 945–952 of: Advances in Neural Information Processing Systems.* 11, 14

Russel, S., & Norvig, P. 2002. *Artificial Intelligence: A Modern Approach.* Prentice Hall. 11, 18

Searle, John. 1990. Is the brain's mind a computer program? *Scientific American*, **262**(1), 26–31. 12

Smith, Linda, & Gasser, Michael. 2005. The Development of Embodied Cognition: Six Lessons from Babies. *Artif. Life*, **11**(1-2), 13–30. 14

Sutton, Richard S., & Barto, Andrew G. 1998. *Reinforcement Learning: An Introduction.* 1st edn. Cambridge, MA, USA: MIT Press. 33

von Förster, Heinz, & Pörksen, Bernhard. 1997. *Wahrheit ist die Erfindung eines Lügners Gespräche für Skeptiker.* Heidelberg: Carl-Auer-Syteme Verlag. 11

von Uexküll, Johann J. 1934. *Streifzüge durch die Umwelten von Tieren und Menschen: Ein Bilderbuch unsichtbarer Welten.* Berlin: J. Springer. 18, 19

Appendix A

Mathematical Tools

A.1 Measures of Similarity in Information Theory

This section explains some basic measures of similarity used in this thesis. All measures can also be defined for continuous variables, though here, only the definitions for discrete variables are covered.

Entropy

The information-theoretic Entropy (*Shannon Entropy*) measures the expected information content of a signal source or variable. For a discrete variable X with n possible values, and respective probabilities of occurrence $p\left(X = x\right)$, its entropy is defined as:

$$H\left(X\right) = -\sum_{x \in X} p\left(X = x\right) \log_2 p\left(X = x\right) \ (bit)$$

Properties

An entropy is always positive:

$$\forall X : \ H\left(X\right) \geq 0$$

The highest possible entropy is equal to a uniform distribution U_n (over n possible discrete values):

$$\forall X : \ H(U_n) = \log_2 n \geq H(X)$$

The lowest possible entropy occurs, when the variable always has a single value:

$$\begin{aligned} & H(X) = 0 \\ \Rightarrow \quad & p\left(X = x_i\right) = \begin{cases} 1 & i = c \\ 0 & i \neq c \end{cases} \ c \in [1 \ldots n] \end{aligned}$$

Intuitively, a lower Entropy means, that the distribution is less similar to a uniform distribution, and more predictable.

Conditional Entropy

Conditional Entropy is defined as the Entropy of variable, given the knowledge of the value of a second variable. For two discrete variables X,Y:

$$H\left(Y|X\right) = \sum_{x\in X, y\in Y} p\left(X{=}x, Y{=}y\right) \cdot \log_2 \frac{p\left(X{=}x\right)}{p\left(X{=}x, Y{=}y\right)} \;(bit)$$

Properties

A Conditional Entropy is always positive:

$$\forall X, Y:\; H\left(Y|X\right) \geq 0$$

A conditional Entropy is *not* commutative:

$$\exists X, Y:\; H\left(Y|X\right) \neq H\left(X|Y\right)$$

A Conditional Entropy always is bounded by the Entropy of the conditioned variable:

$$\forall X, Y:\; H\left(Y|X\right) \leq H\left(Y\right)$$

Kullback-Leibler Divergence

The Kullback-Leibler Divergence is a widely used measure of difference between two probability distributions. For the case of two discrete distributions P, Q:

$$D_{KL}\left(P\|Q\right) = \sum_i P\left(i\right) \log \frac{P\left(i\right)}{Q\left(i\right)}$$

and for the case of two continuous distributions P, Q:

$$D_{KL}\left(P\|Q\right) = \int_{-\infty}^{\infty} p\left(x\right) \log \frac{p\left(x\right)}{q\left(x\right)} dx$$

Where $p = \frac{dP}{dx}$ and $q = \frac{dQ}{dx}$ are the probability densities. Properties important of the KL-Divergence to subsequent arguments are:

The KL-Divergence always is a positive value:

$$\forall P, Q:\; D_{KL}\left(P\|Q\right) \geq 0$$

Zero Divergence means, that distributions are identical.

