A Neural Dynamic Account of Ideomotor Theory: Contingency Learning and Goal Oriented Behavior

MASTER THESIS

IN THE

DEGREE PROGRAM

"MASTER OF SCIENCE"
IN PHYSICS

AT THE FACULTY OF PHYSICS AND ASTRONOMY OF THE RUHR-UNIVERSITÄT BOCHUM

Author:
Stephan SEHRING
born in Essen

Supervisor:
Sophie AERDKER
First Examiner:
Prof. Dr. Gregor SCHÖNER
Second Examiner:
Dr. Jürgen DREHER

Bochum May 31, 2022

Ein Neuronales Dynamisches Modell Ideomotorischer Theorie: Kontingenzlernen und Zielorientiertes Verhalten

MASTERARBEIT

IM

STUDIENGANG

"MASTER OF SCIENCE"
IM FACH PHYSIK

AN DER FAKULTÄT FÜR PHYSIK UND ASTRONOMIE DER RUHR-UNIVERSITÄT BOCHUM

Autor:
Stephan SEHRING
geboren in Essen

Betreuung durch:
Sophie AERDKER
Erster Prüfer:
Prof. Dr. Gregor SCHÖNER
Zweiter Prüfer:
Dr. Jürgen DREHER

Bochum Mai 31, 2022

Abstract

Humans act in the world as autonomous beings who can set their own goals and act on them. As such, they strive to influence the world in which they live to bring about desired outcomes. Ideomotor Theory claims that actions required to achieve a goal are triggered by the mental representation of the desired outcomes. Assuming that goal-directed behaviour is driven primarily by the expected outcome, the question arises as to how an embodied nervous system translates representations of desired outcomes into actions that actually achieve them. Related questions are why certain outcome states are selected as goals while others are not, and how selected goals are stabilized against opportunistic sensory stimuli. Motor and perceptual experiments find that motor behaviour and attentional selection are characterized by an early preference for familiarity followed by a later preference for novelty. A similar dynamic could also explain the stabilization and habituation of goals.

This thesis provides a neural dynamic process model that is capable of ideomotor action by autonomously selecting its goals and of achieving them by reusing previously learned action-outcome contingencies. The model is implemented using Dynamic Field Theory, a mathematical modeling framework of neural dynamics at the level of neural populations. The build DFT-Architecture is endowed with the ability to act and perceive, by grounding it in a small simulated robot. The embodied architecture is capable of autonomous learning by forming intentional states of performed actions and observed outcomes. The dynamics of goals are investigated in a toy experiment loosely modeled after task switching paradigms. The experiment affords the small robot to interact with colored buttons and to learn the outcomes of its interactions.

The experiment demonstrates the models ability to use its goal representation to drive outcome oriented action. Additionally it shows how early familiarity preference and later novelty preference can account for initial stabilization of goal selections and their later habituation, which is consistent with similar patterns found in motor experiments.

This thesis aims to demonstrate that motor signatures such as perseveration and habituation can be accounted for in an embodied agent at the level of goals. More general, the modeled ideomotor processes may provide a grounding for a more open neural dynamic model of skill acquisition and reuse.

Zusammenfassung

Menschen agieren in der Welt als autonome Wesen, die sich ihre eigenen Ziele setzen und nach ihnen handeln können. Als solche streben sie danach, die Welt so zu beeinflussen, dass die gewünschten Ergebnisse erzielt werden. Die Ideomotorische Theorie besagt, dass Handlungen, die zur Erreichung eines Ziels erforderlich sind, durch die mentale Repräsentation des gewünschten Ergebnisses ausgelöst werden. Wenn man davon ausgeht, dass zielgerichtetes Verhalten in erster Linie durch das erwartete Ergebnis gesteuert wird, stellt sich die Frage, wie ein verkörpertes Nervensystem Repräsentationen gewünschter sensorischer Zustände in Handlungen umsetzt, die das gewünschte Ergebnis dann auch tatsächlich erreichen. Damit zusammenhängende Fragen sind, warum bestimmte Zustände als Ziele ausgewählt werden, andere hingegen nicht, und wie ausgewählte Ziele gegenüber opportunistischen Sinnesreizen stabilisiert werden. Motorik- und Wahrnehmungsexperimente zeigen, dass motorisches Verhalten und attentionale Selektion durch eine frühe Bevorzugung von Vertrautem gekennzeichnet sind, gefolgt von einer späteren Bevorzugung von Neuem. Eine ähnliche Dynamik könnte auch die Stabilisierung und Habituation an Ziele erklären.

In dieser Arbeit wird ein neuronales dynamisches Prozessmodell vorgestellt, das zu ideomotorischen Handlungen fähig ist, indem es autonom seine Ziele auswählt und durch Ausnutzung von zuvor erlernten Handlungs-Ergebnis-Assoziierungen erreicht. Das Modell wird mit Hilfe der Dynamischen Feldtheorie implementiert, einem mathematischen Modellierungsrahmen für neuronale Dynamik auf der Ebene neuronaler Populationen. Die aufgebaute DFT-Architektur wird mit der Fähigkeit ausgestattet, zu handeln und wahrzunehmen, indem sie in einem kleinen simulierten Roboter geerdet wird. Die verkörperte Architektur ist in der Lage, autonom zu lernen, indem sie intentionale Zustände von ausgeführten Handlungen und beobachteten Ergebnissen bildet. Die Dynamik der Ziele wird in einem Spielzeugexperiment untersucht, das lose an ein Aufgabenwechselparadigma angelehnt ist. Das Experiment ermöglicht es dem kleinen Roboter, mit farbigen Knöpfen zu interagieren und die Ergebnisse seiner Interaktionen zu lernen.

Das Experiment demonstriert die Fähigkeit des Modells, seine Zielrepräsentation zu nutzen, um ergebnisorientiertes Handeln zu steuern. Darüber hinaus zeigt es, wie die frühe Vertrautheitspräferenz und die spätere Neuheitspräferenz für die

anfängliche Stabilisierung selektierter Ziele und ihre spätere Habituation verantwortlich sein können, was mit ähnlichen Mustern in motorischen Experimenten übereinstimmt.

Die Arbeit zeigt, dass motorische Merkmale wie Perseveration und Gewöhnung in einem verkörperten Agenten auf der Ebene der Ziele erklärt werden können. Allgemeiner könnten die modellierten ideomotorischen Prozesse eine Grundlage für ein neuronales dynamisches Modell hierarchischer Fähigkeitenakquisition sein.

Contents

List of Figures

Chapter 1

Introduction

Humans are continuously confronted with a highly complex and ever-changing world. In relation to this world, human beings do not behave merely as passive observers of a world made up of material objects. Rather, they act in it as autonomous agents pursuing their own goals. The world therefore reveals itself to human beings not only as a place of material objects, but also as a place of action, in which both knowledge about material properties and their meaning in relation to one's own goals and actions play a role (Peterson, 2002). To achieve a goal, one must be able to recognize relevant properties of the environment on the one hand, but on the other hand one also needs knowledge of how to interact with the environment and what results can be expected from them. Learning such action-outcome relations and using them to achieve goals is a fundamental problem that humans must solve in the face of novel situations.

As an illustrative example. let's imagine an individual that is currently in the process of learning to play a new chord on his guitar. Lets also imagine, that he is a novice guitar player, that has just learned some basics. Producing clean tones is quite difficult, as it requires precise finger control and, in a chord, close coordination with strumming patterns.

To learn a new chord, this imaginary person must first know how it is played. For this he may try out different fret combinations in an attempt to find the one correct combination that leads to the desired chord sound. Once he has found the right combination, he can later reuse the gained knowledge to produce it again. If he for example wants to play a song, the student can use previously acquired knowledge to translate chord notations on a note sheet into the correct finger positions on the guitar frets. In addition, he can independently recognize whether the tones produced correspond to the desired chords.

In this sense, he has linked the concept of the learned chord (e.g., symbolized by the name) with individual hand movements and the resultant tone. If you stay in this picture, you will find that you can refer to playing the chord on different levels of abstraction. Either as a single action, as individual finger movements or even as individual muscle movements. Either way, once the chord has been learned, referring to it at a higher-level of abstraction seems to be sufficient for us to produce it subsequently.

The question now arises as to how this is possible. How is it that we can effectively act in the world to achieve some abstract goal, without any notion of all the underlying mechanisms of our anatomy and neurophysiology, or the complex physical laws governing the material world? That is, how can our nervous system transform abstract conceptual goals about some high-level environmental states, into low-level body movements that actually achieve the desired outcome?

This question has been raised in one form or another for quite some time now. Stock and Stock (2004) have traced one branch of research dealing with this question, called Ideomotor Theory, back to the mid-19th century. In a nutshell, Ideomotor Theory claims that the mental representation of a goal (the anticipation of an outcome), is sufficient to initiate an action that brings about the desired outcome. The key process that enables this ideomotor action, is through the association between goal and action representations. When moving in the environment, we continuously perceive the outcomes of our actions. By learning these action-outcome relations we can associate the goal representation with an action, that has previously brought about the desired outcome. Research in this field is guided by this rather simple consideration and is aiming to provide theoretical accounts and evidence of the cognitive processes involved to achieve this functionality. That expected outcome plays an important role in the initiation of action has been shown by experiments demonstrating signatures such as stimulus-response compatibility and Simon effects (Kornblum, Hasbroucq, and Osman, 1990; Kunde, Hoffmann, and Zellmann, 2002; Erlhagen and Schöner, 2002). Expected outcomes are used in many computational systems regarding outcome anticipatory autonomous learning (Butz, Sigaud, and Gérard, 2003).

This thesis aims to provide a neurally plausible process model is capable of ideomotor action by autonomously forming associations between neural representations of goal and action and can subsequently use these associations to achieve self selected goals. The model is endowed with the ability to act and perceive, by giving it control of an embodied robot in a simulated environment. To provide a neurally plausible account, the proposed model is implemented as a neural dynamic network on the level of neural populations. This is based on Dynamic Field Theory

(DFT) (Schöner, Spencer, and DFT Research Group, 2015), which provides a mathematical modeling framework describing the dynamics of neural populations (see chapter 2 for a detailed description). DFT represents cognitive variables governing behavior and though as attractor states of the dynamics. Cognitive processes evolve according to their their recurrent connectivity in neural dynamic networks (Schöner, 2020).

Modeling the processes of ideomotor action amounts to specifying the nature of the mental representations relating to goals and action, laying out their conceptual level of abstraction in relation to lower-level perceptual/motor information and proposing what kind of mental processes are involved for goals to drive behavior (Jarecki, Tan, and Jenny, 2020). The next paragraphs lay out the core elements that guide the development of the proposed model and its implementation.

Schöner (2020) has argued that the neural representations which enable cognition are embedded in low-dimensional feature dimensions. The resulting feature representation specify the relevant features of perceived events and to be performed movements. Feature representations are either extracted from higher dimensional sensory information in the case of perceptions or mapped onto motor commands in the case of movement plans. Importantly, perceptions and actions are represented in a shared representational framework (Hommel et al., 2001).

To form an action plan for a given goal, one needs knowledge about which actions lead to the desired outcome. In its simplest form, this corresponds to knowledge about action-outcome contingencies, that map the causal relation of perceived events and the action that preceded them. Contingency representations are propositional, since its meaning is determined by its inner structure of the feature representations that make up the causal relation of the contingency. Learned action-outcome contingencies can be conceptualized as beliefs (Tekulve and Schöner, 2020).

Forming beliefs amounts to the binding of the feature representation relating to the performed action and the observed outcome. Autonomous agents like humans, can learn contingencies autonomously while acting in the environment. For this, the agent must observe the environment and detect whether a change was caused by a performed action. In the simplest case an action causes an immediate perceivable outcome. A model capable of belief learning must detect and represent these action-outcome events that indicate a belief formation episode. DFT provides a neurally plausible mechanism for the binding of feature representations, by deploying Hebbian learning in a reinforcement learning paradigm. Tekulve and Schöner (2020) has shown how beliefs about contingencies can be formed by Hebbian learning modulated by a reward signal.

Using beliefs to drive action requires a process of recalling a known contingency representation. According to Hommel et al. (2001) bound feature representations pertaining to action-outcome contingencies can be recalled by partial feature cues, that may originate either from perceived features in the environment or other cognitive feature representations, for example of goals. This is similar to a pattern completion mechanisms, which has been identified as a key mechanism involved in the opportunistic emergence of desires as a result of associated perceptions (Papies and Barsalou, 2014).

Taking the above into account, this thesis posits that goals can be linked to appropriate actions via the belief of a contingency matching the desired outcome. The feature representation of a goal can cue the bound feature representation of a belief, which then recalls the complete belief representation, including the action that is needed to produce the desired outcome. The opposite pendant to top-down cuing of beliefs, is bottom-up cuing from perceived features of the environment. As stated above, opportunistic activation of beliefs may be a motivational factor, especially if the outcome of can be considered rewarding. In DFT-terms, opportunistic activation of a belief can lead to the activation of a goal representation which can then drive action. This thesis deploys the belief architecture of Tekulve and Schöner (2020) to model this link between goal, action and perception.

Opportunistic activation offers a mechanism to investigate motivated behavior, i.e. how goals are selected and maintained. Pursuing one goal precludes the simultaneous pursuit of most other goals, since achieving a desired outcome requires purposeful actions that preclude the achievement of any other outcome. Once a goal has been selected, the question arises as to how it is stabilized against opportunistic input, and at which point old goals are discarded in favor of new ones. A widely used paradigm to investigate this is the voluntary task switch paradigm (Arrington and Logan, 2004).

At the level of action, this kind of stabilization has been associated with signatures such as motor perseveration found in infants in A not B tasks (Thelen and Schöner, 2001; Dineva and Schöner, 2018), or signatures of infant's orientation response, which show early familiarity preference followed by later novelty preference (Perone and Spencer, 2013; Schöner and Thelen, 2006). Aerdker, Feng, and Schöner (2022) have provided a unified account of motor habituation and dishabituation, that shows an analogical mapping between these signatures of the motor and perceptual domain. The DFT account build by Aerdker, Feng, and Schöner (2022) demonstrates that early familiarity preference followed by later novelty preference emerges generically in action selection neural fields because of memory traces in both excitatory and inhibitory neural layers. This thesis lifts these signatures on the level of actions to the level of goals. Selection decisions of goals are initially stabilized by excitatory memory traces and eventually destabilized due to inhibitory memory traces. This can be seen as a early familiarity preference as outcomes are being achieved, followed by a later novelty preference as the architecture habituates to the outcome.

Embodying a neural architecture means grounding it in sensor and motor surfaces with which it can perceive and act in the environment. To model the ideomotor processes described above, the architecture should be able to perceive and act in its environment and should be able to learn from the outcome of its actions. For this, the build DFT-Architecture is structured according to Tekülve's (2021) neural process model of intentionality.

Intentionality is a philosophical notion, that refers to the capability of the mind to form mental states that are directed at, or are about states of affairs in the world (Searle, 1983). Searle (1983) categorizes mental states according to their psychological modes. He differentiates 6 psychological modes based on their direction of fit (DoF). Mental states of the mind-to-world DoF, represent some property or event of the environment (how the world is). These include perception, memory and belief. In contrast, mental states that represent to be produced properties or events in the environment are of the world-to-mind DoF (how the world should be). Psychological modes of the world-to-mid DoF include desire, prior-intention and intention in action. Tekülve (2021) has demonstrated that an embodied DFT model implementing all 6 psychological modes can autonomously learn and act in its environment.

Previous models of Ideomotor Theory often sidestep these detailed considerations for embodied agents and focus more on computational implications for autonomous learning (Herbort and Butz, 2012; Hommel et al., 2001). In contrast, this thesis investigates how the nervous system of an embodied agent can endow the agent with the ability to learn autonomously, set its own goals and use its knowledge to act in an outcome-oriented fashion.

In summary, this thesis uses DFT to build a neural dynamic account of Ideomotor Theory. The model can select its own goals and demonstrates outcome-oriented and opportunistic action. This capability is demonstrated by implementing it in an embodied simulated robot and endowing it with the ability to act and learn in the environment. For this the DFT-Architecture is structured along the lines of Tekülve's (2021) process model of intentionality. To account for stabilization and inhibition in goal selection, the model includes memory traces at the goal level. The behavior of the model is tested in a toy experiment loosely modeled after task switching paradigms (Arrington and Logan, 2004). It affords the robot to navigate a simulated arena that is filled with colored buttons. The robot can perform abstract actions on these buttons. Some action-color combinations may result in an outcome sound. The experiment demonstrates how early familiarity preference followed by later novelty preference, can emerge at the level of goals.

Chapter 2

Dynamic Field Theory

The model created in this work is based on Dynamic Field Theory (DFT; Schöner, Spencer, and DFT Research Group, 2015). DFT provides a mathematical modelling framework in which cognitive behavioural variables are linked to activation patterns of neural populations. This chapter first briefly explains the theoretical background and principles that underpin DFT. For this, Sections 2.1, 2.2 and 2.3 draw a conceptual line from its theoretical roots as a connectionist theory of artificial neurons, through the dynamical description of cognitive representations formed by neural populations, to the dynamics of neural fields, which form neural representations via activation distributions of neural populations along embedded feature dimensions.

Sections 2.4 and 2.5 then give an overview of the individual building blocks used in this thesis to implement different cognitive functions. Section 2.4 gives an overview of how neural field architectures can implement basic cognitive functions such as transient detection or working memory. The description is limited to the mechanisms used in this thesis. Section 2.5 describes how neural plasticity and learning are accounted for in DFT. Finally, Section 2.6 gives an overview of how the process model of intentionality developed by Tekülve (2021) guides the integration of the various cognitive functions into a grounded model of ideomotor theory in this thesis.

The descriptions of these foundations are heavily based on Tekülve (2021), Schöner (2020) and Schöner, Spencer, and DFT Research Group (2015).

2.1 Neural Modeling Framework of Cognition

At it's core, DFT is a mathematical modeling framework that aims to provide neurally grounded models of cognitive processes involving behavior and thought. As a neurally grounded theory it tries to capture how the dynamics of behavior and cognition emerge from the dynamics of the underlying neural substrate. In that sense, DFT is a connectionist approach, as it assumes that thought, perception and other cognitive processes are ultimately encoded in changing activation patterns at the level of interconnected neurons. This a very common axiomatic perspective from neuroscience. Its mathematical formulation has let to the development wide range of different biologically motivated models of the central nervous system, and are notably the foundation of modern artificial intelligence systems in which large artificial neural networks are used solve a wide range of tasks including everything from autonomous driving to playing computer games (Vinyals et al., 2019).

Mathematical models generally involve a simplified artificial neuron as the smallest functional unit of a neural network. The description of an artificial neuron is predicated on the assumption that relevant information is not located at the level of individual activation peaks of the membrane potential but at the level average firing rates. In that way a neuron in a neural network can be conceptualized in an input-output fashion, receiving input from other neurons in the networks according to their activation and synaptic strength, and projecting output to other neurons if a certain input threshold has been reached. Formulated mathematically, an artificial neuron is a simple input-output function:

$$u_j(t+\tau) = g\left(\sum_{i=1}^{N} \omega_{ij} u_{ij}(t)\right). \tag{2.1}$$

Here $u(t)$ is the activation variable and models the average firing rate. The synaptic connection strength between two neurons is modeled with ω. The activation function g is typically a sigmoid or step function. The activation of a neuron at time step $t+\tau$ is positive, if the activation input of time step t reaches the threshold of g.

Neurons in the central nervous system operate on a time scale of milliseconds. Cognitive processes, on the other hand, develop on a scale of hundreds of milliseconds. Moreover, behaviour that emerges from cognition has been shown to correlate most closely with the activity of small populations of neurons, rather than individual neurons (Schöner, 2020). DFT therefore takes the stance that behavioral variables governing such cognitive processes are best described at the level macroscopic activation patterns of neural populations.

2.2 Neural Dynamics

As activation in neural populations is not perfectly synchronized, DFT takes the step of formulating the evolution of these activation patterns as non-linear dynamical systems in continuous time. Although not necessarily linked to concrete neural population in the brain, DFT assumes that behavioral variables nonetheless evolve in continuous time according to the activation dynamics of neural populations. In that way DFT grounds models of cognition in neural dynamic states that are linked to behavioral variables of a model (Schöner, Spencer, and DFT Research Group, 2015).

2.2.1 Dynamic Neural Nodes

The average activation rate of a population of neurons is represented by a neural node u, and the evolution of its activation is modeled as a dynamical system in continuous time. In the context of a neural architecture, a neural node can represent for example a detection state or a concept representation. Analogous to equation 2.1, a neural architecture on the level of neural populations can be modeled as a network of neural nodes. The activation of one node influences the activation of other nodes in the architecture and vice versa. Formally the coupled dynamics of N such neural nodes can be expressed as an N-dimensional coupled differential equation:

$$
\begin{aligned}
\tau \dot{u}_1 &= -u_1 + h + s_1(t) + \sum_{j=1}^{N} k_{1j} g(u_j) + \xi_1(t) \\
\tau \dot{u}_2 &= -u_2 + h + s_2(t) + \sum_{j=1}^{N} k_{2j} g(u_j) + \xi_2(t) \\
&\dots \\
\tau \dot{u}_N &= -u_N + h + s_N(t) + \sum_{j=1}^{N} k_{Nj} g(u_j) + \xi_N(t).
\end{aligned}
\tag{2.2}
$$

Here h is the resting level, to which the activation relaxes to if there is no input. $s_i(t)$ models the external input to the *i'th* node for example from sensors. $g(u_j)$ is the sigmoidal activation function $g(x) = \dfrac{1}{1+e^{-\beta x}}$, and determines the threshold at which activation of the neural node u_j is passed on to other nodes. If the activation variable u_j becomes close to or larger than 0, activation $g(u_j)$ becomes positive, at which point the node u_i projects activation to other nodes in the network. k_{ij} models the connection strength between nodes. If $k_{ij} > 0$ node u_i receives excitatory input from u_j, while $k_{ij} < 0$ corresponds to an inhibitory connection. $\xi(t)$ simulates noisy input by generating small perturbations, for example in a Gaussian distribution. τ regulates the time scale of the dynamics.

To understand the dynamics of the architecture it is useful to look at the dynamics of a single neural node. Figure 2.1 depicts the dynamics of a single node in phase space for different input strengths. For each input level, the dynamics has different fixed points, which are marked in the figure by the grey dots. The non-linearity around $u = 0$ comes from the self excitation $g(u)$.

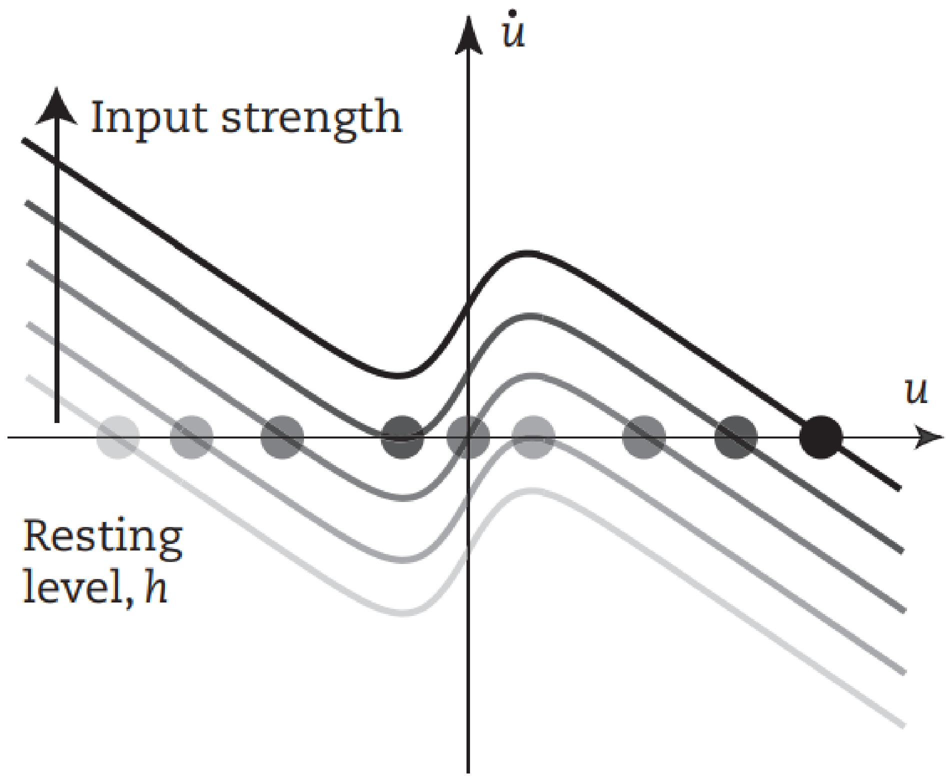

FIGURE 2.1: Phase space representation of single neuron dynamics for varying levels of input strength. Fixed points are marked by dots. Figure 1.15 from (Schöner, Spencer, and DFT Research Group, 2015).

The input strength depicted in 2.1 can be seen as the sum of external $s(t)$ and internal input from other nodes of the architecture. For low input levels there is only one stable fixed point at $u = s + h$ (light gray in Figure 2.1). If input increases two more fixed points appear in a pitchfork bifurcation. The dynamics than have one unstable fixed point close to $u = 0$ and two attractors, one on each side of the repellor (gray in Figure 2.1). In this case the dynamics becomes bistable. Depending on the initial condition the activation can relax to one of the two attractors. The attractor in the $u > 0$ regime is stabilized by the self excitation of the neuron. For even higher input, only the $u > 0$ attractor is stable (black in Figure 2.1).

In higher dimensions the stability characteristics of a fixed point is determined by the Eigenvalues of the Jacobi-Matrix of equation 2.2. Important here is that the dissipative $-u$ terms make the dynamics stable as the state of the architecture $\vec{u}(t)$ asymptotically converges to the closest current attractor. This sustains representations of mental states over a time span of seconds or even minutes against fast transient input.

The attractor of the architecture is highly dependent on the recurrent connections characterized by k_{ij}. Strong mutual inhibition for example can lead to a selective dynamics, where only 1 neural node can be active at a time. External input can cause bifurcations, which lead to temporally discrete instabilities resulting in new stable states. The most important instabilities, which enable neural architectures to perform cognitive functions are discussed in Section 2.3.2. In the framework up to now it is not clear yet what kind of input is presented and how individual neurons come to be linked to particular features.

2.3 Dynamic Neural Fields

DFT aims to capture cognitive processes of an embodied nervous system. This means that DFT-Architectures are considered to be connected to the sensorimotor surface at all times. Neural populations in an architecture represent information corresponding to the sensor or motor modality they are connected to. This information can be subject to fast changes and is quite noisy, while our cognitive representations relating to perceptions and behavior persist over longer time periods. For instance, our visual experience remains invariant against fast shifts of visual input resulting from the saccades we perform approximately 3 times a second (Schöner, 2020).

In addition to being quite noisy, perceptual sensory input is high-dimensional. For example the human retina features more than 100 million photo receptors. Of course the actual useful information contained in such an image is highly constrained by the properties of the environment. Continuity in surfaces, colors, positions and other features of perceived objects highly reduce the dimensions needed to characterize an image (Schöner, Spencer, and DFT Research Group, 2015).

Neural substrate close to the sensorimotor surface can be seen as either extracting low-dimensional feature representations from high-dimensional noisy stimuli or as mapping low-dimensional motor representations onto higher dimensional actuator movements. For example, neurons in the primary visual cortex are tuned to different features of visual stimuli including orientation, topographic location on the retina, or motion direction. Together, neurons with different tunings form overlaying maps of different feature combinations, that represent the features of a visual stimuli (Swindale, 2000).

A central hypothesis DFT commits to, is that representations important for our cognition are based on low-dimensional neural feature representations that are stabilized by recurrent connections and map information pertaining to perception and movement generation along continuous feature dimensions.

2.3.1 Toward a Continuous Description

DFT models the feature tuning of neural populations by embedding the dynamics of discrete sets of neural nodes into low-dimensional feature dimensions. The mathematical formulation takes the form of dynamic field equations:

$$\tau \dot{u}(x,t) = -u(x,t) + h + s(x,t) + k(x') \circledast g(u(x',t)). \tag{2.3}$$

These dynamics have an analogous form to that of discrete activation variables (see equation 2.2). The discrete values for u, s and k are replaced by continuous functions along the feature dimension x. The dimension x can stand for different features, such as color, position, or sound pitch. Attractor states are now continuous functions as well. For each position x the value of $u(x,t)$ relaxes to the closest value for which $\dot{u}(x) = 0$. The resting level h is set to be the same at every location. The sum in equation 2.2 modeling the recurrent connection of the network is replaced by the convolution of the activation function $g(u(x))$ with an interaction kernel $k(x)$:

$$\begin{aligned} \left[k(x') \circledast g(u(x',t))\right](x,t) &= \int k(x-x')g(u(x',t))dx' \\ &\simeq \sum_{j=1}^{N} k(x_i - x_j)g(u(x_j)). \end{aligned} \tag{2.4}$$

The connection between neural fields and discrete nodes becomes clear when approximating the convolution integral by sampling x at N equidistant locations. The approximation then recovers the recurrent connection sum of equation 2.2.

Neural fields can also have more than one dimension. For example, a field might encode both the horizontal as well as the vertical position of an object on the retina. In that case the field is defined over both dimensions and the convolution of the field equation integrates over both dimensions:

$$\tau\dot{u}(x,y,t) = -u(x,y,t) + h + s(x,y,t) + k(x,y) \circledast g(u(x,y,t)). \tag{2.5}$$

When the activation u at position x passes the threshold ($g(u(x) > 0$) it starts to excite activation in its surrounding where the kernel is positive and it inhibits activation in its surrounding where the kernel is negative. (When u is below threshold no activation is passed on to neighboring neurons). The shape of the kernel determines the kind of coupling (inhibitory or excitatory). The interaction is characterized by the parameters pertaining to local excitation ω_{le}, local inhibition ω_{li} and global inhibition ω_{gi}

$$k(x) = \omega_{le}e^{-x/2\sigma_{le}} - \omega_{li}e^{-x/2\sigma_{li}} - \omega_{gi}. \tag{2.6}$$

σ_{le} and σ_{li} are the standard deviation of the local interactions. Figure 2.2 depict the three prototypical kernels. Local excitation stabilizes peaks of activation, when external input pushes activation through threshold. Local excitation in combination with strong global inhibition is a selective kernel that stabilizes a single peak of activation against competing external input. A kernel with local excitation and mid range inhibition allows for self-stabilizing peaks of activation.

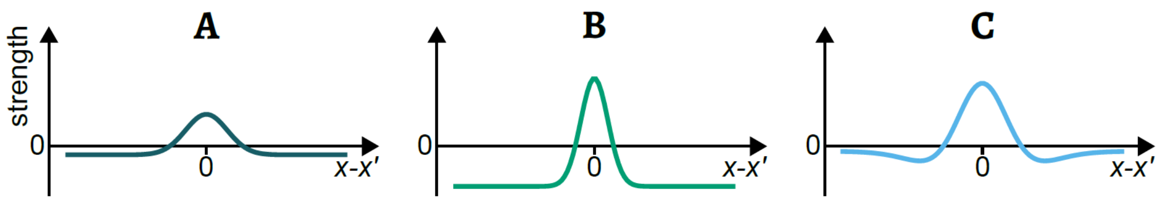

FIGURE 2.2: Common interaction kernels with A) local excitation, B) local excitation and global inhibition and C) local excitation and mid range inhibition. Figure 2.2.2 from (Tekülve, 2021).

Figure 2.3 illustrates how a kernel with local excitation and global inhibition selects and stabilizes local peaks of activation. The field receives 2 peaks of external

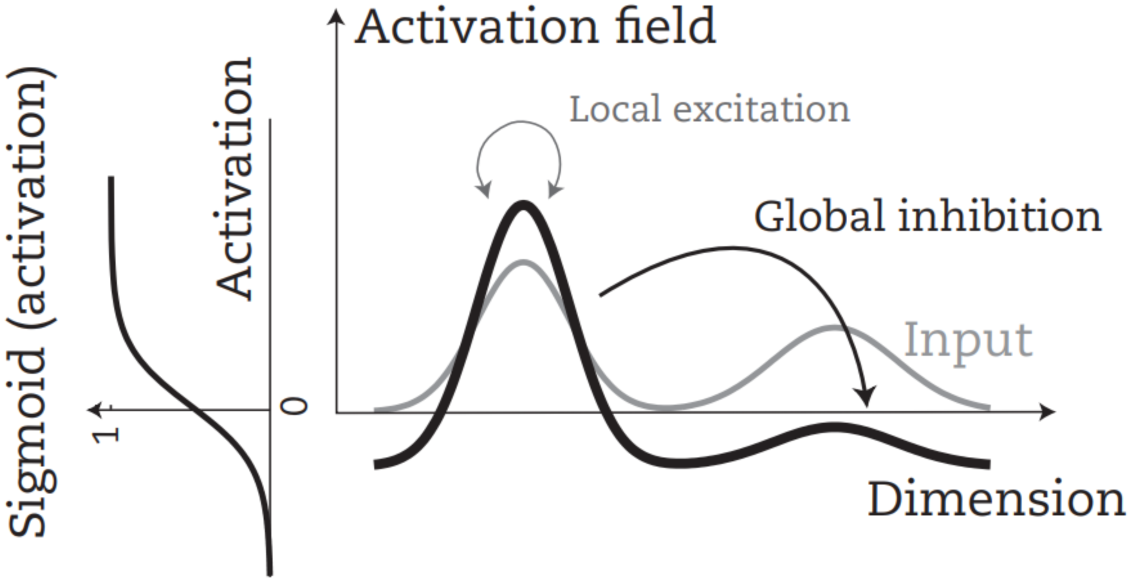

FIGURE 2.3: Left: Activation function $g(u)$. Lateral interactions take place in areas with positive activation functions. Right: Schematic representation how local peaks of activation form as a result of lateral interactions. Figure 2.5 from (Schöner, Spencer, and DFT Research Group, 2015).

input at different feature locations. When external input pushes activation at one location above threshold, the activated regions start to excite nearby locations, while suppressing activation globally. As a result a single localized peak of activation emerges.

These peaks of activation form the fundamental representational units of DFT. A peak encodes the feature value of the position of the peak along the feature dimension. Ultimately the specific meaning of a field is determined by its internal dynamics and its connection to the remaining architecture. The feature dimension is determined by the ultimate connection of the field to the sensorimotor surface.

2.3.2 Instabilities in Dynamic Neural Fields

Dynamic Neural Fields (DNFs) are stable. They asymptotically converge to the state at which $\dot{u}(x) = 0$. When external input is below threshold, this corresponds to $u(x) = h + \sum_i s_i(x)$. The lateral interactions defined by the interaction kernel cause bifurcations in the dynamics, when activation at one location passes the threshold of

$g(u)$. These instabilities are the primary mechanism that enable neural dynamic architectures to perform cognitive functions, as they account for changes in otherwise stable cognitive representations.

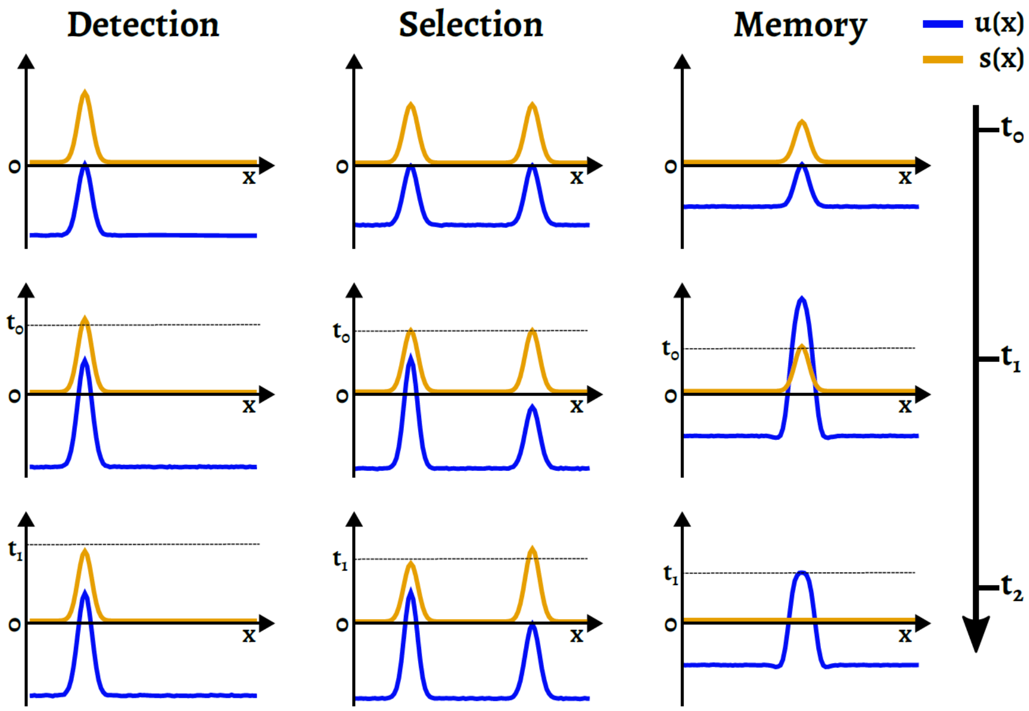

FIGURE 2.4: Activation snapshots demonstrating the dynamics of detection instability (left), selection decision (middle) and working memory (right). Figure 2.2.3 from (Tekülve, 2021).

Figure 2.4 demonstrates the dynamics of the 3 main instabilities used in DNFs to account for different cognitive events. The detection instability (left) signifies the detection of external input that is just strong enough to pass the threshold (t_0, t_1). It occurs as the local excitation of the interaction kernel adds to the attractor (t_1). This stabilizes the detection peak against noisy input. When external input fluctuates and decreases after a selection instability has occurred, the activation peak signifying the detection remains (t_2). Only when the input is reduced by a sufficiantly large amount, does the detection peak fall back below threshold. This is called a reverse detection instability. A DNF like this might for example be connected to the retina image and represent the detection of a certain color.

The selection instability (middle) occurs when a DNF is presented with multiple peaks of external input (t_0) and if the interaction kernel features strong global inhibition and local excitation. The global inhibition leads to a competitive dynamics between the input peaks. If one passes activation threshold in a detection instability, the global inhibition inhibits the other input peaks, such that they remain below threshold (t_1). The combination of local excitation and global inhibition stabilizes the selection decisions, as the selection peak remains stable, even if competing input increases (t_2).

The memory instability (right) occurs in DNFs that have an interaction kernel with strong local excitation. A peak that passes the detection instability (t_0, t_1) receives enough recurrent input, that it remains, even as external input is removed (t_2). The memory instability can be used as a working memory representations, that sustain perceptual information, even as the stimulus that caused the detection has since vanished.

Discrete neural nodes (see Section 2.2.1) have analogous instabilities. Recurrent connections are modeled by the elements $k_i j$ of the interaction sum. Similar to the continuous case the instabilities self-excitation leads to detection instabilities, and self-sustained activation, while global inhibition leads to a selective dynamics.

2.3.3 Connections between Dynamic Neural Fields

Models of cognitive processes can rarely be achieved by a single neural field. To account for more complex cognition, neural fields and nodes are coupled together into a dynamic neural field architecture. Connecting two fields amounts to passing supra-threshold activation from the source field to the target field. The passed activation is weighted with a connection kernel before being added to the target fields rate of change. The target field then evolves according the the sum of external activation passed to it from other fields:

$$\tau \dot{u_{tar}}(x,t) = -u_{tar}(x,t) + h_{tar} + k(x) \circledast g(u_{tar}(x,t)) + \sum_i c_{src_i \rightarrow tar}(x) \circledast g(u_{src_i}(x,t)). \tag{2.7}$$

The connection kernel $c_{src \rightarrow tar}$ has the form of a Gaussian. Commonly the standard deviation of the kernel is chosen to be sufficiently small, such that the convolution integral simplifies to a simple scalar multiplication.

It is also possible to connect fields that have different dimensions. To do this, the source field has to be either expanded or contracted depending on the dimensionality of the target field. An input to a target field can be expanded by keeping input constant across the missing dimensions. A peak in a one dimensional field then has the form of a ridge input when expanded to two dimensions. In that case, a ridge input from the source field at position x is constant for all y:

$$\tau \dot{u}_{tar}(x,y) = -u_{tar}(x,y) + h + k(x,y) \circledast g(u_{tar}(x,y)) + c_{src\rightarrow tar}(x) \circledast g(u_{src}(x)). \tag{2.8}$$

To connect a higher dimensional source field to a lower dimensional target field requires the source field to be contracted, such that it fits the dimensionality of the target field. This is done by integrating over the feature dimensions that are contracted:

$$\tau \dot{u}_{tar}(x) = -u_{tar}(x) + h + k(x) \circledast g(u_{tar}(x)) + c_{src\rightarrow tar}(x) \circledast \int g(u_{src}(x,y))dy. \tag{2.9}$$

In general, the dimensionality of the target and the input have to match. For larger field architectures implementing different feature representations it is important that fields pertaining to different features are not directly connected. For example a one dimensional field defined over the color dimension should not pass activation to a one dimensional field pertaining to the pitch of perceived sounds. This is important because fields ultimately receive their meaning through their connection to the sensorimotor surface. Crossing activation streams leads to ambiguity regarding the meaning of a peak in a neural field (It is for example not possible to hear the color of an object).

Activation streams may however meet when passed e.g. as a ridge input into a higher dimensional field that represents different feature values. Then the lower dimensional fields only pass input along the feature dimensions they represent, while being constant across different dimensions. For example, a field defined over both the sound-pitch and the color dimension can receive ridge input from both sound and color fields. If the two dimensional field is tuned such that it only activates at coordinates where both ridge inputs overlap, it can be seen as a binding of the corresponding sound and color representations. In this way, activation peaks pertaining to different features can be combined without leading to ambiguity regarding their representational meaning.

2.4 Cognition from Neural Field Architectures

This section covers a range of cognitive field architectures that perform different cognitive functions and are used as building blocks for the architecture build in this thesis. The cognitive functions covered are peak detector, boost and concept nodes (Section 2.4.1), transient detection (Section 2.4.2), short-term memory (Section 2.4.3), match fields (Section 2.4.4) and the sequencing of behavior (Section 2.4.5).

2.4.1 Peak Detector, Boost and Concept Nodes

Individual neural nodes can receive input from contracted neural fields and project activation to neural fields via expansion. Depending on the tuning of the neural node and the type of the connection, the node can perform a set of logical tasks.

A peak detector node receives input from a neural field. It is tuned such that it activates, as soon as a supra-threshold peak of activation forms in the connected field. The node represents the activation status of the connected field. If active, the peak detector node signifies that at least one peak is present in the connected field. A peak detector node can also be used in combination with multiple neural fields. Depending on the resting level of the node, it can then be seen as a categorical representation counting the number of active fields connected to the node.

A neural node projecting its activation to a higher-dimensional neural field is called a boost node. The expanded input to the neural field, homogeneously raises its resting level. Boost nodes are used to modify the dynamic regimes of the fields they project to. For example a field may be tuned such that external input peaks can only induce a detection instability, if it is combined with a homogeneous boost from a boost node (see Figure 2.5 top row). Depending on the interaction kernel, the boost can be used to put the field into a self-stabilized, selective or even self-sustained regime (see Section 2.3.2). Boost nodes can also simultaneously act as peak detector nodes. In this way these neural nodes act as representations of categorical states of the architecture and can induce cognitive processes that are conditional on this state.

Concept Nodes represent learned categorical concepts. Unlike boost and peak detector nodes, the concept nodes do not have a homogeneous connection to higher-dimensional fields. Instead, they are connected with Hebbian connections (see Section 2.5.2), which map the connection of a concept to continuous feature values via a weight pattern. For concepts that have already been learned, the weight pattern

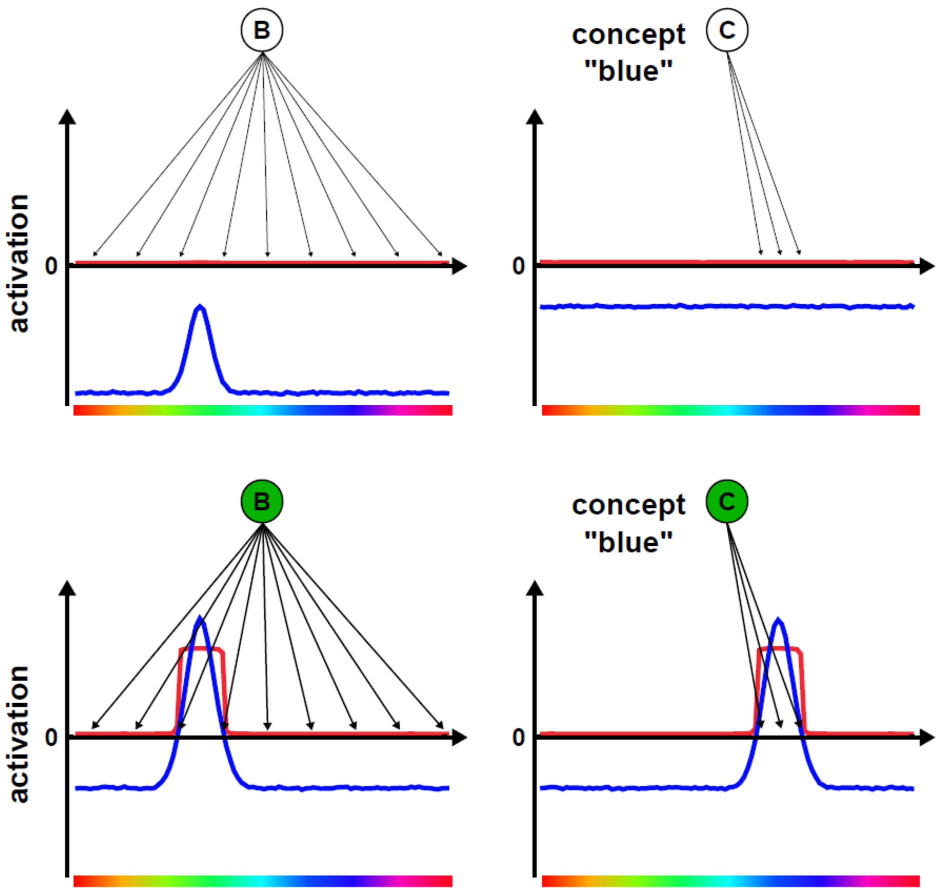

FIGURE 2.5: Schematic depicting how boost (top) and concept nodes (bottom) can influence the dynamics of a connected neural field. The boost node passes a homogeneous boost to the connected field and thus induces a detection instability. The color concept node in the shown example represents the color blue. It passes a local boost to the neural field centered at the blue hue values of the connected field. Figure 2.3.1 from (Tekülve, 2021)

is fixed. It is these weight patterns that give each concept node its meaning. Figure 2.5 shows an example of how a concept node representing the colour blue induces a peak in a colour field corresponding to the blue color hues.