$$D_{KL}\left(P\|Q\right) = 0 \;\Rightarrow\; P = Q$$

For independent, marginal distributions P_1, P_2 and Q_1, Q_2, we can decompose the KL-Divergence of the joint distributions P_1P_2 and Q_1Q_2 into two additive KL-Divergences:

$$D_{KL}\left(P_1P_2\|Q_1Q_2\right) = D_{KL}\left(P_1\|Q_1\right) + D_{KL}\left(P_2\|Q_2\right)$$

In the special case of $P_2 = Q_1 = H$:

$$D_{KL}\left(P_1 H \| H Q_2\right) = D_{KL}\left(P_1 \| H\right) + D_{KL}\left(H \| Q_2\right)$$

Iff the distribution $P_1 H$ is *centrally symmetrical* (Kotz *et al.* , 2006), and thus identical to HP_1, we can decompose the KL-Divergence of the left hand-side into:

$$\begin{aligned} D_{KL}\left(P_1 H \| H Q_2\right) &= D_{KL}\left(H P_1 \| H Q_2\right) \\ &= D_{KL}\left(H \| H\right) + D_{KL}\left(P_1 \| Q_2\right) \end{aligned}$$

The same decomposition is possible given a centrally symmetric HQ_2 :

$$\begin{aligned} D_{KL}\left(P_1 H \| H Q_2\right) &= D_{KL}\left(P_1 H \| Q_2 H\right) \\ &= D_{KL}\left(H \| H\right) + D_{KL}\left(P_1 \| Q_2\right) \end{aligned}$$

as $D_{KL}\left(H \| H\right) = 0$, we can then conclude:

$$D_{KL}\left(P_1 \| Q_2\right) = D_{KL}\left(P_1 \| H\right) + D_{KL}\left(H \| Q_2\right)$$

if either $P_1 H$ or $Q_2 H$ are centrally symmetric distributions. This especially is the case, when either $P_1 = H$ or $Q_2 = H$, i.e. when the respective marginal distributions are identical, irrespective of their cross-correlation.

Using the assumption of central symmetry, we can compute an upper boundary for both KL-Divergences from/to state H, without knowing H:

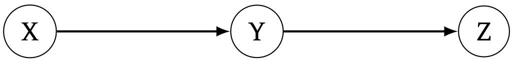
$$\begin{aligned} \left(P_1 = H\right) \vee \left(Q_2 = H\right) \Rightarrow \quad & D_{KL}\left(P_1 \| H\right) && \leq D_{KL}\left(P_1 \| Q_2\right) \\ & D_{KL}\left(H \| Q_2\right) && \leq D_{KL}\left(P_1 \| Q_2\right) \end{aligned} \tag{A.1.1}$$

Data Processing Inequality

The *Data Processing Inequality theorem* states, that any transformation of a parametrized distribution X yielding distribution Y cannot have a higher *Ali-Silvey class* distance measure (e.g. the Information Distance I_D, or Kullback-Leibler-Divergence) with respect to its parameters:

$$I_D\left(X\left(\theta_1\right), X\left(\theta_2\right)\right) \geqslant I_D\left(Y\left(\theta_1\right), Y\left(\theta_2\right)\right)$$

In a (for the purpose of this thesis) more convenient form of this theorem, it expresses this inequality with Mutual Information $I\left(\cdot;\cdot\right)$ in a Bayesian Network. For variables X,Y,Z forming a Markovian Chain, i.e. when $p\left(x,y,z\right) = p\left(x\right)p\left(y|x\right)p\left(z|y\right)$:



The Data Processing Inequality states, that:

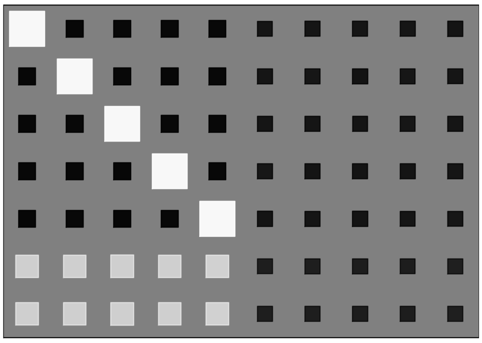
$$I\left(X;Y\right) \geqslant I(X;Z)$$

$$I\left(Z;Y\right) \geqslant I(X;Z)$$

In other words, the Mutual Information of variables Y and Z with the original variable X cannot increase along a chain of transformations.