Concept nodes can also act as peak detectors, for example, detecting a color concept in a perception. They then abstract continuous perceptual values into mental representations that are independent of exact feature values. A concept node representing the colour blue will detect all shades of blue regardless of the exact hue.

2.4.2 Transient Detection

Transient detectors detect newly formed peaks or transient events (movements of peaks) in a connected field. A transient detector consists of two neural fields that are identical except for their timescale and the input they receive. The field operating on a faster timescale u_e represent the detected transient, receives excitatory input from the source field u_{src} and inhibitory input from the field operating on a slower time scale u_i. The slower field only receives external input from the source field u_{src}.

The faster field represents the detection that a new peak emerged or shifted position in the connected field. This representation is destabilized by the slower inhibitory field after some time. The timescale of the slower field determines how long a transient detection is maintained. The equations 2.10 and 2.11 are analogous with the exception of the inhibitory coupling of u_e with u_i and $\tau_e < \tau_i$.

$$\begin{aligned}\tau_e \dot{u}_e(x,t) = &- u_e(x,t) + h + k(x) \circledast g(u_e(x)) + c_{src}(x) \circledast g(u_{src}(x)) \\ &- c_i(x) \circledast g(u_i(x))\end{aligned} \tag{2.10}$$

$$\tau_i \dot{u}_i(x,t) = -u_i(x,t) + h + k(x) \circledast g(u_i(x)) + c_{src}(x) \circledast g(u_{src}(x)) \tag{2.11}$$

The equations 2.10 and 2.11 are analogous with the exception of the inhibitory coupling of u_e with u_i and $\tau_e < \tau_i$.

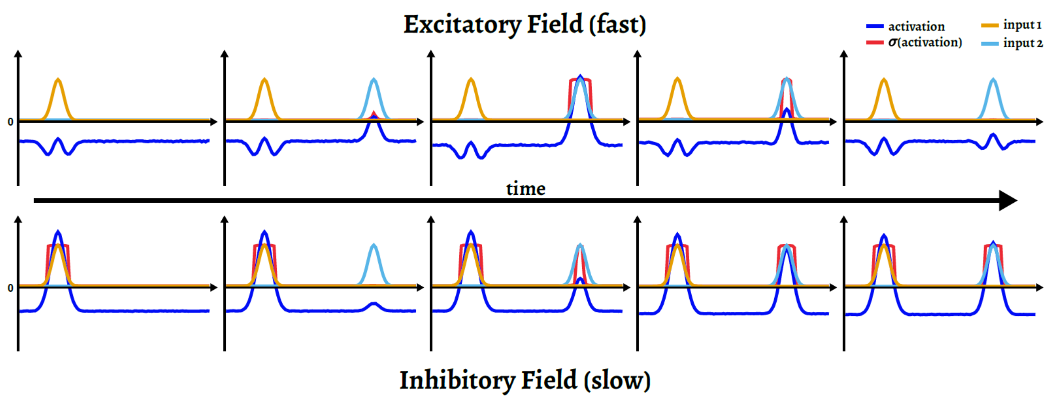

FIGURE 2.6: Schematic showing activation snapshots of a transient detector that has detected the formation of a newly formed peak in the source field. Figure 2.3.5 from (Tekülve, 2021)

Figure 2.6 depicts how the fast excitatory field of the transient detector, becomes destabilized by the slower inhibitory field . A new peak emerging from input 2 leads to the formation of a corresponding peak in the fast excitatory field. This peak represents that a transient event has been detected (top row, columns 2 and 3). After some time the slow inhibitory field also forms a peak corresponding to input 2. This peak inhibits the excitatory field, which pushes it through the reverse detection instability (columns 4 and 5).

2.4.3 Short-Term Memory

The architecture build in this thesis features a special kind of transient detector, which from its dynamics can be interpreted as a short-term memory representation. The short-term memory representation detects a newly formed field in a connected source field and stores its feature value for a short period of time, even if the corresponding peak in the source field has since decayed.

This is done by giving the fast excitatory field an interaction kernel that features a strong local excitation and a strong global inhibition. In addition the inhibitory field only receives input from the associated fast excitatory field and not from the source field. The modified interaction kernel puts the excitatory field into a self-sustained and selective regime. The selective dynamics enforces a capacity limit on the memory representation. As long as a peak is sustained in the excitatory field, no new input may be stored in memory. A stored peak decays as soon as the inhibitory field becomes active.

In previous work, capacity limits and memory decay were a result of decaying memory traces in memory substrate (Tekülve, 2021). This work uses selective and self-sustained transient detectors as a shortcut, since memories of more than just the last salient perception are not required in this thesis (see Section 3.5.5).

2.4.4 Match Fields

A peak in a match field represents the condition that the corresponding peaks of two connected fields overlap. It can be seen as a comparison of the content of two fields. When the content is similar enough, the match fields forms a peak at the corresponding feature location.

$$\begin{aligned}\tau\dot{u}_m(x,t) = &- u_m(x,t) + h_m + k(x) \circledast g(u_m(x)) \\ &+ c_{src_0 \rightarrow tar}(x) \circledast g(u_{src_0}(x,t)) + c_{src_1 \rightarrow tar}(x) \circledast g(u_{src_1}(x,t))\end{aligned} \tag{2.12}$$

The match field u_m receives input from two source fields u_{src_0} and u_{src_1}. The resting level h is tuned such that input from one source field is not sufficient to induce a supra-threshold peak. A peak in the match field then represents the feature value that both the connected fields currently encode the same feature.

A higher-dimensional match field can also be used with ridge input from two lower-dimensional fields along different feature dimensions. In that case, the match field forms a supra-threshold peak where the ridge input overlap. This can be seen as a binding of the two lower-dimensional representations into a unique higher-dimensional one. For example, a two-dimensional match field defined over color and position may receive ridge input from two one-dimensional fields, one encoding color, the other position. The match field then forms a supra-threshold peak at the position where the two ridge inputs overlap. This could for example then represent the unique position of an object with a certain color.

2.4.5 Behavioral Sequences

Complex behaviors arise from the integration of elementary actions into a unified process that organizes them in such a way that the more complex behaviour emerges. The organization of individual actions requires a mechanism with which they can be either initiated or terminated according to need. The elementary control unit (ECU; Richter, Sandamirskaya, and Schöner, 2012) uses a combination of boost and peak detector nodes to control the activation and inhibition of individual actions. The ECU enables the modular reuse of elementary actions for different cognitive functions. Complex behaviors emerge by sequencing ECUs into sequences.

Figure 2.7 shows two schematic representations depicting the structure of ECUs. The structure of the ECU can be divided into the control part and the content part. The control part consists of the intention node, the condition of satisfaction node (CoS) and, if necessary, the memory node. The intention node controls the initiation of the behavior defined by the content and the CoS node detects the successful execution. The memory node stores that the ECU was successfully executed in the past (if required).

The content of the ECU is determined by the intention field and the CoS field. The intention field defines the intended behavior. It projects ultimately projects onto the sensorimotor surface where it can drive action. The behaviour is represented by relevant features that uniquely define it. The CoS field detects whether the desired

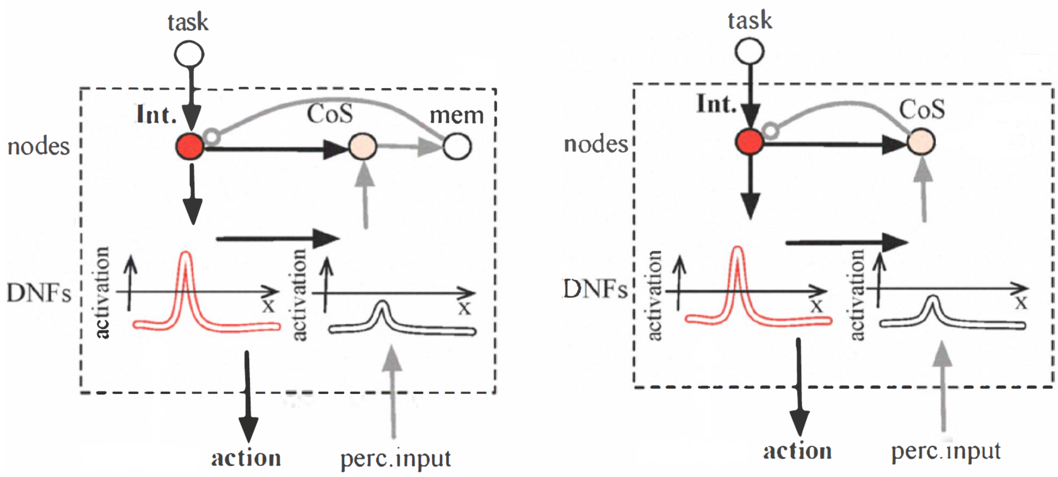

FIGURE 2.7: Schematic of the elementary control unit with (left) and without (right) memory representation. Adapted figure 2 from (Sandamirskaya, Richter, and Schöner, 2011)

behaviour has been successfully executed

$$\tau_{int}\dot{u}_{int} = -u_{int} + h_{int} + c_{tsk\rightarrow int}g(u_{tsk}) - c_{cos|mem\rightarrow int}g(u_{cos|mem}). \tag{2.13}$$

To initiate a behaviour, the intention node receives external input. This input can come, for example, from a perception or from another ECU. In addition, the intention node receives inhibitory input from either the cos or the memory node. This connection ensures that the intention is terminated as soon as the behaviour is completed (equation 2.13).

$$\tau_{intF}\dot{u}_{intF}(x) = -u_{intF}(x) + h_{intF} + c_{int\rightarrow intF}g(u_{int}) + c_{tsk\rightarrow intF} \circledast g(u_{tsk}) \tag{2.14}$$

The intention field receives input from the external task input, which creates a sub-threshold preshape. When active, the boost from the intention node pushed the sub-threshold peak through the detection instability. The content of an ECU, the desired end-state, is thus defined by the external task input. The intention field projects its peak to further fields, which finally end in the motor surface and set processes in motion that correspond to the desired behaviour. (equation 2.14)

The CoS field is a match field that receives perceptual input originating from the sensor surface and from the intention field. The CoS field detects the match between the desired final state and the currently perceived state (equation 2.15).

$$\begin{aligned}\tau_{cosF}\dot{u}_{cosF}(x) = &- u_{cosF}(x) + h_{cosF} + c_{intF\rightarrow cosF} \circledast g(u_{intF}(x)) \\ &+ c_{percF\rightarrow cosF} \circledast g(u_{percF}(x))\end{aligned} \tag{2.15}$$

The CoS field is connected to the CoS node. This node acts as a peak detector and represents the successful completion of the desired behaviour. In general the CoS node terminates the behavior controlled by the ECU (Figure 2.7 right). However, depending on the behavioral requirements, an intention node may be kept active, even after the CoS has been achieved (For example when lifting an object, the end effector grasping the bottle has to stay closed continually).

$$\tau_{cos}\dot{u}_{cos} = -u_{cos} + h_{cos} + c_{int\rightarrow cos}g(u_{cos}) + c_{cosF\rightarrow cos}\int g(u_{cosF}(x))dx \tag{2.16}$$

If the fact that a behavior has been completed is required for further processes (for example when sequencing different tasks over longer time periods), the ECU can utilize a memory node (Figure 2.7 left). The memory node receives input from the cos node and operates in the self-sustained regime (equation 2.17). Like the cos node the memory node terminates behavior, but remains active, which can influence behavior in other parts of the architecture.

$$\tau_{mem}\dot{u}_{mem} = -u_{mem} + h_{mem} + kg(u_{mem}) + c_{cos\rightarrow mem}g(u_{cos}) \tag{2.17}$$

Complex behaviors require the sequencing of individual ECU. For example, when controlling a robot arm to lift an object, it is first necessary to move the end effector toward the object, then the end effector has to grasp the object, only after which the robot arm can lift the object to a new location.

To enable the sequencing of different behaviors, ECUs are coupled together using precondition nodes (see Figure 2.8). Precondition nodes represent that a behavior represented by one ECU should only be initiated once all the necessary previous behaviors of the sequence have been completed. The task node activates the precondition nodes in combination with boosting the intention nodes of all ECUs in the sequence. These boosts in combination with the precondition nodes define the behavioral sequence.

$$\tau_{pr}\dot{u}_{pr} = -u_{pr} + h_{pr} + c_{tsk\rightarrow pr}g(u_{tsk}) - c_{cos|mem\rightarrow pr}g(u_{cos|mem}) \tag{2.18}$$

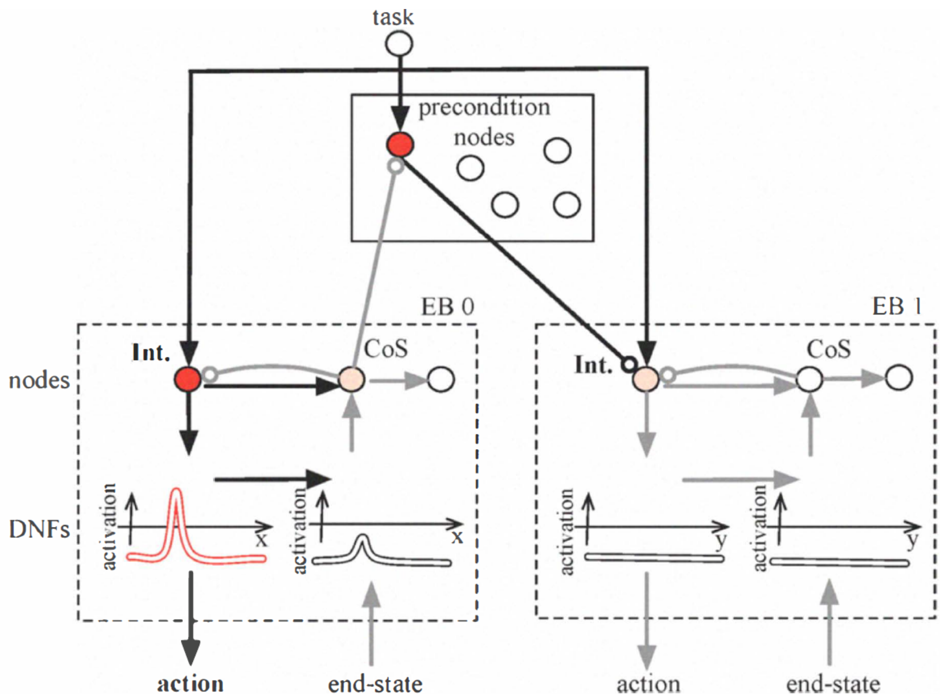

FIGURE 2.8: Schematic of how precondition nodes enable sequencing. Adapted figure 3 from (Sandamirskaya, Richter, and Schöner, 2011)

The precondition nodes receive input from the task node and inhibitory input from the CoS node of an ECU that signifies the precondition for an ECU later in the sequence (equation 2.18). The later ECU only initiates once the precondition node is deactivated by the CoS of the previous ECU.

2.5 Neural Plasticity

Neural plasticity describes a process in which the nervous system to change and develop new behavior, through changes in the connection patterns inside the nervous system. In the context of DFT describes a form of synaptic plasticity, in which field parameters and connections between fields change according to previously experienced activation patterns. Plasticity enables an architecture to form long lasting associations, learn new skills or to form long-term memory.

In DFT, the two main methods implementing a form of neural plasticity are memory traces and Hebbian connections. Both methods are based on Hebbian learning principles (Hebb, 2005). Memory traces model a form of Hebbian plasticity in the internal dynamics of a neural field. Hebbian connections model the formation of plastic connection patterns between neural fields.

2.5.1 Memory Traces

Plasticity internal to a neural field is implemented by the memory trace. Memory traces are associated with a field and either pass back inhibiting or excitatory input. The memory trace itself tracks the activation history of the connected field, by building up activation at feature values, where the connected field has previously formed peaks. The buildup activation in the memory trace then heterogeneously modifies the resting level of the neural field.

$$\begin{aligned} \dot{m}(x,t) =& \frac{r(t)}{\tau_+}(-m(x,t) + g(u(x,t)))g(u(x,t)) \\ &+ \frac{r(t)}{\tau_-}(-m(x,t))(1 - g(u(x,t))) \end{aligned} \tag{2.19}$$

Activation builds up according to equation 2.19. The evolution of activation $\dot{m}$ is determined by the timescale for the build up of activation τ_+ and the decay of activation τ_-. At feature values with supra-threshold activation in u the memory trace converges to $g(u)$, while at feature values without supra-threshold activation, the memory trace converges to 0. Depending on the decay timescale, the memory trace tracks the activation history over a longer or shorter time period. A Memory trace can also be coupled to a reward signal $r(t)$. In that case the memory trace only develops when the condition signifying the reward is met. In DFT the reward signal can be conceptualized as a peak detector node, that passes a reward to the memory traces when the condition symbolized by the peak detector node has been met. The memory trace then tracks the activation history over multiple reward periods.

2.5.2 Hebbian Connections

Hebbian connections provide plastic connections between neural fields and nodes of any dimension. In contrast to the fixed connections between neural fields discussed above in Section 2.3.3 these connections can be used without restrictions of the connected fields or the feature dimensions they code for. A Hebbian connections is a any-to-any connection in which a feature location of one field can be connected to any feature locations of the other field. The connectivity between the field locations is given by a weight pattern $l_{i,j}$ that determines the connection strength.

This weight pattern is plastic, meaning that it can develop according to a reward based Hebbian learning rule, or it can also be set manually. The Hebbian learning rule between two one-dimensional fields u_i and u_j over the feature dimensions x

and y has the form:

$$\dot{l}_{i,j}(x,y,t) = -\eta r(t) g(u_j(y,t))[l_{i,j}(x,y,t) - g(u_i(x,t))]. \tag{2.20}$$

The connection weight between to feature positions x_0 and y_0 strengthens and converge to 1, when both fields have supra-threshold peaks at these locations. If however only $u_j(y_0)$ is active the connection strength converges to 0. In places where u_j is below threshold the weight pattern remains fixed. In this way the Hebbian learning rule associates activity patterns between two connected fields. η is the learning rate and determines the speed of memory buildup and decay. $r(t)$ is the reward signal and activates the plastic formation of a connection weight pattern.

The plastic formation of weight patterns between fields can be used to provide an architecture the capability to learn associations between separate representations pertaining to different features. Hebbian connections can also be used for dimensionality reduction or expansion. Concept nodes in this thesis for example are connected to the continuous feature values they represent via fixed Hebbian connections (see Section 2.4.1) that are manually tuned.

In this work, Hebbian weights that are not manually set are initialized with zero-weights. Meaning that the connected fields first have to form associations of activation patterns before they pass activation to each other. When an association has formed, a peak in the source field can induce a peak in the target field, according to the learned weight pattern.

2.5.3 Belief Architecture

A more advanced form of learning is the abstraction of causal relationships between perceived events in the world. Learning such causal relationships can be seen as the formation of beliefs about how the world works. Beliefs are propositional, since they assert that some perceived events are caused by some other events that happened either before or at the same time.

In DFT a belief is represented by a *Belief Node* that is reciprocally connected to a set of *Role Concept Nodes*. These concept nodes represent the combined content of the belief. Each *Role Concept Nodes* pertains to some aspects of one of the related events, which defines its role in the causal relationship. Because of the reciprocal connections, activation of any element of a belief, leads to the recall of the entire belief representation. Reciprocal connections can be formed using Hebbian learning.

Tekulve and Schöner (2020) developed an intricate belief architecture, that enables an agent to autonomously form beliefs about action-outcome contingencies and to use these to guide action. An excerpt of the belief architecture is in Figure 2.9. The depicted belief system learned color mixing rules, by inferring a causal relation of how the color of a canvas changed when a coat of paint was applied to it.

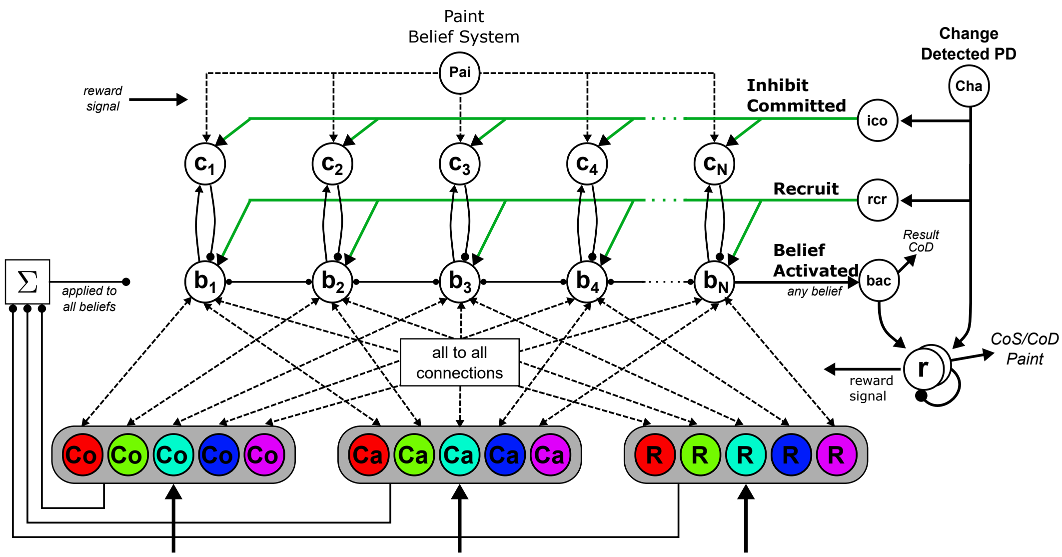

FIGURE 2.9: Excerpt from Tekülve's (2021) belief system for learning colour mixing rules. You can see the role nodes that represent the contingencies, together with the belief, commit and other nodes that organise the formation and recall of beliefs (Tekulve and Schöner, 2020)

The *Role Concept Nodes* form the feature representation of a learned contingency and are depicted at the bottom of Figure 2.9. The *Paint Cote* (Co) and *Canvas* (Ca) role nodes represent the performed action, while the *Result* (R) nodes represent the resulting color. Together, these nodes form a triplet, specifying the learned action-outcome association. For belief formation, all *Role Concept Nodes* receive perceptual or memory input, which represents the to be learned contingency.

Belief Nodes (b) represent beliefs about action-outcome contingencies and are embedded in the belief system. *Belief Nodes* are connected to the *Role Concept Nodes* with all-to-all bidirectional Hebbian connections. An active *Belief Node* activates the *Role Concept Node* triplet representing the contingency. The weight pattern of the reciprocal Hebbian connections determines the content of the belief.

The *Commit Nodes* (c) represent the commitment of a connected *Belief Node* to a learned contingency representation. A *Commit Node* receives input from the *Belief Node* it is associated with, from the *Inhibit Commited* (ico) node and is additionally connected via Hebbian connections to the *Belief System Node* (Pai). The *Belief System Node* is active per default and passes a boost, if the Hebbian connection has been strengthened in a previous learning episode.

A belief formation episode is initiated by the *Outcome Detection Node*. The *Outcome Detection Node* represents that a performed action has resulted in a perceived outcome. It activates the *Inhibit Commited* node, the *Recruit Node* (rcr) and the *Reward Node* (r). In combination with the *Belief Active Node* (bac), which detects if a *Belief Node* in the belief system is active, these nodes organize the formation of a new belief.

To learn a new belief, the learning process first involves selecting a *Belief Node* with which to associate the observed contingency. The selection process is controlled by the *Inhibit Commited* and *Recruit Nodes*. The *Inhibit Commited* operates on a faster timescale than the *Recruit Boost Node*. Thus, when learning is initialized by the *Outcome Detection Node*, the *Inhibit Commited* node activates first, followed by the *Recruit Node*.

The *Inhibit Commited* node inhibits *Belief Nodes* that are already associated with learned beliefs. This is done by boosting the *Commit Nodes*, which activates those that belong to an already commited *Belief Node*. To select a new *Belief Node* the *Recruit Node* passes a homogeneous boost. *Belief Nodes* are coupled inhibitorally with each other, leading to a selective dynamics. The recruit boost thus leads to a competitive selection of a non-committed *Belief Node*.

If the contingency represented in the *Role Concept Nodes* has been learned in the past, a *Belief Node* is already active before the *Recruit Node* activates. In this case, input from the *Role Concept Nodes* is tuned such, that the already committed *Belief Node* matching the current feature representation has an advantage in competitive selection. This makes sure that no new substrate is recruited to learn the same belief again.

The *Reward Node* receives input from the *Outcome Detection Node* and the *Belief Active* node. It activates when all connected nodes are active. The *Reward Node* gives a reward signal to the Hebbian connections between the *Role Concept Nodes* and the *Belief Nodes* as well as to the Hebbian connections between the *Commit Nodes* and the *Belief System Node*. The reward signal strengthens connections between active nodes via Hebbian learning (see Section 2.5.2). In this way, the *Belief Node* becomes reciprocally connected to the active *Role Concept Nodes*. The associated *Commit Node* becomes in turn connected to the *Belief System Node* signifying that the *Belief Node* is committed. Because of the stability of learned states, reward signals for already learned beliefs lead to little to no changes in the plastic weights.

Partial feature cues are sufficient to lead to reactivation of a belief. If the activated *Role Concept Node* has been associated with a belief in the past, it will activate the corresponding *Belief Node* through the reciprocal Hebbian connections. The active *Belief Node* will then in turn activate the *Concept Nodes* in the remaining two roles, thus recalling the complete action-object-outcome triplet representing the contingency.

2.6 Intentionality

This thesis uses the different elements and structures described above and integrates them into a unified architecture that is organized along the lines of the process model of intentionality implemented by Tekülve (2021). Searle (1983) categorizes intentional states according to their psychological modes. He differentiates 6 psychological modes based on their direction of fit (DoF). Mental states of the mind-to-world DoF, represent some property or event of the environment (how the world is). These include perception, memory and belief. In contrast, mental states that represent to be produced properties or events in the environment are of the world-to-mind DoF (how the world should be). Psychological modes of the world-to-mid DoF include desire, prior-intention and intention in action. Tekülve (2021) has demonstrated that an embodied DFT model implementing all 6 psychological modes can autonomously learn and act in its environment.

Figure 2.10 depicts a simplified schematic of how the psychological modes are differentiated along the lines of their DoF (horizontal separation). In DFT intentional states correspond to supra-threshold neural nodes or peaks in neural fields. As they are always be linked to the sensorimotor surface, they represent either a state of the perceived environment or a desired state of the environment. An important part of an intentional state is its condition of satisfaction (CoS). It represents if the state of affairs its referring to is currently achieved. For Intentional states of the mind-to-world DoF, this is generically the case. An active node or peak constitutes its own CoS. This is not the case for the intentional states of the world-to-mind DoF. As they represent a desired state of affairs that has (potentially) not been achieved yet, its CoS is separate from the neural representation of the intentional state. The CoS is achieved, when the perceived state of the environment matches the desired state specified by the intentional state. In addition to the CoS, intentional states of the world-to-mind DoF can also end in a condition of dissatisfaction. This is the

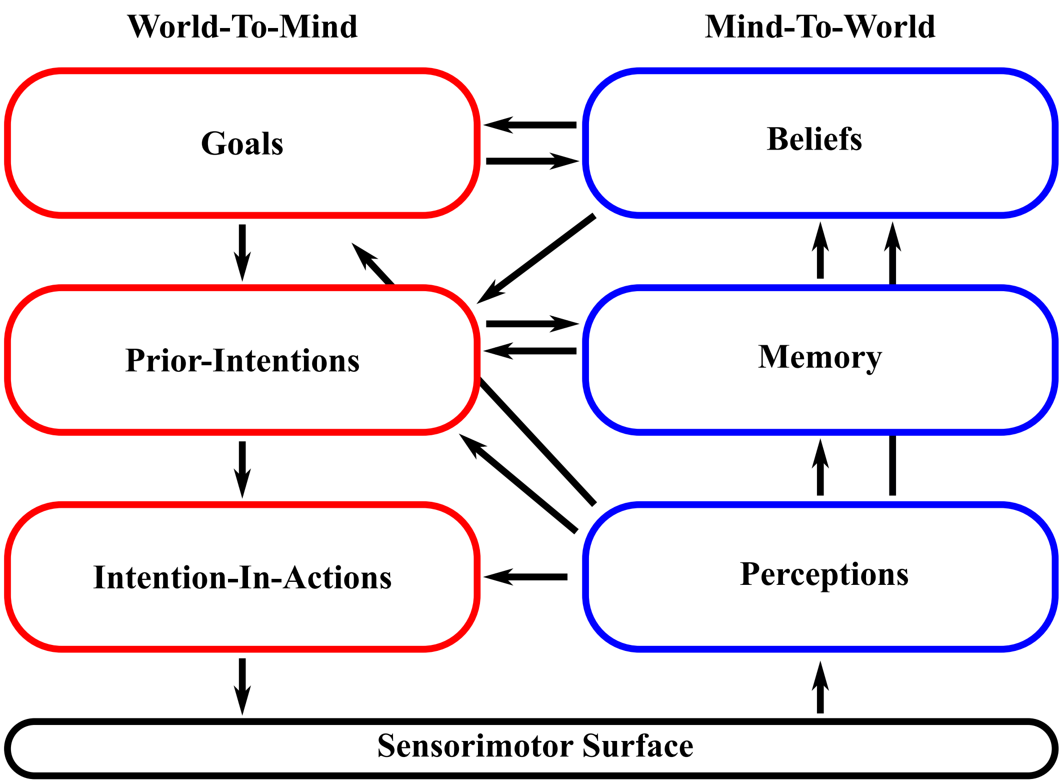

FIGURE 2.10: Schematic of how the different psychological modes can be differentiated according to their direction of fit (DoF) and their distance to the sensorimotor surface. Modes of the world-to-mind DoF are outlined red, and blue when of the mind-to-world DoF.

case either if the desired state of affairs is not achievable, or if an attempt to bring it about failed.

The psychological modes of each DoF can be ordered based on their distance to the sensorimotor surface (vertical separation). Distance here meaning how directly they couple to the surface, which is linked to their level of abstraction and their ability to generalize. Intentional states that are always directly coupled to the sensorimotor surface are perceptions (when receiving input) or intention-in-actions (when projecting input). Perceptions only activate when receiving direct input from sensors. They represent perceived features and properties of the environment. An active intention-in-action on the other hand always drives motor systems. Depending on the type of the intention-in-action, its content can be represented along continuous feature spaces (e.g. representing end effector positions) or as binary state descriptions (e.g. representing the state of an LED).

Intentional states that are not directly linked to the sensorimotor surface, are grounded indirectly through their eventual connection to the intentional states of perception and intention-in-action. On the mind-to-world side, these include intentional states belonging to the psychological modes of memory and of beliefs. Memory states represent memorized features of past perceptual states, that can for example be used by the architecture to inform current action plans. Beliefs are propositional in nature, since they assert a causal relation between different previously perceived events. The content of a belief representation therefore depends on its inner structure, which abstracts rules from past experiences. This separates them from memory, since memory states only recall past perceptions without any notion of their causal relation.

On the world-to-mind side, the intentional states that are not directly linked to the sensorimotor surface belong to the psychological modes of either goal or of prior-intention. Prior-intentions can be seen as action plans that require the organization of multiple lower level intention-in-actions into action sequences. A prior-intention itself is however also dependent on top-down input coming from the goal representation. A goal represents a desired state of affair in the environment that is not part of some other higher-level world-to-mind process. It can be seen as the highest level prior-intention of the architecture. Depending on the state of the architecture, a neural representation implementing a world-to-mind intention may sometimes play the role of prior-intention, if it is part of a higher level goal, while at other times it may itself be conceptualized as a goal if it is not part of any other prior-intention. This hierarchical structure is implemented in DFT by building nested sequences of ECUs, in which the content of a higher level ECU (representing a prior-intention) initiates a sequence of lower level ECUs. These may in turn also be prior-intentions and project to even lower level ECUs. Eventually this nested sequence projects onto ECUs representing the intention-in-actions. These project directly onto the motor surface and thus control the elementary motor behavior of the agent. This kind of hierarchical structure is very flexible, as higher level prior-intentions can be reused in a modular fashion by other higher-level intentions (Tekülve et al., 2019).

On the mind-to-wold side, perceptions are implemented utilizing the different instabilities and concept representations described in Sections 2.3.2 - 2.4.3. As the short-term memory system described in Section 2.4.3, is really a transient detection in memory mode, it is best situated in the psychological mode of perception. Beliefs are implemented using the belief architecture described in Section 2.5.3.

Chapter 3

Architecture and Experiment

The previous chapter presented the theoretical background that guide the structure of the neural dynamic architecture developed in this thesis. The architecture is designed to control a small robot and engage in a simulated scenario. As described in Chapter 1, the architecture should demonstrate the capability to select and reach its goals on the basis of known action-outcome contingencies. Initial stabilization against opportunistic activation and later habituation should be reflected in goal selection decisions. The models behavioral dynamics are examined in a simulated multi-phase toy experiment.

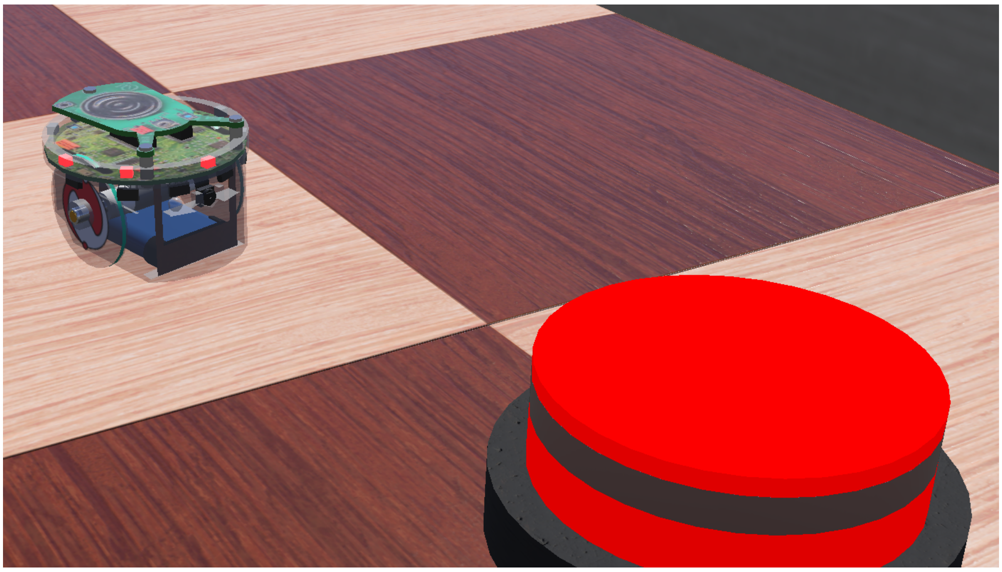

FIGURE 3.1: Screenshot of simulated environment in Webots (Olivier, 2004). The E-Puck robot on the left is blinking three of its front LEDs toward the red object on the right.

Section 3.1 provides an overview of the scenario. This includes a description of the simulated environment and the basic behavior of the agent robot. Section 3.2 describes the different experiment phases. Section 3.3 provides some technical details about the robots sensors and actuators, their connection to the DFT-architecture, and the general functionality of objects and supervisor. Section 3.4 provides a general description of the structure and flow of the architecture designed for the experiment. Sections 3.5, 3.6 and 3.7 finally give a detailed description of the architecture. Section 3.5 describes the sub-architectures of the mind-to-world direction of fit. Subsequently, section 3.6 describes the connection to the robots actuators and the organization of outcome-oriented behavior. Section 3.7 then describes the Belief system and its embedding in the rest of the architecture.

3.1 Scenario Description

The simulated environment consists of an arena, a supervisor, a set of colored objects and the agent robot. Figure 3.2 depicts a schematic overview of the scenario environment. The experiment is controlled by the supervisor and takes place within the arena. For this, the supervisor places the agent robot in the arena together with a set of colored objects. The agent robot is a simulated E-Puck, which is a small robot on two differential wheels that features a camera, microphone, and an LED array (see Figure 3.1). The robot is equipped with the ability to navigate the arena and to interact with objects by performing one of 3 actions. Actions consists of blinking 3 LEDs in sequence. They are differentiated by which LEDs were used in the sequence. Actions are object-oriented, as the E-Puck has to stand in front of a target object, for it to result in a perceivable outcome.

Objects can be seen as emulating colored buttons that can be interacted with in different ways (e.g. pushing, turning, pulling, ...). They are differentiated by their color (see figure 3.1) and can play sounds when certain actions are performed on them. Whether and which sounds are played is determined by action-outcome contingencies. A contingency relates a performed action of the robot to an outcome sound the object plays. Each object has a set of 3 contingencies that determines how an object of that color responds to each of the 3 actions. A contingency can include one of three outcome sounds (high-, medium- or low-pitch) or it can be a null contingency, in which case the object does not respond to the specified action (see figure 3.2). Causing outcome sounds is postulated to be rewarding for the agent robot, which supports reward based learning and justifies opportunistic activation.

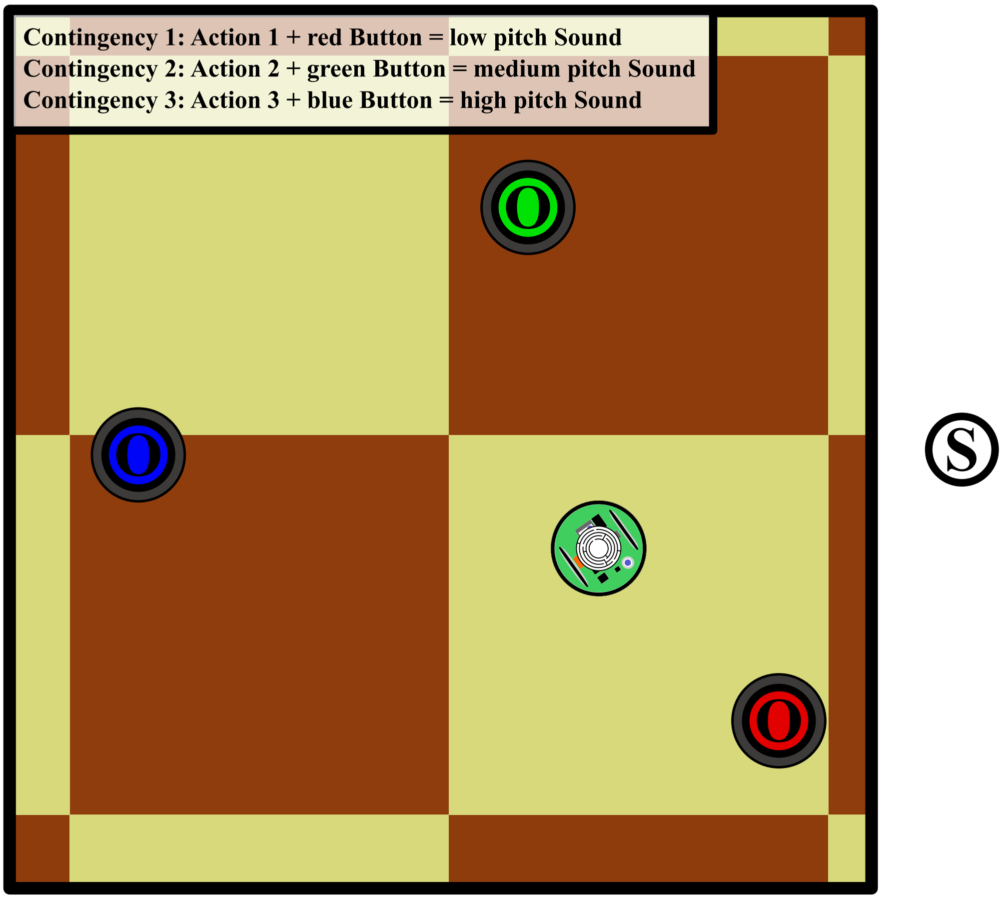

FIGURE 3.2: Schematic of scenario environment. Colored circles represent objects (O). The E-Puck is represented by the middle circle. An example set of contingencies is listed in the legend. For example contingency 1, if the robot were to perform action 1 on a red object, it would play a low-pitch sound. The supervisor is shown to the side (S).

It is the E-Pucks tasks to learn these contingencies and to later use this knowledge to select and achieve desired goal sounds. The supervisor organizes the different experiment phases. It controls the positions of the objects in the arena and can add and remove objects from the arena. In addition, the supervisor gives task signals to the E-Puck, which encourage the robot to display certain behaviors. These signals are used to communicate the different experiment phases.

3.2 Experiment

The experiment consists of multiple phases, each of which is designed to probe certain characteristics of the E-Pucks behavior. The first phase demonstrates the capability of the E-Puck to explore the action space, learn action-outcome contingencies, and to opportunistically engage in goal-oriented behavior (see Section 3.2.1). The following three test-phases test if the E-Puck was able to learn the action-outcome contingencies of the environment (see Section 3.2.2). In the goal-selection phase, the dynamics of goal selection decisions are examined. This phase investigates if the dynamics at the level of goals account for early familiarity and eventual novelty preference of goal selections (see Section 3.2.4). The final phase is the goal-stability phase. This phase examines the stability of active goal states over multiple goal reaching attempts. The stability is probed by presenting the E-Puck with distractor objects, which may destabilizes currently active goal states via opportunistic activation. Early familiarity preference should initially stabilize active goal states against opportunistic input (see Section 3.2.4).

TABLE 3.1: Action-Outcome Contingencies used in the experiment.

Actions	Object Color	Sound Pitch
1	red	low-pitch
2	green	medium-pitch
3	blue	high-pitch
3	yellow	low-pitch

The different phases of the experiment are simulated in continuous time and in the same arena. The course of the individual phases is controlled by the supervisor. This is done via the task signal and by the sequence object placements. The task signal is send to the E-Puck. It tells the E-Puck the current phase of the experiment and influences behavior in accordance with the task (See section 3.3.5). Contingencies are fixed at the beginning of the simulation and are listed in table 3.1. After each phase, memory traces in the architecture are reset. This avoids interference from earlier phases.

3.2.1 Learning Phase

The first experiment phase tests whether the architecture can learn contingencies in the environment. For this purpose, the arena is filled with a number of different colored objects at random locations. The task signal from the supervisor motivates the E-Puck to interact with objects in its field of view (see Section 3.6.6). While navigating in the arena, the E-Puck perceives new objects on which it performs actions. After each action, the supervisor shuffles the positions of the objects around, such that they are distributed in different random locations. This is done to ensure that the E-Pucks comes across different objects in a somewhat even manner, as it navigates the arena. The supervisor keeps track of which actions were performed on which objects. After all action-outcome combinations have occurred at least once, the phase is over and the test-phases begin.

As the robot starts the experiment with no contingency knowledge, it should initially display a random action pattern. Novelty preference in action selection should disfavor actions that previously had no rewarding outcome, which should guide exploration of the action-color space. As the E-Puck gains knowledge of action-outcome contingencies it should switch to a more exploitative behavior as it opportunistically generates desired outcome sounds (see sect 3.6.7).

3.2.2 Test Phases

The three test-phases demonstrate that the E-Puck has indeed learned the different action-outcome contingencies. In addition, these phases are designed to show the capability of the E-Puck do opportunistically switch between strategies to facilitate desired outcomes, and to ignore distractor objects. To show this, the supervisor provides a goal sound to the E-Puck, via the task signal. The task signal activates the "low-pitch", "medium-pitch" or "high-pitch" goal representations in the architecture (see Section 3.6.6).

To test if the E-Puck has learned to achieve these goal, it is presented with a sequence of different objects. After an object has been placed in front of the E-Puck, the supervisor records weather or not it has performed an action on that object and if a sound was produced. If no action is carried out in a certain amount of time, the supervisor will count this as the E-Puck ignoring the object. In each phase the sequence of objects presented to the E-Puck is first a distractor object, followed by a target object that can be used to reach the goal. Afterwards both the distractor and the target object are shown at once.

After behavior for each goal has been tested, the three test-phases are over and the goal selection phase begins.

3.2.3 Goal Selection Phase

The goal selection phase examines selection decisions at the outcome level. The E-Puck can freely choose a goal sound it wants to produce and is then presented with a target and followed by a distractor object. The goal selection decision is facilitated by the selection signal of the supervisor. This signal activates the goal representations of the E-Puck. Their selective dynamics leads to a single active goal state (see Section 3.6.6). Distractor objects probe goal stability, as opportunistic activation of contingency knowledge associated with that object favors a different goal. After reaching a goal, the active goal representation decays, at which point a new goal selection signal can initiate a new goal selection decision.

Goal selection decisions, produced sounds and performed actions are being recorded by the supervisor over multiple goal selection decisions. This generates a time course of that measures how selection decisions evolve over time. Initial stabilization as a selected goal is being achieved should lead to an early familiarity preference. Eventual habituation should lead to a change in selection decisions, indicating a kind of novelty preference in selection decisions.

3.2.4 Goal Switching Phase

The goal switching phase probes the stability of an active goal representation over multiple goal reaching attempts. In contrast to the goal selection phase, the E-Puck chooses one goal sound and is then presented with a long sequence of different distractor and target objects. Instead of initiating a goal selection decision after each goal reaching attempt, the selection signal stabilizes an active goal state such that it doesn't decay (see Section 3.6.6). In this way, the E-Puck chooses a goal at the beginning of the goal switching phase and then tries to achieve it over multiple goal reaching attempts.

As the E-Puck is presented with different objects and achieves its desired outcome, the selected goal should eventually habituate. The supervisor observes the current goal state and probes their stability by presenting distractor objects. Opportunistic activation from these distractor objects can destabilize the current goal and induce a new goal. The time course of active goals and the E-Pucks behavior, demonstrate how goal representations and their stability evolve as they habituate over multi goal reaching episodes.

3.3 Technical Details

The agent robot and the environment is simulated in Webots, a mobile robot simulation software (Olivier, 2004). The DFT-Architecture is implemented in Cedar, a software to implement embodied DFT-Architectures (Lomp et al., 2016). The DFT-Architecture interfaces with the robot controller in Webots through TCP read and write sockets. TCP sockets constitute the motor surface (in the case of write sockets) and the sensor surface (in the case of read sockets). It is this connection that grounds the architecture in a simulated sensorimotor surface. The full implementation of the DFT-Architecture, the Webots scenario and an exemplary data set is provided in a GitHub repository[1].

3.3.1 Movement

By default, the E-puck moves in a simple obstacle avoidance pattern. Obstacle avoidance is handled in Webots by the robot controller. As long as the E-Puck encounters no obstacle (either the arena walls or objects), it travels in straight lines. When it encounters an obstacle the heading direction is adjusted to avoid collision. The E-Puck possesses 8 directional infrared distance sensors spread evenly across the outer ring of the chassis. Sensors provide a continuous stream of distance readings[2]. The sensors position on the chassis and its sensor reading give information on the direction and distance of an obstacle.