A.2 Hinton Diagram

The Hinton diagram is a visual tool to qualitatively asses all elements of a matrix. Here is a simple example:



- the area of the squares equals the absolute value of the related matrix element, relative to the biggest element.
- white squares indicate positive, black squares indicate negative element values

Additionally,, when statistical averages over several matrices are computed in this thesis, the contrast to the background colour is used to code for certainty. The transparency $\alpha \in [0.0 \ldots 1.0]$ of a square relates to the standard deviation σ of element a_{ij}:

$$\alpha = e^{-\sigma(a_{ij})}$$

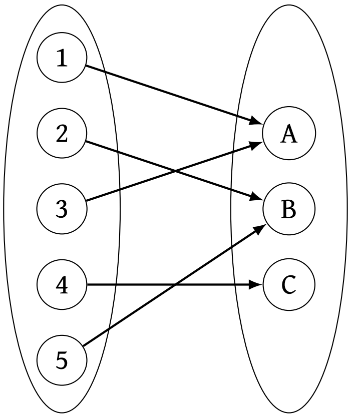
Therefore, opaque squares indicate certain, stable weights, whereas translucent squares indicate strongly fluctuating weight values.

A.3 Properties of Functions

For analytical evaluation, two basic properties of functions (associating elements from domain to co-domain) are of importance: Surjectivity and Injectivity. If a function between a domain and a co-domain is both surjective and injective, it is said to be bijective.

Surjective Function

For any given element of the co-domain, there is *at least one* relation to elements of the domain.

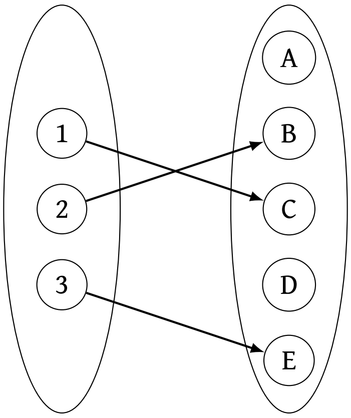


A surjective function implies, that all values of the target domain can occur.

In the context of the Cognitive Body model, the existence of a surjective map from environmental to agent states implies that every agent state has (at least one) specific environmental state associated with it, and thus is *semantically grounded*, or *associated* with this state. Likewise, *unassociated states* are therefore not (yet) bound to a certain set of environmental states.

Injective Function

For any given element of the co-domain, there is *at most one* relation to an element of the domain.



An injective function implies, that all elements of the co-domain uniquely distinguish elements of the domain, i.e. that an inequality relation in the co-domain also holds true in the target domain.

In the context of the Cognitive Body model, the existence of an injective map from environmental to agent states implies, that the maximum possible separation of environmental states has been attained, and no information about them is lost.

Appendix B

Addendum

B.1 Abstract

The thesis *Making Sense* derives principles for an *autonomous cognitive apparatus* (being a physical part of an agent), to enable it to transparently access features of the agent's physical body and environment. It regards action and observation to be aspects of fundamentally the same process that enables an agent to define itself. This process is shaped not only by the brain, but also the sensorimotor contingencies of the somewhat arbitrary, "attached" environment.

The presented *Cognitive Body* model formulates a framework for structuring the interactions of an agent into the manipulation and observation of hidden environmental states. It is agnostic to the type of learning algorithms used, but formulates the constraint of an *invariant causal chain* to create accessible approximations (and ideally copies) of the hidden, inaccessible, environmental state within the agent's cognitive apparatus (i.e. its brain). This is done by incorporating both acting and sensing as indispensable processes. It is closely related to and inspired by the concept of sensorimotor contingencies (Philipona *et al.* , 2003).