Heading direction and obstacle avoidance is controlled in an attractor dynamics approach, in which the change in heading direction $\frac{d\Phi}{dt}$ is modeled as a dynamical system (Bicho, Mallet, and Schöner, 1998). Each distance sensor reading gets translated into a repellor state along the heading direction dimension, while the target heading direction is represented as an attractor state

$$\frac{d\Phi}{dt} = \sum_i f_{sensor_i}(\Phi, d_i) + f_{target}(\Phi). \tag{3.1}$$

Here Φ is the heading direction of the E-Puck, while f_{sensor_i} are repellor forcelets and f_{tar} is the target attractor forcelet. The repellor strength f_{sensor_i} depends on the

[1] https://github.com/Sehrispk/Neural-Dynamic-Account-of-Ideomotor-Theory.git

[2] The E-Pucks distance sensors have a maximum detection range of $d_{max} = 70\,\text{mm}$.

sensor reading and is modeled by:

$$f_{sensor_i}(d_i) = \beta_1(\Phi - \psi_i)\exp\left(\frac{-d_i}{\beta_2}\right)\exp\left(\frac{-(\Phi-\psi_i)^2}{2\sigma^2}\right), \tag{3.2}$$

where Ψ_i is the position of the i_{th} distance sensor, and d_i its distance reading. The parameters β_1, β_2 and σ control the overall strength and width of the forcelet. The target attractor is modeled by:

$$f_{target} = -\lambda \sin(\Phi - \psi_{target}), \tag{3.3}$$

where Ψ_{target} is 0 by default, as the dynamics are calculated in the E-Pucks egocentric reference frame. The parameter λ controls the strength of the attractor. At each time step the change in heading direction is calculated using Equation 3.1 and transformed into wheel velocities. The DFT-Architecture can set a new target heading direction Ψ_{target} by forwarding it through the sensorimotor surface (see Section 3.6.2).

The traveling velocity is set to a constant value v_0. In order to perform actions, the E-Puck stops at a specific distance in front of objects. For this, the DFT-Architecture can forward a brake signal through the sensorimotor surface, which sets the velocity to $0\,\frac{\mathrm{m}}{\mathrm{s}}$ (see Section 3.6.3).

3.3.2 LEDs

The E-Puck has 10 LEDs. Eight of these LEDs are distributed evenly on the outer ring of the chassis. The 9th LED is located on the body and the 10th right next to the camera, at the front. The DFT-Architecture can control each LED status by a set of neural nodes. Their status is routed through the sensorimotor surface. When a node is active, the corresponding LED is switched on (see Section 3.6.4). The current status of the LEDs is passed to the architecture, where they are coupled to the LED detection nodes. The perceived LED status is used to determine the progress of an action (see Section 3.5).

The different actions the E-Puck can perform are symbolized by LED sequences that are carried out by activating three LEDs in a row. Blinking the LEDs 1-3 in a sequence corresponds to action 1, LEDs 4-6 to action 2 and 7-9 corresponds to action 3.

3.3.3 Camera

Visual input is captured from a 52x39 pixel camera on the front of the E-Puck. The captured color images are forwarded through the sensorimotor surface in real time. Camera images are initially preprocessed in Cedar. For this purpose, the color channels are extracted from the HSV image and grouped into 20 bins. The resulting images are then contracted vertically and merged into a 20x52 pixel image. For each horizontal image position the summed color intensity of each color bin is obtained.

3.3.4 Supervisor

A supervisor robot in Webots has special access to the world tree of the simulation. This enables the supervisor to read and set properties of other objects and robots in the simulation. In addition, a supervisor can create and delete objects and robots as well as pass arguments to robot controllers. In this scenario, a supervisor robot is used to setup and manage the experiment. The supervisor places target and distractor objects in the environment and (re)sets the E-Pucks position. It also measures trajectories, played sounds, action decisions, and goals, which is later used to evaluate the E-Pucks behavior. The supervisor is supposed to emulate the role of a "real" supervisor in a laboratory experiment. To this end, it communicates with the E-Puck, by giving tasks and detecting reported goals.

3.3.5 Communication Channels

The communication between the supervisor, objects and E-puck is implemented with emitters and receivers in Webots. These simulated devices emulate the communication with radio, Bluetooth or infrared signals. The different communication channels are depicted in Figure 3.3.

The E-Puck receives task signals from the supervisor and perceived sounds from objects[3]. Task signals reflect the current phase of the experiment and represent a task given to the E-Puck. The signal is passed on to the DFT-Architecture through the sensorimotor surface, where it influences its behavior. Sound signals are emitted by objects. Played sounds are distinguished by their frequency. A sound vector is passed to the architecture, in which each entry corresponds to the amplitude in a

[3] Webots has no support for audio simulation. To get around this limitation, emitters and receivers send/receive audio signals on a dedicated channel.

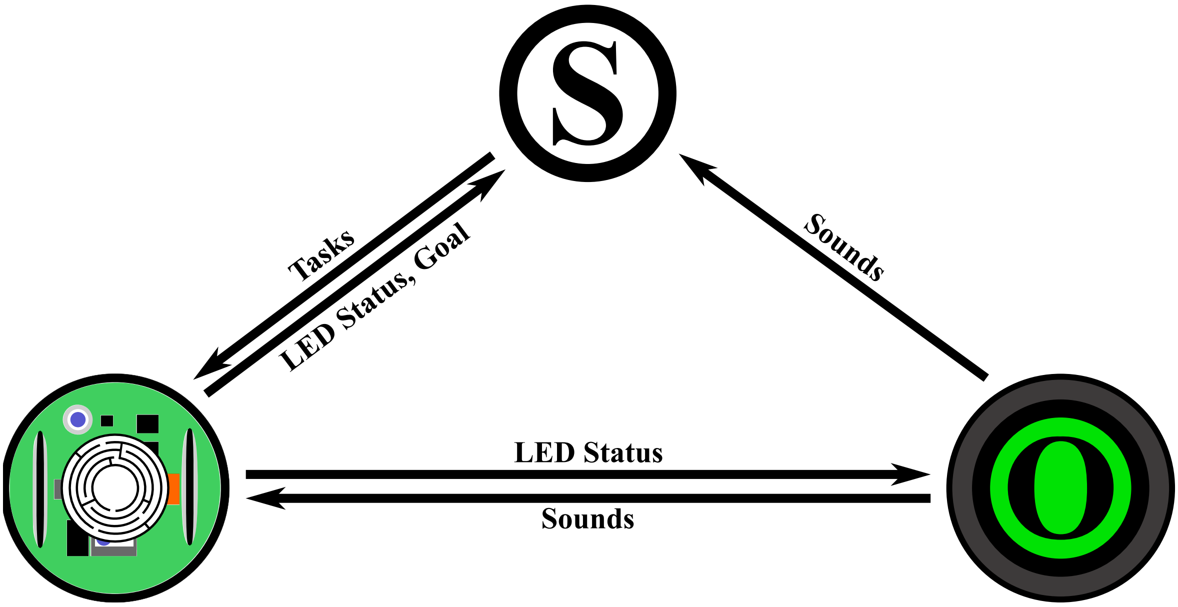

FIGURE 3.3: Schematic of the different communication channels between Objects (O), Supervisor (S) and E-Puck. The supervisor transmits task signals to the E-Puck and records its current goal and LED status. The E-Puck transmits its LED status to objects and supervisor. Sounds are played by objects and are transmitted on a separate channel.

certain frequency range. This vector is passed on to the DFT-Architecture through the sensorimotor surface (see Section 3.5.1).

The E-Puck transmits its current goal and LED status to the supervisor and to objects in the arena. The LED status is used to detect if the E-Puck produced an action. The object closest to the E-Puck produces a sound if the performed LED sequence corresponds to a one of its action-outcome contingencies. The goal is set selected by the DFT-Architecture and send to the robot controller through the sensorimotor surface. The supervisor uses the current goal of the E-Puck to determine which objects count as targets or distractors.

3.4 Architecture Overview

The E-Pucks behavior is determined by a dynamic field architecture that connects to its sensors and actuators. The architecture realizes tasks by forming intentional states that represent perceived features and events of the environment (mind-to-world) as well as intentional states of to be produced features and events (world-to-mind).

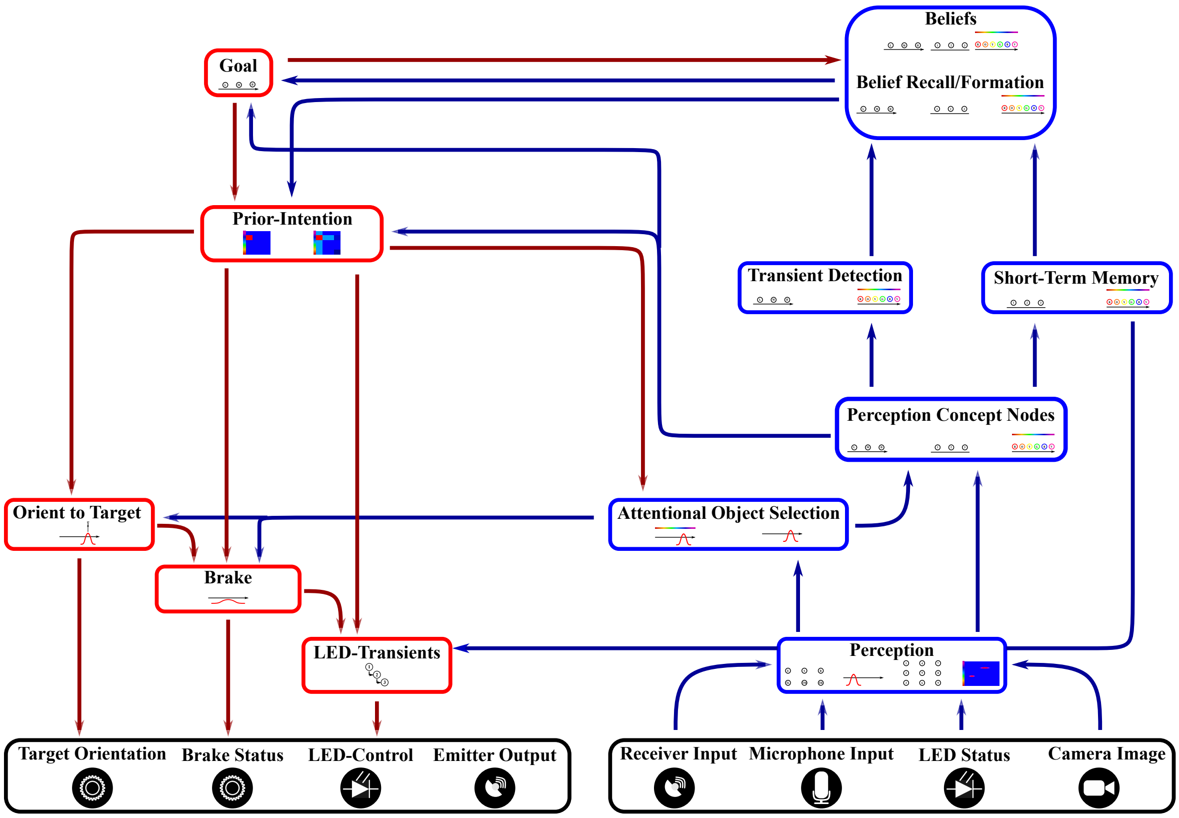

FIGURE 3.4: Schematic overview of the DFT-architecture. Modules are separated into their respective role of mind-to-world (right side) or world-to-mind direction of fit (left side). The architecture is grounded through its connection to the sensorimotor surface (bottom). Blue boxes indicate representations of the mind-to-world DoF and red boxes of the world-to-mind DoF.

Figure 3.4 shows a schematic overview of the intentional states realizing the agents behavior. The different intentional states are separated into their respective role of mind-to-world and world-to-mind direction of fit (DoF). Starting from the mind-to-world DoF, the general flow can be summarized as follows.

As the E-Puck navigates the environment, its sensors provide a continuous stream of data that contain information about the current state of the environment. The *Perception Fields* extract relevant data from this sensor stream and form representations of perceived objects and features (see Section 3.5.1). The architecture then selects a single object in its field of view and represents it with the dynamic fields of the *Attentional Object Selection* (see Section 3.5.2). *Perception Concept Nodes* then discretize continuous perceptual representations into concept representations, leading to a discrete concept representation of attended object, perceived sounds and performed actions. The more abstract representations of *Transient Detection* and *Short-Term Memory* extract changes of the perceived environment and stores past perceptions of the *Perception Concept Nodes*. Together these two representations tell the E-Puck if performed actions caused a notable outcome and signal if a cognitive event has happened that probes the belief system (see Sections 3.5.4 and 3.5.5).

The *Belief System* enables the E-Puck to learn and recall action-outcome contingencies. *Belief Recall* and *Formation* intentions organize which cognitive events lead to cuing of the *Belief System* and which events lead to the formation of a new belief. A belief is learned if the E-Puck perceives a sound after it has performed an LED-Transient. The belief then represents the contingency consisting of the performed action, the target object and the perceived sound. Beliefs can be recalled either as a result of the activation of a new goal state or as a result of perceiving a new object in the field of view. The former recall path enables outcome-oriented behavior by providing a strategy to guide action, while the latter recall path causes opportunistic activation of goal states as a result of a perceived object (see Section 3.7).

While the mind-to-world DoF forms representations of perceived features, events and learned beliefs, the intentions of the world-to-mind DoF enable the E-Puck to perform LED-Transients on target objects and to behave in an outcome oriented fashion. This is achieved by hierarchy of nested intentions that are grounded in the motor surface and thus have control of the E-Puck actuators. Individual intentions are represented as elementary control units (ECUs). They influence behavior through their connection to the sensorimotor surface and detect their condition of satisfaction CoS via their connection to the perceptions of the mind-to-world direction of fit. Intentions are organized into sequences using precondition nodes.

Starting from the motor surface (bottom left of Figure 3.4), the three intention in actions *Orient to Target*, *Brake* and *LED-Transient* are directly linked to the motor surface and represent motor primitives available to the DFT-Architecture (see Sections 3.6.2, 3.6.3 and 3.6.4). The *Orient to Target* intention orients the E-Pucks heading direction toward a specified target object. The *Brake* intention makes the E-Puck stop in front of a target object. The three *LED-Transients* constitute the possible object interactions available to the E-Puck. Each one performs an LED-Transient by activating three LEDs in sequence (see Section 3.3.2).

The three motor primitives are chained into an action sequence by the *Prior-Intention*, that specifies a target object and the LED-Transient to perform. The *Prior Intention* can activate either as a result of a task signal from the supervisor or as a result of an active goal, in which case the performed action is determined by a recalled strategy.

The *Goal* representation is the highest level world-to-mind intention of the architecture. It represents a desired outcome sound and organizes goal oriented behavior by recalling an action strategy through the belief system and activating the *Prior Intention* to engage in outcome-oriented action. A *Goal* can activate either as a result of a supervisor signal or by opportunistic activation through the recall of an action-outcome contingency initiated by perceiving a new object (see Section 3.6.6).

The content of *Goal* and *Prior Intention* representations are modulated by memory traces. These memory traces guide exploration, lead to initial stabilization and eventual habituation of selected goals (see Sections 3.6.6 and 3.6.7)

3.5 Mind-to-World

Figure 3.5 depicts the sub-architectures belonging to the psychological mode of perception. The E-Puck perceives the world with its camera, microphone, receiver and LED sensors. Features that are relevant to the given scenario are extracted from sensor data, which serves as input to perception fields. From this perceptual representation, the architecture abstracts relevant information by forming feature representations of attended objects and extracting discrete concept representations from lower level continuous perceptual features. Ultimately the meaning of these representations is determined by their internal dynamics and their ultimate connection to the sensor surface. These representations make their contents available to other parts of the architecture where they can initiate or terminate actions, activate intentional states, or lead to the formation of action-outcome beliefs.

3.5.1 Perception

The bottom part of Figure 3.5 depicts the perception fields and nodes which are directly linked to the sensorimotor surface and form the base representation of the perceived environment. This representation is already prepossessed in the sense that relevant information is first extracted from raw sensory data, before it is passed on to the architecture (see Sections 3.3.2 - 3.3.5). Other parts of the sensory stream are disregarded. This implicitly assumes a form of attention, as the architecture only forms representations about environmental features that are necessary for the task

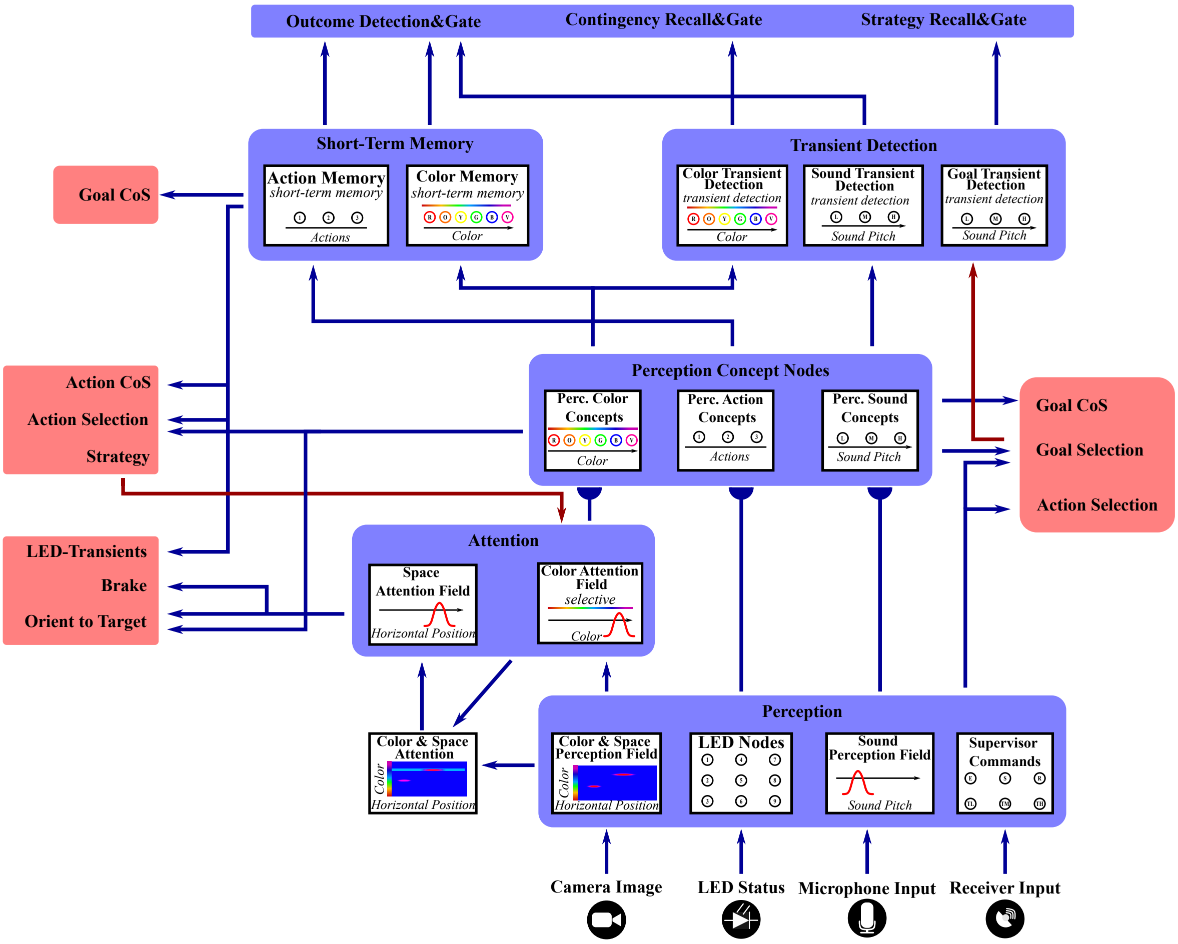

FIGURE 3.5: Schematic of the sub-architectures that make up the the psychological mode of perception of the mind-to-world direction of fit. The depicted networks form representations about states and events in the environment. Each perceptual representations is ultimately grounded through perception fields in the sensorimotor surface.

at hand. This kind of selective attention is inherently necessary to engage in meaningful behavior[4]. This is justified by assuming an analogous situation to a "real" laboratory experiment, in which the participant has already been briefed about the rules of the experiment and is thus primed to pay attention to corresponding features.

The two-dimensional *Color/Space Perception Field* receives input from the preprocessed camera image. Interactable objects have a higher color saturation compared to their surroundings and thus stand out from the background. The *Color/Space*

[4]There is a virtually infinite number of environmental features one could attend to. Since human attentional resources are limited, it has to be focused on information most important for a given task. The minds ability to do this touches on the topic of cognitive control (Botvinick and Braver, 2015).

Perception Field is tuned such that background noise remains below the detection threshold. The field is a representation of objects perceived in the environment. It distinguishes objects by their color and contains information about their horizontal positions and radial extent on the camera image. Since objects in this scenario are always of the same size, the latter is a measure of the objects distance. The *Color/Space Perception Field* is connected to the *Color Attention Field* and the *Color/Space Attention Field*.

The *Sound Perception Field* represents perceived sounds as activation along the pitch dimension. Sound input is prepossessed in Webots and thus noise free and normalized (see Section 3.3.5). The *Sound Perception Field* is connected to *Sound Concept* nodes.

The perception *LED Nodes* indicate the status of the robots LEDs. Each LED is represented by one node. A node above threshold indicates that the LED is turned on. The bodily self-perception of the E-Pucks LEDs enables the architecture to perceive the completion of an action it is currently engaged in. For this, the perceptual *LED Nodes* project their activation to the *Action Concept* nodes.

The *Supervisor Command* nodes act as binary detectors and represent the tasks transmitted by the supervisor. They are connected to the *Goal Selection* and *Action Selection* nodes, which are part of the *Goal* and *Prior-Intention* of the world-to-mind DoF. This establishes a link between the tasks given by the supervisor and the architectures intentions of the world-to-mind DoF. It is this link which is responsible for the different behavioral patterns of the E-puck in different experimental phases (see Section 3.6.5). The communication between supervisor and E-Puck serves as a simplified placeholder for a more detailed model. One can easily imagine a scenario in which communication takes place via visual or audio signals which are then translated by the E-Puck into task-concept-representations by means of appropriate cognitive processes. Since this is not the focus of this thesis the shortcut described above is taken, whereby cognitive processes containing signal information are combined into a single concept node directly linked to the sensor surface.

3.5.2 Attention

For object-oriented action, the E-Puck focuses attention on one object at a time. This is done using two one-dimensional attention fields, which are depicted just above the perception fields of Figure 3.5.

The *Color Attention Field* is selective and receives contracted input from the *Color/Space Perception Field*. A peak going through detection instability represents a selection decision and indicates that the E-Puck focuses its attention on objects of the corresponding color. The *Color Attention Field* receives additional sub-threshold input from the *Strategy* representation, which is part of the world-to-mind DoF (see Section 3.6). When a strategy has been selected for a given goal, the sub-threshold *Strategy* input selectively boosts the target color values. This top-down contribution modulates attentional selection such that inputs matching the target color are boosted and ultimately selected over inputs from objects of different colors. This connection can be seen as a shortcut that implements a simple form of feature guidance. A more comprehensive model could feature a complete scene representation, in which target object could be searched via top-down and bottom-up feature cues (Grieben and Schöner, 2021). Without top-down contribution the selection decision is random. The content of the *Color Attention Field* is extracted by the *Object Color Concepts* nodes and are thus made available to higher level cognitive processes.

The *Space Attention Field* receives input from the *Color/Space Attention Field*, which uses the ridge input from the *Color Attention Field* to extract the position associated with the selected object color. The *Space Attention Field* field projects its contents to the intention in actions *Orient to Target* and *Brake*, where it influences movement and their CoS. Together the two one-dimensional attention fields form a unique representation of an attentionally selected object.

3.5.3 Concepts

The aforementioned *Color Concept*, *Sound Concept* and *Action Concept* nodes categorize perceived feature values into more abstract discrete concepts. They are connected to perception fields via Hebbian connections with fixed weight distributions. It is assumed that the E-Puck already knows the concepts they represent. *Concept Nodes* make their content available to higher level cognitive processes, which are not dependent on specific detailed environmental representations along continuous feature dimensions but on more general properties of perceived objects and the environment.

The *Action Concept* nodes receive input from the perception *LED Nodes*. The fixed Hebbian connections between the *LED Nodes* and the *Action Concept* nodes connect three LEDs to a corresponding *Action Concept* node. The resting level of the action concept nodes is tuned such, that they will pass activation threshold when all three

LEDs associated wit the corresponding LED-Transient are active, which is also an indication of the CoS of that action. *Action Concept* nodes are connected to the short-term *Action Memory* nodes, which in turn projects its content to other parts of the architecture.

The *Sound Concept* nodes receive input from the *Sound Perception Field* and categorize perceived sounds into the concepts of "low-pitch", "medium-pitch" and "high-pitch" sounds. They are connected to the *Sound Transient Detection* nodes and to the *Goal Selection* and *Goal Match* representations which are part of the world-to-mind direction of fit.

Color Concept nodes categorize the color space into the six color concepts "red", "orange", "yellow", "green", "blue and "violet". The *Color Concept* nodes (see Figure 3.5) receive input from the *Color Attention Field* and represent the color concept of the currently attentionally selected object. As outgoing connections they project their content to *Color Transient Detection* and *Color Memory* nodes which are part of the mind-to-world DoF, (see Sections 3.5.4 and 3.5.5).

In addition, they determine the precondition of the *Orient to Target* intention and pass opportunistic activation to *Action Selection* nodes (which determine the content of the *Prior-Intention*). Opportunistic activation of *Action Selection* nodes is a shortcut that assumes a general knowledge about which actions can be performed on a perceived object. This assumption is made because the two representations are of different type with one representing perceived objects and the other being an action plan that may be seen as representing an action phrase. It is important to make this shortcut clear, as it is this connection that grounds the *Prior-Intention* representation. A more complete model could include a knowledge representation that first recalls the various actions possible on a perceived object, before passing opportunistic activation to a corresponding action representations. For brevity, the above mentioned shortcut was taken (see Section 3.6.7).

3.5.4 Transient Detection

Transient Detection nodes provide the architecture with the ability to detect transient feature changes in the environment. Transient detections are of particular importance for the architecture, since they represent the central mechanism with which the *Belief System* is addressed. Perceived transients can lead to belief formation episodes or can trigger belief recall episodes that test whether new perceptions are related to previously learned action-outcome contingencies. The dynamics of the architectures *Transient Detection* nodes correspond to that of the transient detection fields explained in Section 2.4.2[5].

The architecture perceives transients related to sounds, colors and currently selected goals. The *Sound Transient Detection* nodes perceive changes in the perceived sound pitch. As sound input is normalized and noise free in this scenario, change detections are limited to detecting newly perceived sounds. *Sound Transient Detection* nodes are connected to the *Outcome Detection* and the *Outcome Gate* nodes. In combination with the *Action Memory* nodes, a perceived sound may indicate that a just performed action caused an outcome, which initializes a belief formation episode (see Section 3.7.2).

The *Color Transient Detection* nodes perceive that a new object is attentionally selected. They typically detect a transient, when a new object enters the field of view. *Color Transient Detection* nodes are connected to the *Contingency Recall* intention, through which they can initiate a belief recall episode, which cues the *Belief System* in an attempt to recall a known contingency associated with the perceived object. A successfully recall attempt can lead to opportunistic activation of an associated goal (see Section 3.7.1).

The *Goal Transient Detection* nodes work analogously to the *Color Transient Detection* nodes and are connected to the *Goal Selection* nodes of the world-to-mind DoF (see Section 3.6.5). As such, they represent changes in the currently active goal state. The *Goal Transient Detection* nodes project their content to the *Strategy Recall* intention. This connection can lead to a belief recall episode, which cues the *Belief System* in an attempt to recall a known contingency related the newly formed goal state. Through this mechanism the architecture can reuse learned beliefs as strategies to guide outcome-oriented action (see Section 3.7.1).

3.5.5 Short-Term Memory

Salient outcomes generally occur only after an action has been performed. This is also the case in this scenario. Result sounds are played after the E-Puck has interacted with an object. Learning action-outcome contingencies thus involves associating past object interactions with current perceptions of outcome sounds.

Short-Term Memory Nodes give the architecture the ability to sustain past perceptual information for a short period of time, even if the corresponding percept has

[5] To keep figure 3.5 clear, only the *Transient Detection Nodes* are shown

since disappeared. They receive input from *Perception Nodes* and operate as short-term memory (see Section 2.4.3)[6].

Object interactions are characterized by the color of the target object and the performed LED-Transient. This information is stored in the *Action Memory* and *Color Memory* nodes. They project their content to the *Action Gate* and the *Color Gate* nodes. In combination with *Sound Gate* nodes they represent the content of a contingency for a belief formation episode (see Section 3.7.2).

The *Action Memory* nodes are connected to the CoS representations of *LED-Transients*, *Prior-Intention* and *Goal*, where an active memory node influences the CoS/CoD states of the different world-to-mind intentions (see Section 3.6.1).

3.6 World-to-Mind

3.6.1 Behavioral Organization

This section goes into detail about how the architectures world-to-mind DoF organizes the E-Pucks motor behavior to produce desired outcome sounds. Actuator control is limited to controlling the E-Pucks heading direction, the brakes and its LEDs. Figure 3.6 shows a schematic overview of the hierarchical organization of the world-to-mind intentions that enable the E-Puck to produce desired outcome sounds. The individual intentions are implemented as ECUs (depicted as red squares) and represent certain states of the environment the E-Puck wants to achieve. Higher level intentions achieve their desired world state by organizing lower level intentions into nested sequences which ultimately culminate in the sensorimotor surface and control motor primitives. They Organize lower level intentions by sequencing individual ECUs using *Precondition Nodes* (P) and by passing task input, that determines the content of lower level intentions.

Low-level motor behavior is goal-oriented, in the sense that the contents of intentions linked to the motor surface, and that of all higher intentions in the sequence, are ultimately determined by the highest level intention in the hierarchy. The highest active intention is the goal of the agent. In this scenario, the highest level intention (*Goal*) represents the desire of the E-Puck to produce certain sounds.[7] The

[6] To keep figure 3.5 clear, the inhibitory nodes are omitted

[7] In the learning phase of the experiment, when engaged in exploratory behavior, the *Goal* might not be active, at which point the *Prior-Intention* would be considered the current Goal of the Agent.

action sequence that enables the E-Puck to interact with objects is implemented by the *Prior-Intention*.

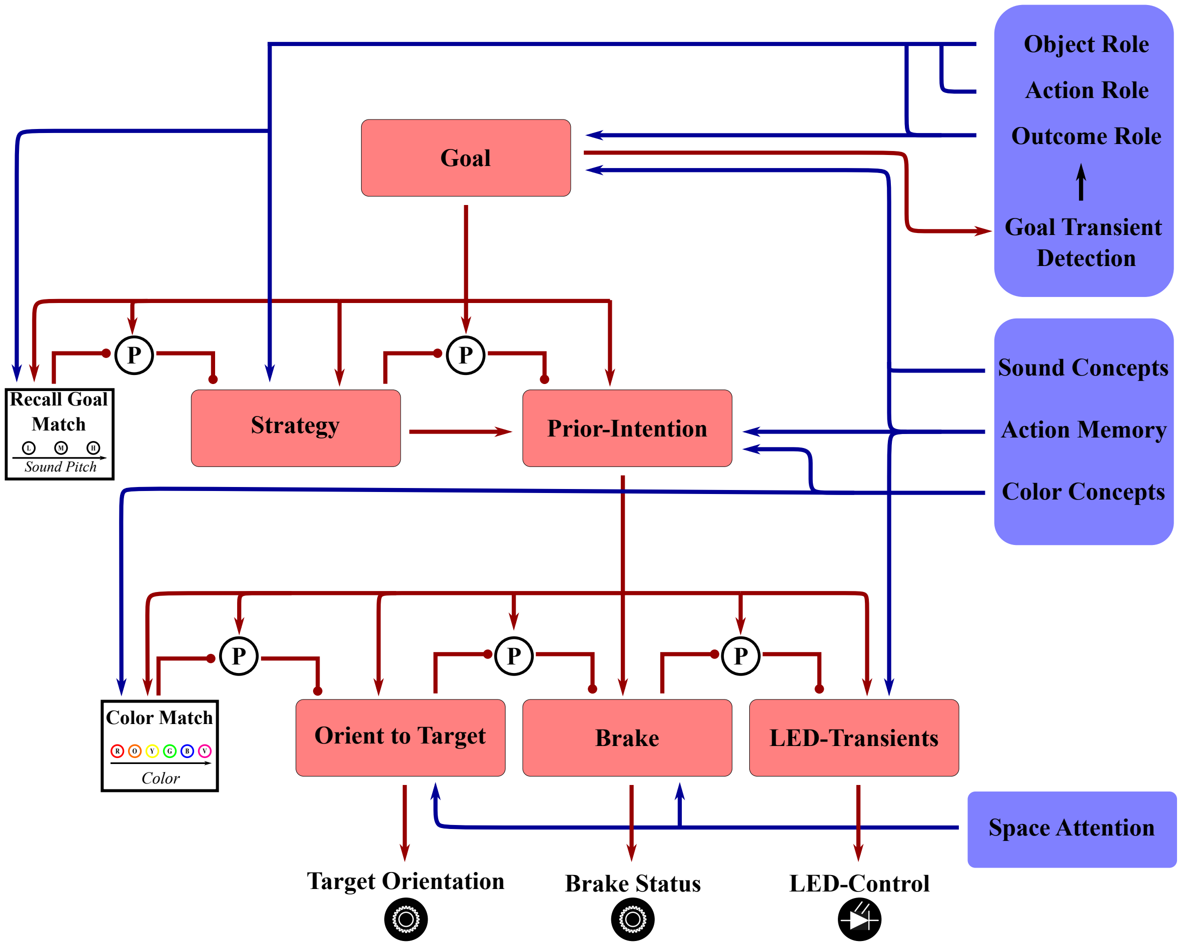

FIGURE 3.6: Overview of the sub-network organizing outcome directed action.

Lower-level intentions and motor primitives like *Orient to Target* or *LED-Transients* inherently achieve the desired world states they represent, since the ECUs that implement them are structured such that they automatically initiate appropiate actions (as defined by the architect of the model). This shortcut assumes that the agent robot has already learned action primitives, and knows how they influence the outcome.

This is not the case for the *Goal* representation. The agent robot has to first learn the contingency knowledge required to determine which actions lead to specific outcome sounds (see Section 3.7). A newly formed *Goal* state can determine the content of the lower-level *Strategy* and *Prior-Intention* representations by recalling the required knowledge from the *Belief* system. The activation pathway is depicted in the upper part of Figure 3.6. A newly formed *Goal* representation cues the *Belief System* through the *Goal Transient Detection* (see Section 3.7). The action-object matching the desired outcome-sound is then stored in the *Strategy* representation as an action plan. This is the main pathway through which the architecture uses contingency knowledge to produce outcome-oriented behavior.

The next sections give a more detailed account of how the individual intentions are implemented. Sections 3.6.2-3.6.4 cover the motor primitives *Orient to Target*, *Brake* and *LED-Transients*. Section 3.6.5 goes into detail about how the *Goal* representation is capable of recalling action strategies from the *Belief System* and how the *Strategy* representation is used to guide outcome-oriented behavior. Finally the Sections 3.6.6 and 3.6.7 go into detail about opportunistic activation and how the dynamics of the content representations at the level of *Prior-Intention* and *Goal* influence behavior. In particular, these sections discuss how memory traces that keep track of past behavior, organize exploration and lead to stabilization and habituation of goal states.

3.6.2 Orient to Target

The implementation of the sub-architecture that enables the E-Puck to to orient toward an object of a specific color is shown in Figure 3.7. The *Orient to Target* intention in action is implemented as an ECU. The target color is specified by the *Color Match* nodes which detects that an attentionally selected object matches the target color.

The ECU implementing the *Orient to Target* intention is activated as part of the *Prior-Intention*. The target color is specified by the *Color Match* nodes. These matching nodes represent that an attentionally selected object matches the target color specified by *Action Selection* nodes of the *Prior-Intention*. The *Orient to Target* intention in action is activated when this precondition is met.

The *Intention Node* passes a homogeneous boost to the *Motor Gate Field*. The *Motor Gate Field* represents the horizontal position of the target object. It is coupled to the *Space Attention Field* and forms a corresponding peak in conjunction with the boost of the *Intention Node*. The peak position of the *Motor Gate Field* is passed on to Webots through the sensorimotor surface. This peak position corresponds to the horizontal position of the target object. In Webots the horizontal peak position determines the target orientation of the attractor dynamics (see Section 3.3.1).

The peak position is determined by calculating the center of activation in the *Motor Gate Field*. The center of activation is equal to the current peak position of the

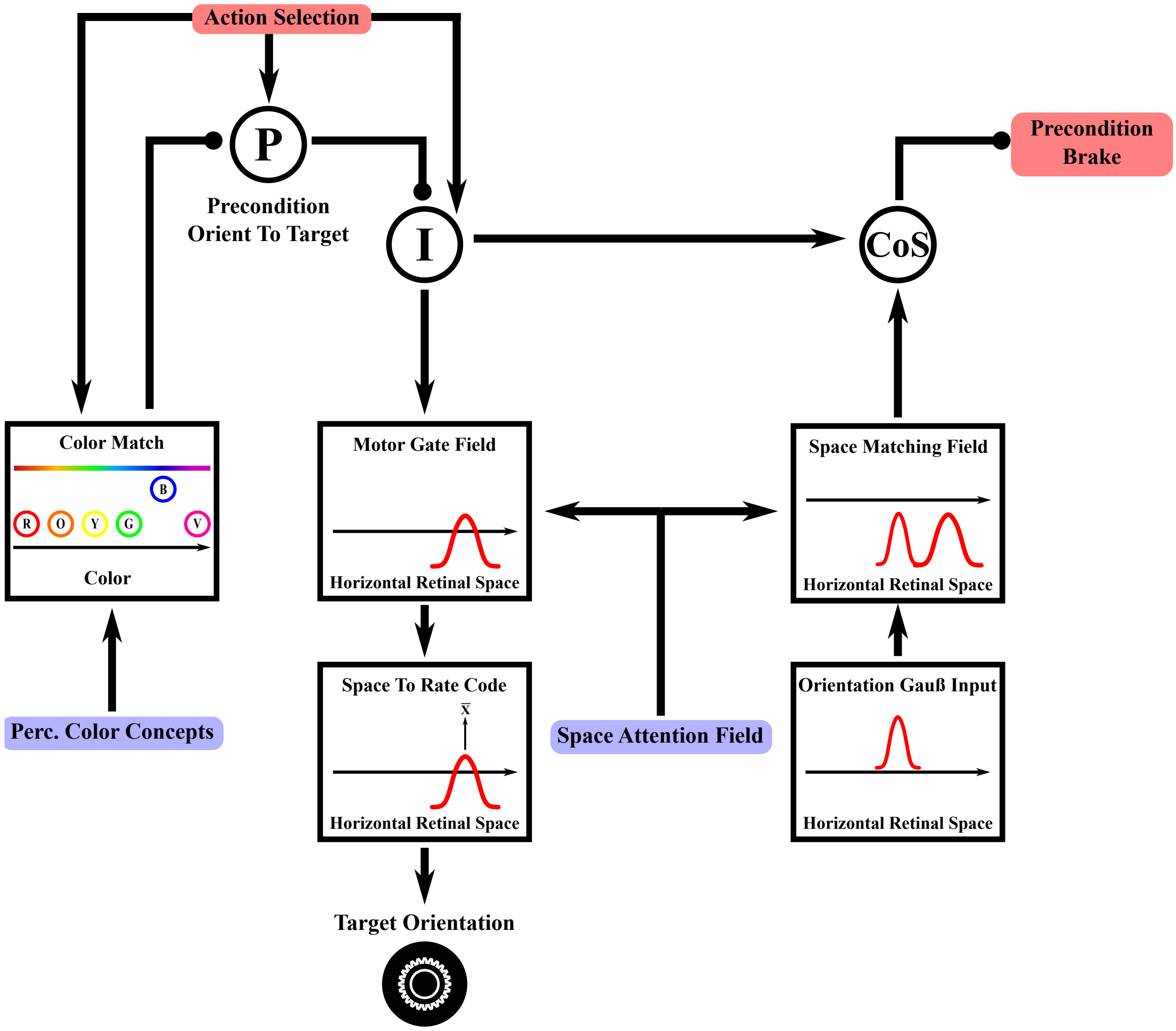

FIGURE 3.7: Schematic of the Orient to Target intention. The intention is initiated when a object matching the target color is attentionally selected.

Motor Gate Field. This operation transforms the representation of the targets position from a space-code to rate-code (see *Space To Rate Code* of Figure 3.7).

The *Orient to Target* CoS is fulfilled when the E-Puck is oriented toward the target object, i.e. when the object is in the center of the field of view. This condition is detected by the *Space Matching Field*, which operates as a matching field. It receives input from the *Space Attention Field* and from a Gaussian preshape (*Orientation Gauß Input*). The Gaussian preshape is positioned in the center of the horizontal field of view and represents the desired orientation. The *CoS Node* inhibits the connected precondition node of the *Brake* intention in action.

3.6.3 Brake

The *Brake* intention is implemented as an ECU and shown in Figure 3.8. It allows the E-puck to stop in front of a selected target object. The *Brake* intention is part of the action sequence specified by the *Prior-Intention*. The intention in action described here is the second action in this action sequence. The precondition for its activation is the *CoS* of the *Orient to Target* intention and the detection that a target object is right in front of the E-Puck.

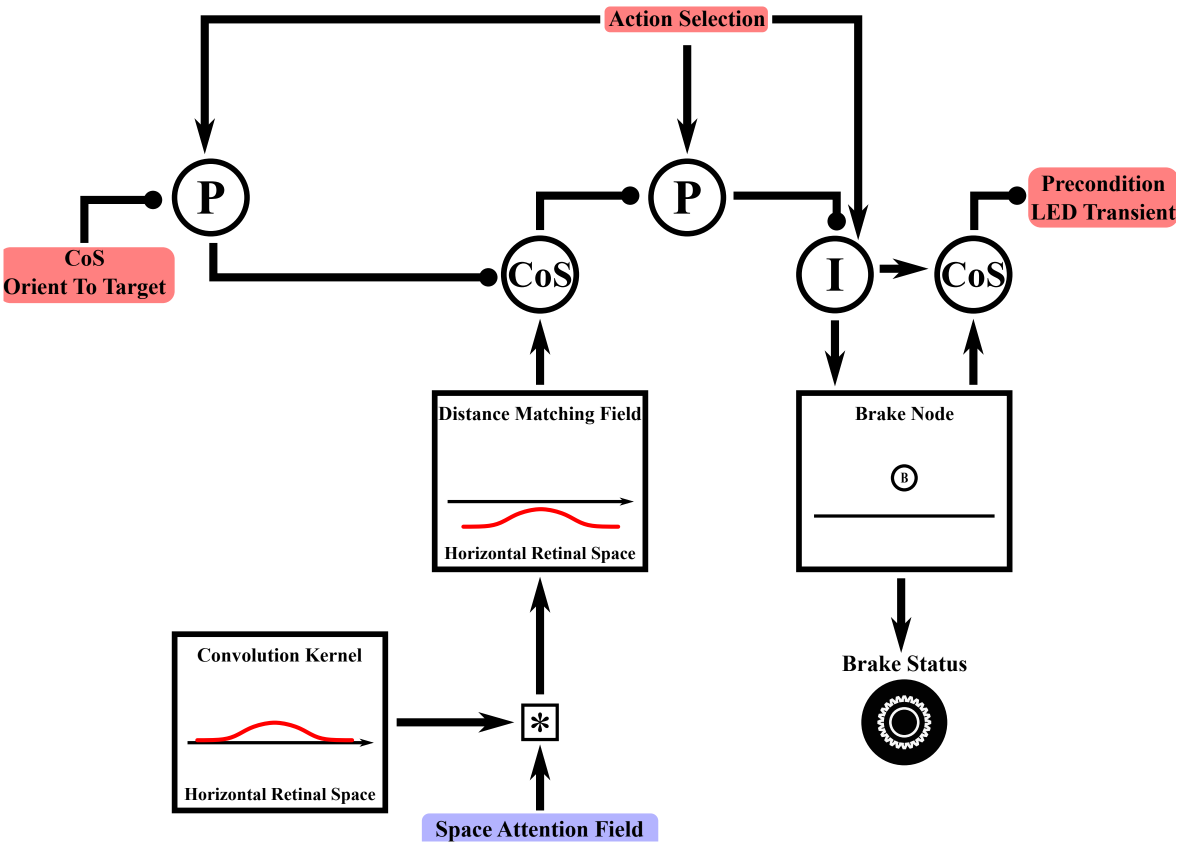

FIGURE 3.8: Schematic of the Brake intention in action. The brake node activates when the E-Puck is close to the target object, as measured by the horizontal radians of the target object on the retinal image.

The *Precondition Node* receives inhibitory input from the *CoS Node* of the *Orient to Target* intention and from the *Distance Matching Field*. The *Distance Matching Field* detects if an object in focus is close in front of the E-Puck. This is determined using the radial extent of the object in the field of view. As a measure of radial extent, the connection kernel transforming the input from the *Space Attention Field* is given the shape of a flat Gaussian curve that occupies the entire width of the field of view (see *Convolution Kernel* in Figure 3.8). The wider the target object becomes in the field of view, the greater the overlap between kernel and the input from the *Space Attention Field*. This is reflected in the integral of the convolution, whereby the output amplitude increases the closer the object is. The convolution therefore serves as a metric to determine the distance of an object.

The E-Puck stops as soon as the *Brake Node* is activated. It is connected to the sensorimotor surface and its activation state controls the the brake status in Webots. As a shortcut the *Brake Node*, constitutes the CoS of the *Brake Intention*. The *CoS Node* inhibits the precondition node of the *LED-Transients* which are the next part of the *Prior-Intention* sequence.

3.6.4 LED-Transients

The *LED-Transient* intention in actions shown in Figure 3.9 implement chained LED sequences, that connect three LEDs in series. The architecture features three of such sequences, each one controlling three different LEDs. These intention in actions constitute the final motor primitive of the action sequence controlled by the *Prior-Intention* and it is these transients that differentiate the possible object interactions the E-Puck can perform (see Section 3.6.5).

The three *LED-Transient* intentions are implemented as ECUs and are activated as part of the *Prior-Intention*. The *Precondition Node* ensures that *LED Transients* are not initiated before the E-Puck stands in front of the target object. The *Action Selection* nodes, select the LED-Transient to perform. LEDs are each represented by an *LED Node*. The *LED Nodes* are connected to the sensorimotor surface and control the status of the LEDs. When an *LED Node* is activated, the corresponding LED of the E-Puck is switched on. For each transient, three of the nine *LED Nodes* making up a transient are connected in series, with LEDs *1*, *4* and *7* each forming the first links in a transient (see Figure 3.9). The *LED Nodes* operate on a slow time scale (in the order of seconds). For example, when *Intention Node 1* is activated, LEDs *1*, *2* and *3* will light up one after the other in a few seconds interval.