To formulate the approximation process, *Making Sense* utilizes two complementary but incompatible points of view. Borrowing from philosophical Constructivism, a *first person view* is assumed, complementary to the objective *third person view* of hard sciences. Learning is conceived as making the behaviour of constructed Umwelt and physical environment (including an agents physical body) similar.

In a series of exploratory simulations, the Cognitive Body model is then applied to make an agent learn to control a hidden environmental state. The state is not directly accessible via the agents signals, but indirectly by controlling a Finite State Machine. The series of simulations explore a simple yet nontrivial world, and highlight key differences and shared properties between the Cognitive Body model and the classical models of Reinforcement Learning and Classification, which are usually used when modelling perception and action independently.

B.2 Zusammenfassung in Deutsch

Die Arbeit *Making Sense* leitet Prinzipien für autonome kognitive Apparate (als Teil eines Agenten) her, mit Hilfe derer Merkmale des physischen Körpers des Agenten und dessen unmittelbarer Umgebung transparent zugegriffen werden können. Die Arbeit sieht Handeln und Beobachten als Aspekte eines grundsätzlichen Prozesses, der Agenten ermöglicht, sich selbst zu definieren. Dieser Prozess wird nicht nur durch das Gehirn, sondern auch durch die sensorimotorischen Möglichkeiten der im Grunde beliebigen Umgebung geformt.

Das präsentierte Cognitive Body Modell stellt einen Rahmen dar, um Interaktionen des Agenten mittels Manipulation und Beobachtung versteckter Umgebungszustände zu beschreiben. Es ist grundsätzlich blind gegenüber der Art der tatsächlich eingesetzten Lernalgorithmen, setzt aber die Bedingung der *invarianten Kausalkette*, um innerhalb des Agenten (in seinem Gehirn) zugreifbare Näherungen (idealerweise Kopien) versteckter Umgebungszustände zu erzeugen. Erreicht wird dies durch Einbeziehung sowohl des Handelns als auch des Beobachtens als dafür unerlässliche Vorgänge. Eng verwandt dazu ist das Konzept der *sensorimotor contingencies* (Philipona *et al.* , 2003), welches die Arbeit auch inspiriert hat.

Um das Verfahren zur Näherung zu beschreiben, verwendet die Arbeit zwei komplementäre, aber nicht vereinbare Sichtweisen. Die *Sichtweise aus erster Person* ist dem philosophischen Konstruktivismus entlehnt, als Ergänzung zu der in den Naturwissenschaften verwendeten objektiven *Sichtweise aus dritter Person.* In diesem Kontext wird Lernen als ähnlich machen des Verhaltens von konstruierter Umwelt und objektiver Umgebung (die den physischen Körper einschließt) aufgefasst.

In einer Serie von Simulationen wird das *Cognitive Body* Modell angewendet, damit ein Agent die Kontrolle über einen versteckten Umgebungszustand erlernen kann. Der Agent kann über seine gegebenen Signale nicht direkt auf den Zustand zugreifen, allerdings kann er es indirekt durch Kontrolle eines endlichen Automaten. Die Simulationsserie untersucht eine einfache, aber nicht triviale Welt, um wichtige Unterschiede und Gemeinsamkeiten des *Cognitive Body* Modells mit dem herkömmlichen Modellen , in welchen Wahrnehmung und Handeln getrennt modelliert wird (*Reinforcement Learning* und *Klassifizierung*), heraus zu arbeiten.

B.3 Curriculum Vitae

2009-2011

Middle European interdisciplinary Master in Cognitive Science, at University of Vienna, Austria. Mobility semester at Budapest University of Technology and Economics, Hungary.

2005-2009

Bachelor in Technical Computer Science (Bakkalaureat Technische Informatik) at Vienna University of Technology, Austria.

1994 -1999

Attendance of the 5 year higher-level secondary industrial and trade college (Höhere Technische Lehranstalt) HTL Wien Donaustadt, Industrial Engineering branch, in Vienna, Austria. Distinguished Graduation.

1986-1994

Attendance of Ground School (Grundschule) and Academic Secondary School (Allgemeinbildende höhere Schule) at the private Schule der Schulschwestern von unserer Lieben Frau, in Vienna, Austria.