The *CoS Nodes* of the *LED-Transients* are connected to the *Action Memory* nodes of the mind-to-world DoF. They activate, when the architecture perceives via the sensorimotor surface that a transient has been completed, i.e. that all LEDs making up that transient have been activated. *CoS Nodes* remain active as long as the short-term memory representation sustains that perception. The *CoS Nodes* of the *LED-Transients* also constitute the completion of the *Prior-Intention*. The *Action Memory* nodes are used as a indication of the CoS, as they sustain the CoS for the period of

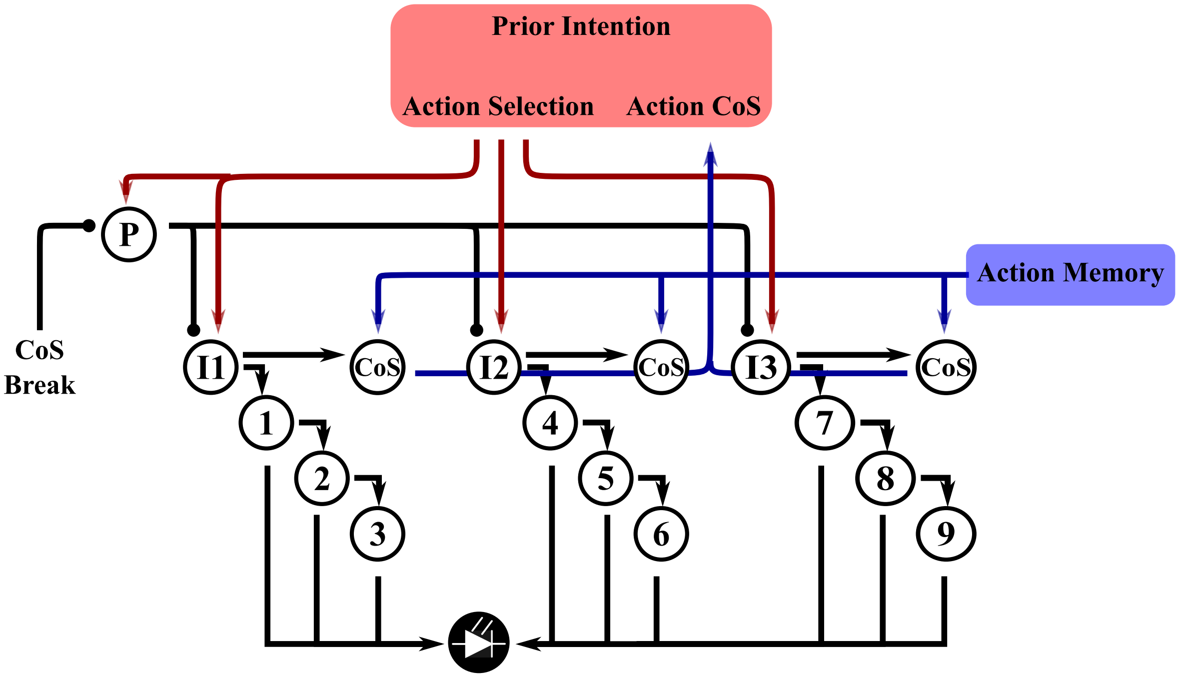

FIGURE 3.9: Schematic of the 3 LED intention in actions. Each action consists of 3 LED nodes chained together in a sequence. LED nodes are connected to the sensorimotor surface and can can be seen as placeholders for more concrete motor actions.

time, in which the memory trace build up in the *Action Selection* nodes takes place (see Section 3.6.7).

3.6.5 Goal-Oriented Behavior

A detailed depiction of the ECUs implementing the higher level intentions *Prior-Intention*, *Strategy* and *Goal* is shown in Figure 3.10.

The content of the ECU implementing the *Goal* level is determined by a set of *Goal Selection* nodes. The *Goal Selection* nodes represent the "pool" of possible goals related to the scenario. In particular they represent the desire of the E-Puck to produce either a "low-", "medium-", or "high-pitch" sound. *Goal Selection* nodes feature strong mutual inhibition, putting them in the selective regime. The selective dynamics leads to competitive selection of goals, which ensures that at most one goal is selected at a time. They receive input from the *Supervisor Command*[8] and *Sound Perception Concept* nodes, as well as from the *Outcome Role* nodes of the *Belief System*. Competing input from different sources (e.g. opportunistic input from the

[8]symbolized by the receiver symbol at the *Goal* level in Figure 3.10

Outcome Role and *Supervisor* nodes) leads to competitive selection decisions in the *Goal Selection* nodes. This can be seen as an implementation of a neural dynamics of competing goals, that models how goals are selected and maintained, depending on the state of the remaining architecture. The dynamics of *Goal Selection* nodes and the role of opportunistic activation and *Supervisor Commands* in goal selection are further elaborated in Section 3.6.6.

The *Goal Selection* nodes are connected to the motor surface and project their content to the emitter of the E-Puck. The emitter continually reports the current goal of the architecture to the supervisor. This allows the supervisor to purposefully place target and distractor objects in the environment (see Section 3.3.5).

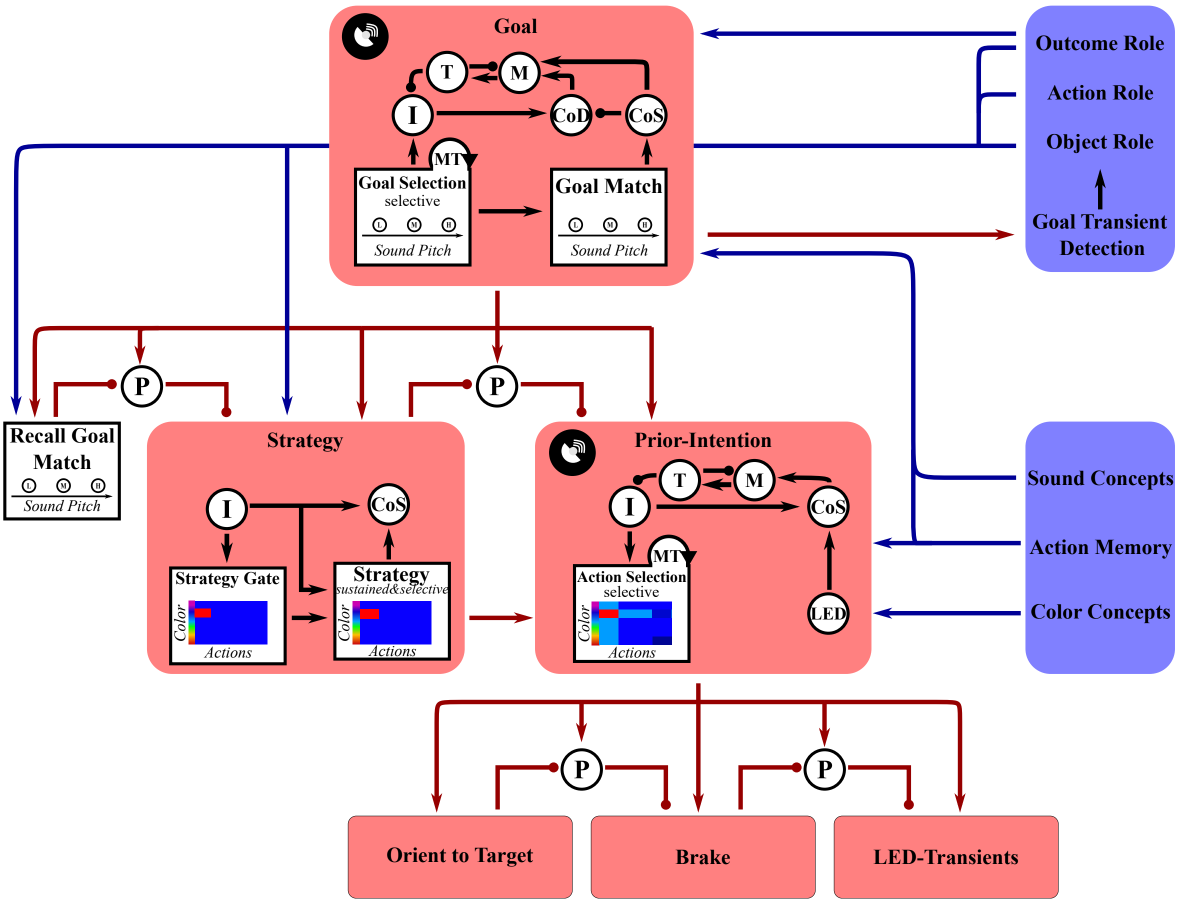

FIGURE 3.10: Schematic of the subnetwork organizing outcome directed action. An active goal states projects to the action selection through a strategy representation. The action selection nodes represent prior intentions and activate their associated action sequence.

The *Intention Node* at the *Goal* level is a peak detector and activates when a goal was selected in the *Goal Selection* nodes. The *Intention Node* organizes behavior by boosting the different *Precondition* and *Intention Nodes* of *Strategy* and *Prior-Intention*.

The *Goal Match* nodes receive input from *Goal Selection* and *Sound Concept* nodes. *Goal Match* nodes detect if a perceived sound matches the desired outcome sound. The *CoS Node* at the goal level is connected to the *Goal Match* and *Action Memory* nodes and activates when a desired sound is heard right after the agent has performed an LED-Transient. The condition of dissatisfaction (*CoD*) is detected by the *CoD Node* which activates after an action was performed (through its connection with the *Action Memory* nodes). The *CoD Node* is receives inhibitory input from the *CoS Node* and operates on a slower timescale. It thus only activates if no matching sound is heard in the period after an action was performed.

Both the *CoS* and the *CoD Nodes* feature a memory representation (M) (see Section 2.4.5). The memory representations are connected to a transient detector (T) (see Section 2.4.2). The transient detector operates on a slower timescale and inhibits the *Intention Node* and the *CoS/CoD* memory (M), which resets the state of the *Goal* ECU. The transient detector is tuned such, that it activates after the *CoS/CoD Nodes* have since decayed. This can be seen as a timer that blocks inhibition for a while, thus delaying the inhibition of the *Goal Selection* nodes. This delay exists to allow for a normalized buildup of the memory traces in *Goal Selection* nodes. The buildup of memory traces in the *Goal Selection* nodes is coupled to the state of *CoS/CoD Nodes*. If the *Goal Selection* nodes were to be inhibited directly through the *CoS/CoD Nodes*, no buildup of activation memory traces would occur.

The *Strategy* ECU controls the working memory representation of recalled action-object pairs that function as action plans. The *Recall Goal Match* nodes, are connected to the *Goal Selection* and the *Outcome Role* nodes. They become active, when the current goal matches the outcome representation of a recalled contingency. When this is the case, the *Intention Node* activates the *Strategy Gate* nodes. The *Strategy Gate* nodes are connected to the recalled *Action Role* and *Object Role* nodes and form a bound action-color representation. The *Strategy* nodes represent an action plan to reach the currently desired goal. They operate in a selective and self-sustained regime, when boosted by the *Intention Node* at the *Goal* level. *Strategy* nodes represent the latest recalled strategy. Input from the *Strategy Gate* nodes destabilizes the memory of the previous strategy and activates. An active *Strategy* node preshapes the *Action Selection* nodes of the *Prior-Intention* and also acts as the *CoS* of the *Strategy* ECU.

The *Prior-Intention* organizes the sequence of motor primitives necessary to execute LED-Transients. The *Prior-Intention* activates either as a result of input from

Supervisor Commands[9] or as part of the organization of outcome-oriented behavior controlled at the *Goal* level. When activates as part of a *Goal* The *Precondition Node* makes sure that goal-oriented action is only initialized after an action plan has been recalled. The *Intention Node* provides a homogeneous boost to the *Action Selection* nodes and boosts its *CoS Node*.

The *Action Selection* nodes determine the content of the *Prior-Intention* and represent all possible action-object combinations. Like *Goal Selection* nodes, they operate in a selective regime. *Action Selection* nodes receive ridge input from both *Color Concept* and *Action Memory* nodes and additional top-down input from the *Strategy* representation. In general, an action sequence is initialized as soon as a selection decision is made in the *Action Selection* nodes. Their dynamics and the role of strategies, object perceptions and *Supervisor Commands* are further elaborated in Section 3.6.7.

The condition of satisfaction of the *Prior-Intention* coincides with the CoS of the corresponding LED-Transient. The *CoS Node* features a memory representation (M), that is connected to a transient detector (T). This setup is analogous to that at the goal level and allows for a normalized buildup of the memory trace in the action selection nodes (see above).

In summary, the overall process of behavioral organization that enables goal-oriented action can be summarized as follows: Whenever the current goal changes (e.g. as a result of *Supervisor Commands*, or opportunistic activation), the architecture attempts to recall a strategy from the *Belief System* that fits the new goal. The precondition of the *Strategy* ECU is fulfilled, if a contingency representation in the *Belief System* matches the current goal. The recalled strategy is then stored in the *Strategy* nodes and projected on to the *Action Selection* nodes. *Action Selection* nodes are grounded in perceptual input and initiate action sequences based on strategy input and current perceptual input (see Section 3.6.7). After each goal reaching attempt the *CoS* and *CoD Nodes* at the goal and action level lead to memory-buildup of memory traces. After memory-buildup, *Goal* and *Action Selection* nodes get inhibited by the transient detections (T), which deactivates all associated intentions. This resets the state of the world-to-mind direction of fit, while memory traces store information of past goal reaching episodes.

[9]symbolized by the receiver symbol at the *Prior-Intention* level in Figure 3.10

3.6.6 Goal Dynamics

The selection decisions of *Goal Selection* nodes are decisive for the behavior of the E-Puck. When active, the E-Puck acts outcome-oriented by trying to produce the desired sound according to the content of the *Goal Selection* nodes. The dynamics governing the selection decisions of *Goal Selection* nodes are given by:

$$\begin{aligned} \dot{u}_{GS_i} = &- u_{GS_i} + h \overbrace{+ k g(u_{GS_i}) - \sum_{j=1}^{3} c_{inh} g(u_{GS_j})}^{\text{selective coupling}} \underbrace{- c_- m_{i-} + c_+ m_{i+}}_{\text{memory traces}} \\ &\underbrace{+ \sum_j c_{tsk_j} g(u_{tsk_j}) + c_{opp} g(u_{role})}_{\text{external input}} . \end{aligned} \tag{3.4}$$

The selective coupling terms implement strong global inhibition and self-excitation of *Goal Selection* nodes, which puts them in a selective regime. The resting level of individual *Goal Selection* nodes is modulated by a pair of memory traces m_+ and m_-. These memory traces develop according to equation 2.19, when a *CoS/CoD* node is active. Their influence on selection decisions is discussed further below.

In general, a selection decision is induced by external input. This external input can either be a task signal from the *Supervisor Command* nodes u_{tsk} or opportunistic input from the *Role Outcome* nodes u_{role}. When an object comes into the field of view of the E-Puck, the *Strategy Recall* intention attempts to recall a matching contingency (see Section 3.7.1). If a matching belief has been learned already, the *Outcome Role* nodes reflect which sounds can be generated with the perceived object (see Section 3.7.1). Thus, *Goal Selection* nodes are only activated opportunistically, if an associated contingency was learned in the past. This is in contrast to opportunistic activation at the level of *Prior-Intentions*, which induces any of the possible actions that could be performed with the perceived object (see Section 3.6.7).

The *Goal Selection* nodes are tuned such that opportunistic input alone is just sufficient to reach detection threshold. The *Goal* is then active as long as it is stabilized by the *Outcome Role* nodes. However, the dynamics of the *Goal Selection* nodes also depend strongly on the input of the *Supervisor Command* nodes. These nodes represent tasks given to the E-Puck in different experiment phases. Depending on the phase, they either activate specific goal sounds (Test Phases), or pass a homogeneous boost to the *Goal Selection* nodes (selection/Stability Phases).

In the three test phases, *Supervisor Command* nodes activate one of the three *Goal Selection Nodes*. The boost is strong enough stabilize the new goal against opportunistic input. In this way the supervisor can specify certain target sounds to the E-Puck, which is used in the experiment to test outcome-oriented behavior (see Section 3.2.2). In the Selection and Stability phases, the *Supervisor Command* nodes give a homogeneous boost to all *Goal Selection* nodes. The boost is just sufficient to cause a selection decision. This way, the supervisor encourages the E-Puck to select a goal sound. The selected goal sound is stabilized by this selection boost and is tuned such that it remains stable against opportunistic input. By encouraging selection decision it is possible to measure selection and stability dynamics of the goal representation.

The *Excitatory Memory Trace* m_+ provides positive input and operates on a faster timescale than the *Inhibitory Memory Trace* m_-. Both traces track the history of goal reaching attempts. Activation is build up after each goal reaching attempt, and is controlled by the *CoS/CoD Nodes*. The *Excitatory Memory Trace* is connected to the *CoS Node*, and builds up activation every time a goal sound was successfully played. The *Inhibitory Memory Trace* is connected to both the *CoS Node* and the *CoD Node*, thus also building up activation when no sound was produced. Together these two memory traces modulate the resting level of the *Goal Selection* nodes. Successful goal reaching episodes temporarily increase the resting level of *Goal Selection* nodes, as the faster positive memory trace builds up. The slower inhibitory trace starts to build up after multiple goal reaching episodes, which eventually leads to a decrease in the resting level. In learning phase, opportunistic activation is then not sufficient to induce a goal, which leads to autonomous exploration. In Selection and Stability phases, inhibition and opportunistic input can then be enough to destabilize active goals in favor of an opportunistically chosen new goal, which can be seen as exploitative behavior.

3.6.7 Action Dynamics

Selection decisions in the *Action Selection* nodes are also important for the behavior of the E-Puck. When active, the E-Puck initiates action sequences consisting of the *Orient To Target*, *Drive to Target* and *LED-Transient intentions*. The dynamics of *Action*

Selection nodes is given by:

$$
\begin{aligned}
\dot{u}_{AS_i} =& - u_{AS_i} + h + c_{int} g(u_{int}) + \overbrace{+kg(u_{AS_i}) - \sum_{j=1}^{N} c_{inh} g(u_{AS_j})}^{\text{selective coupling}} \underbrace{-c_{-} m_{i-}}_{\text{memory trace}} \\
& \underbrace{+c_{exp} g(u_{exp}) + c_{opp} g(u_{Color}) + c_{str} g(u_{str})}_{\text{external input}} .
\end{aligned}
\tag{3.5}
$$

The selective coupling terms put the *Action Selection* nodes into a selective regime. Similar to the *Goal Selection* nodes, the resting level of the *Action Selection* nodes is modulated by an inhibitory memory trace m_{-}. This memory trace develops according to equation 2.19, whenever the *CoS* of the *Prior-Intention* is active.

The *Action Selection* nodes receives external input from the *Strategy* nodes u_{str} and the *Explore Node* of the *Supervisor Commands* u_{exp}. In addition, the *Intention Node* u_{int} of the *Prior-Intention* ECU passes a homogeneous boost. This boost signals a general readiness to act, by lifting the *Action Selection* nodes into a regime in which selection decisions can be triggered. The *Color Concept* nodes u_{Color} project opportunistic ridge input for a perceived color. This opportunistic input is a short cut that assumes a general knowledge of what actions are possible on perceived objects. A more complex model would include knowledge representations that tell the agent robot, what actions are meaningful (e.g. for colored buttons, this general knowledge would include a representation that buttons can be pushed).

The *Action Selection* nodes are tuned to activate when opportunistic and other external input overlap. For example, an action is initiated when input from *Strategy* nodes and opportunistic ridge input overlap (when the perceived object color matches the strategy). In the learning phase, the E-puck receives the task of trying out actions in the environment. This task is represented by the *Exploration Node* of the *Supervisor Commands*. The *Exploration Node* activates the *Prior-Intention* and gives a homogeneous boost to the *Action Selection* nodes. Opportunistic input from the *Color Concept* nodes is then sufficient to cause opportunistic activation. Since selective, an action is randomly chosen to be executed on the perceived object.

The resting level of *Action Selection* nodes is modulated by an inhibitory memory trace. Build up activation in this memory trace passes a small inhibition to the *Action Selection* nodes which gives non-inhibited nodes a small advantage when competing for activation. In learning phase, this leads to an exploration of the action-object space as new actions are preferred over repeating already performed actions. The

Action Selection nodes have a bound color-action representation, so that the buildup of memory trace results in inhibition of a unique action-object pair.

3.7 Belief System

To learn contingencies, the *Belief System* forms associations between perceived outcome sounds and performed actions. The *Belief System* implemented in this thesis works analogously to that described in Section 2.5.3. In this scenario, actions are represented by the color of the target object and the concept representation of the performed LED-Transient. Outcomes are represented by the sound pitch of the perceived outcome sound (see Figure 3.11).

The *Role Concept Nodes* implementing these contingency representations operate in a selective regime and are tuned such that an active *Belief Node* is sufficient to activate the corresponding *Role Concept Nodes*. The mutual activation of *Role Concept Nodes* and the *Belief Node* leads to a self sustained belief representation. After a belief has been activated, it stays active until additional input to the *Role Concept Nodes* destabilizes it.

Active beliefs influence the rest of the architecture through the connections emanating from the *Role Concept Nodes*. The *Outcome Role* nodes are connected to the *Goal Selection* nodes. Via this pathway, a recalled belief can lead to opportunistic activation of goals. *Action Role* and *Object Role* nodes are connected to *Strategy Gate* nodes. Through this connection, the architecture is able to form an action *Strategy*, if the *Outcome Role* node of a currently active belief matches the current goal (see Section 3.6.5).

The *Role Concept Nodes* receive input from the rest of the architecture through *Gating Nodes*. These *Gating Nodes* can be seen as an interface to the remaining architecture. They control the incoming flow of activation to the *Role Concept Nodes*. The *Recall Intentions* and *Outcome Detection* can enable incoming activation by boosting certain *Gating nodes* (see bottom part of Figure 3.11). In that way, they define which cognitive events can lead to recall attempts or learning episodes.

3.7.1 Belief Recall

The *Recall Strategy* intention and the *Recall Contingency* intention attempt to reactivate learned beliefs in the *Belief System*, so as to make their content available to other parts of the architecture. Unlike the ECUs shown so far, *Recall Intentions* are

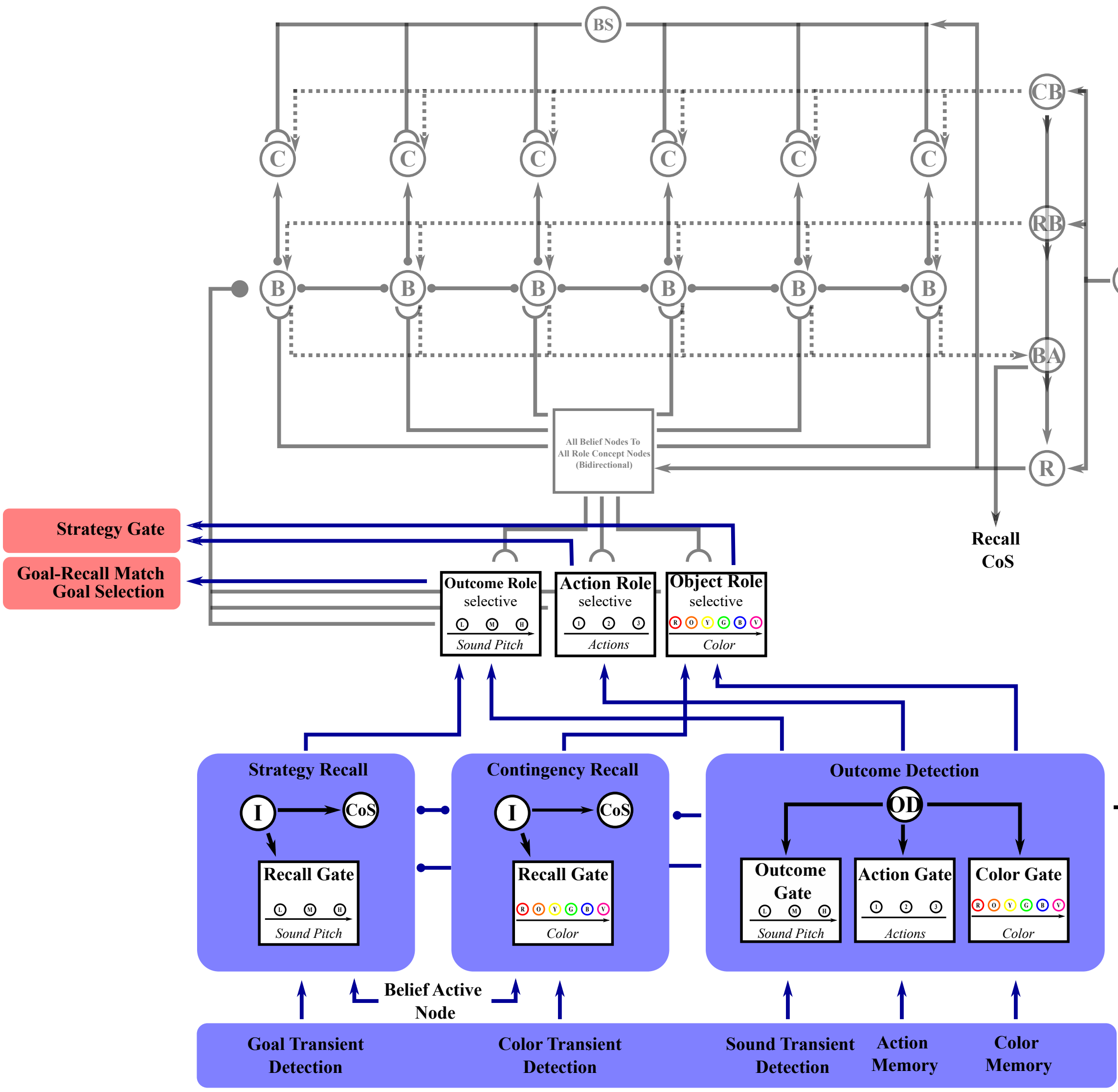

FIGURE 3.11: Schematic of the belief architecture. Beliefs are represented by belief nodes, which come to stand for action-object-outcome triplets through Hebbian learning of their synaptic connection to role concept nodes. Recalled beliefs are stored in role nodes. These nodes project the contents of the last recalled belief to the rest of the architecture.

not directly chained into action sequences. Instead they operate through mind-to-world perceptual input. In particular, the *Recall Strategy* intention is connected to the *Goal Transient Detection* and the *Recall Contingency* intention is connected to the *Color Transient Detection* (see bottom left part of Figure 3.11).

Together, the *Recall Intentions* define which cognitive events trigger the recall of a previously learned belief. In general, a *Recall Intention* is triggered when a new object moves into the field of view, or a new goal is selected. A new object that moves into the field of view is detected by the *Color Transient Detection* nodes, while the formation of a new goal is detected by the *Goal Transient Detection*.

In the former case, the perception of a colored object can lead to the opportunistic recall of a contingency that involves the perceived color. As the *Outcome Role* nodes project their activation to the *Goal Selection* nodes, this opportunistic recall can lead to a selection decision in the *Goal Selection* nodes and thus the opportunistic formation of a new goal (see Section 3.6.6).

If the recall is initiated by the formation of a new goal, the recall attempt can lead to the recall of a contingency representation that matches the desired outcome sound. The *Action* and *Object Role* nodes then represent an action-object combination that has previously lead to the desired sound. The architecture then stores this action-object combination in the *Strategy Nodes* which represents an action-plan and can drive outcome-oriented action (see Section 3.6.5).

The two *Recall Intentions* are structured analogously. They are coupled to inhibit each other and receive additional inhibitory input from the *Outcome Detection*. These connections ensure that at most one *Recall Intention* is active at any given time. The inhibitory connection from the *Outcome Detection* node gives priority to the formation of new beliefs. A *Recall Intention* stays active for as long as the corresponding *Transient Detection* is active. During this time, the intention tries to activate a matching belief in the *Belief System*. This is done by boosting the corresponding *Recall Gate* nodes, which then pass activation of the *Transient Detection* node to the *Role Concept Nodes*.

Input to the *Role Concept Nodes* is then sufficient to lead to supra-threshold activation. If the activated *Role Concept Node* has been associated with a belief in the past, it will activate the corresponding *Belief Node* through the reciprocal Hebbian connections. The active *Belief Node* will then in turn activate the *Concept Nodes* in the remaining two roles, thus forming the complete action-object-outcome triplet representing the contingency. If no belief has been associated with the recall cue in the past, the cued *Role Concept Node* will not activate a belief, as the weight distribution of the Hebbian connections do not form a connection to a *Belief Node*.

If a belief is already represented in the *Role Concept Nodes*, the existing belief representation is first deactivated by the recall cue. This happens generically by the cue from the *Recall Gate Nodes*. The selective *Role Concept Nodes* are tuned such that the input of *Gate Nodes* destabilizes already active nodes. In this case, the content of the *Role Concept Nodes* no longer matches the active *Belief Node*, which disables it because of the inhibitory input of the *Role Concept Nodes* to the *Belief Nodes* (top left of Figure 3.11). As a result, the other two *Role Concept Nodes* also fall below the detection threshold. When this happens, only the cued *Role Concept Node* remains active. This node can then activate a new *Belief Node* as described above.

The *CoS Nodes* represent the success of a recall attempt. The *CoS Node* receives input from the *Intention Node* and can be activated by the *Belief Active Node* of the *Belief System*.

3.7.2 Belief Formation

Learning a contingency consists of forming an association between a *Belief Node* and an active contingency representation in the *Action Role*, *Object Role* and *Outcome Role Nodes*. Associations are formed by Hebbian learning. The *Outcome Detection Node* initiates learning episodes and is connected to the *Sound Transient Detection* nodes and the *Action Memory* nodes (see bottom right side of Figure 3.11).

The short-term memory representation of the *Action Memory* nodes defines a time window in which a perceived sound is recognized as being the result of a just performed action. In DFT terms, the *Outcome Detection Node* acts as a peak detector that only activates when receiving input from both the *Sound Detection* nodes and the *Action Memory* nodes. This can be seen as a form of agency detection. Only those sounds perceived right after an action was performed are considered to be caused by the E-Puck.

The *Outcome Detection Node* inhibits the two *Recall Intentions* to prevent spurious recall attempts while learning is taking place. This is important, as the E-Puck may attentionally select a new object in its field of view, before the learning episode of a previous action has finished. Spurious input form the *Color Transient Detection* nodes would then activate a second *Object Role* node, leading to an ambiguous contingency representation.

To organize learning, the *Outcome Detection Node* provides a homogeneous boost to the *Outcome Gate*, *Action Gate* and *Color Gate* nodes. The *Color Gate* and *Action Gate* nodes are connected to the corresponding short-term memory representations. The *Outcome Gate* nodes are connected to the *Sound Change Detection* nodes. Together these gating nodes form an action-object-outcome triplet that define the contingency. When boosted, the *Gate Nodes* pass their content on to the *Role Concept Nodes* (see Figure 3.11).

Input from the *Gate Nodes* is strong enough to deactivate already active beliefs. Thus, when the *Outcome Detection Node* becomes active, the *Role Concept Nodes* come to reflect the content of the contingency representation in the *Gate Nodes*.

To initiate learning the *Outcome Detection Node* activates the *Learning Node* of the *Belief System* (see right side of Figure 3.11). The belief formation process is described in Section 2.5.3. If the contingency represented in the *Role Concept Nodes* has not been learned in the past, it becomes associated to a new *Belief Node* via Hebbian learning. The learning rate is tuned such, that contingencies are learned during one belief formation episode (one-shot learning). The learned belief can then be recalled by a *Recall Intention* which may then enable outcome-oriented action.

Chapter 4

Results

The previous chapter described the developed architecture, designed to provide a grounded account of Ideomotor Theory. This included the different sub-architectures enabling the E-Puck to act and solve tasks given by the supervisor, the mechanisms organizing outcome oriented behavior as well as opportunistic activation and how dynamics of memory traces at goal- and action-level modulate the dynamics of selection decisions.

The previously described capabilities of the architecture are now shown by plotting the recorded time courses and activation snapshots of neural fields and nodes comprising the architecture. A full depiction of all neural fields and nodes is omitted however, as most significant events are only caused by varying subsets of relevant fields and nodes. Instead this chapter demonstrates individual features of the architectures behavior by selectively plotting the activation of relevant fields in time windows that are exemplary for the behavior to be shown.

In Section 4.1, the general behavior of the architecture is demonstrated. This includes how it is able to navigate the environment, select target objects and engage in object oriented action. In addition, this section demonstrates how the architecture forms beliefs about observed action-outcome contingencies, and how it can use these beliefs to engage in outcome-oriented action and opportunistically induce goals. The test phases demonstrate this behavior as described in Section 3.2.2.

The next sections (4.2, 4.3, 4.4) go through the other experiment phases. To demonstrate the behavior in each phase, these sections first provide exemplary activation time courses of individual action episodes, that show how individual action decisions come about. After that, an overview of the performed actions and goal decisions is shown for the given experiment phase and related back to the behavior at individual action episodes. In this way these sections show how memory traces at action and goal level modulate individual action and goal decisions, which when viewed over multiple action episodes give rise to early familiarity preference and later novelty preference behavior proposed in chapter 1.

4.1 General Behavior

This section demonstrates the general behavior of the architecture. The aim is to show how the architecture is able to act in the environment, and how it can perform object oriented actions. In addition this sections demonstrates the capability of the architecture to form beliefs about perceived action-outcome contingencies, and how these beliefs can be used to engage in outcome oriented action, as well as how activation of these beliefs can lead to opportunistic activation of goal states.

4.1.1 Object Selection

The actions available to the E-Puck are object oriented. Outside of the general obstacle avoidance behavior, intention in actions are always related to a target object. Thus, to interact with an object the E-Puck must first attentionally select a target object. The sub-architecture responsible for this is shown in Section 3.5.

Figure 4.1 shows an example of how the architecture selects a single object in the visual field of the camera. For this purpose, snapshots of the camera image and relevant fields are shown at three different points in time. The bottom row shows camera images. There are three objects in the visual field of the E-Puck (yellow, green and red).

In the two rows above are activation snapshots of the two-dimensional *Color/Space Perception* and *Color/Space Attention* fields. The horizontal dimension corresponds to the horizontal position of the objects in the camera image. The vertical dimension corresponds to the color value of the object. The *Color/Space Perception* fields amounts to the visual scene representation of the architecture.

Snapshots of the one-dimensional *Color* and *Space Attention Fields*, which form a unique representation of a selected object, are shown in the two rows above. In addition to the activation of the *Color Attention Field*, the time course of the *Color Concept* nodes is shown, which shows how color concepts are extracted from continuous hue values.

Attentional selection of objects occurs by first selecting a target color, and then extracting the position of an object of the corresponding color from the *Color/Space Perception Field*. The first snapshot of the *Color/Space Perception Field*, shows how the

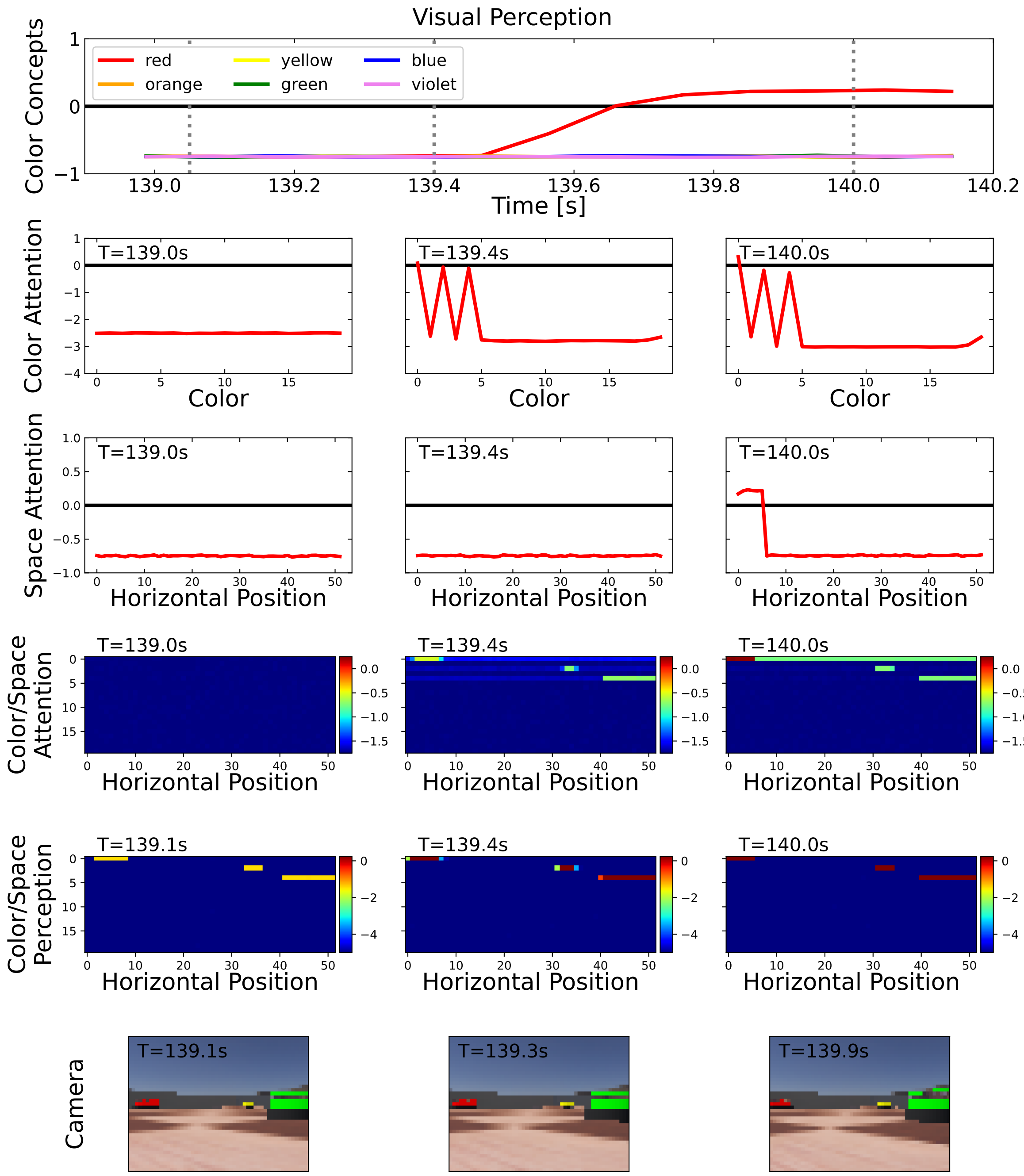

FIGURE 4.1: Activation time courses and snapshots of selected neural fields and nodes, related to the attentional selection of a perceived object.

representation of the three perceived objects is formed, which have just come into view (*Color/Space Perception*, T = 139.1 s). After the peaks of the three objects have formed, a target color is selected in the *Color Attention Field* by competitive selection (*Color Attention*, T = 139.4 s). When a color is selected (*Color Attention*, T = 780 s), the ridge input to the *Color/Space Attention Field* selects an object of the target color (*Color/Space Attention*, T = 140.0 s). The horizontal position of the selected object is represented in the space attention field (*Space Attention*, T = 140 s). In the depicted example, the architecture selects the red object from the three perceived objects. Accordingly, the perceptual *Color Concept Nodes* detect that a red target object is attentionally selected (*Color Concepts*, 139.4 s < T).

The extraction of discrete concepts from a stream of continuous feature values is also done for sound perception (see Section 3.5.3). For brevity, only the visual perception channel is depicted here.

4.1.2 Action Episode

Figures 4.2 and 4.3 show an overview of an exemplary action episode. An action episode starts with the activation of the *Prior-Intention*, includes the sequence of the 3 intentions in actions *Orient to Target*, *Brake* and one of the three *LED-Transients*. The action sequence terminates with the CoS of the *Prior-Intention* (The transient detection of the CoS memory) (see Section 3.6.1).

Figure 4.2 shows screenshots of the Webots simulated sampled at different points in time. Screenshots are numbered according to their temporal order. The first row of Figure 4.3 shows the activation time course of the *Intention* and *CoS Nodes* of the intention in actions and the *Prior-Intention*. The three time courses below show the signals which are transmitted from the individual intention in actions to the motor surface and thus control the E-puck. The *Target Orientation* shown in Figure 4.3 belongs to *Orient to Target* intention and controls the orientation of the E-Puck (see Section 3.6.2). The *Brake Node* controls the brakes of the E-Puck and belongs to *Brake* intention (see Section 3.6.3). The *LED Nodes* belong to one of the 3 LED-Transients (see Section 3.6.4).

The *Prior-Intention*, which initiates an action sequence, can be activated in two ways. Either by the strategy representation as part of outcome oriented action, or by the explore *Supervisor Commands*. Figure 4.3 shows the latter case. The *Prior-Intention* node is kept active by the supervisor command. As long as no object comes into the field of view, no selection decision occurs in the *Action Selection* nodes. The E-Puck then drives in the default obstacle avoidance pattern (*Images*, (1)).

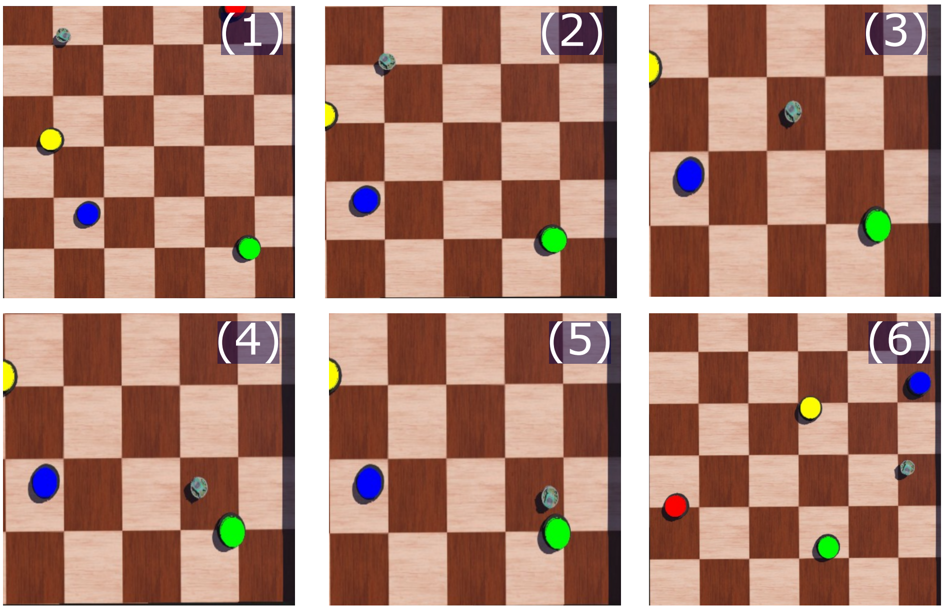

FIGURE 4.2: Screenshots from Webots showing an overview of an action episode.

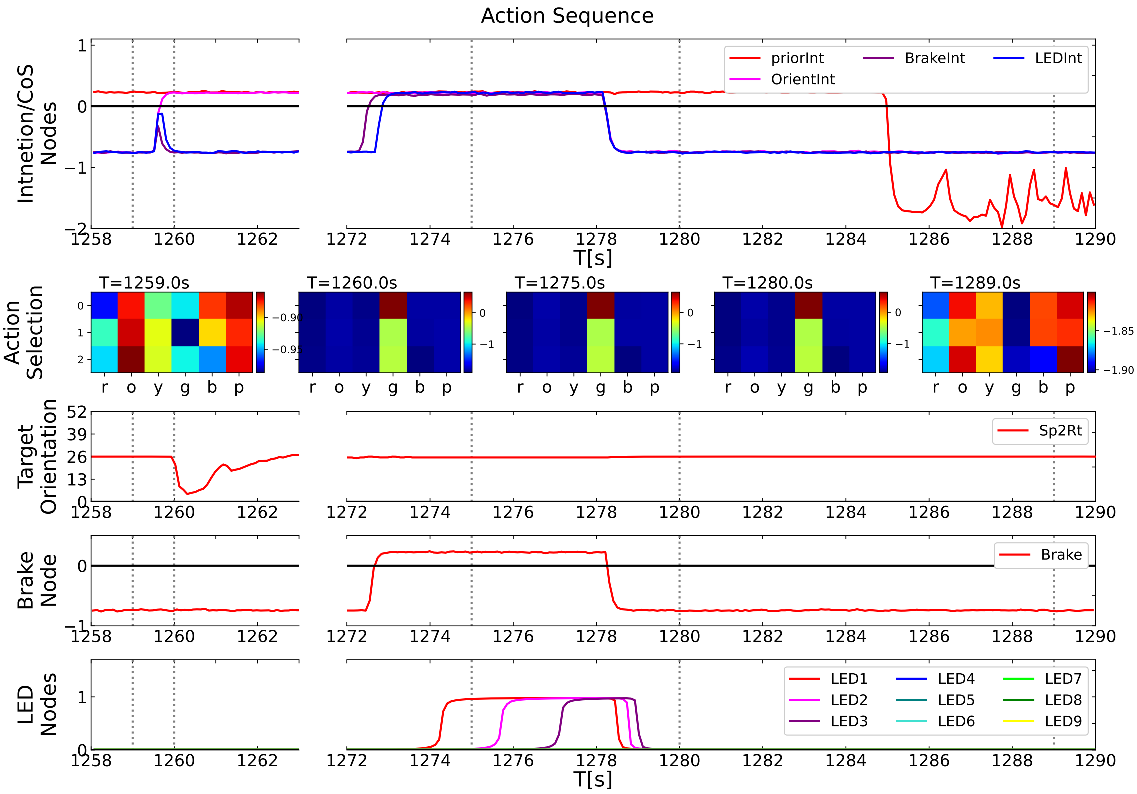

FIGURE 4.3: Activation time courses and snapshots of selected neural fields and nodes, showing an overview of an action episode.

When an object enters the field of view and is attentionally selected, the corresponding perceptual *Color Concept* nodes pass a ridge input to the *Action Selection* nodes. This, in conjunction with the global boost of the supervisor command, leads to a selection decision that initiates an action sequence targeted at the perceived object (*Action Selection*, T = 1260 s). The supra-threshold activation of the *Action Selection* nodes activates the associated action sequence by boosting the corresponding *Precondition* and *Intention Nodes*, which then sequentially pass their motor commands to the sensorimotor surface.

The intention in action *Orient to Target* controls the orientation of the E-Puck by projecting the current horizontal position of the selected object as the target orientation onto the motor surface. When inactive, the target orientation lies in the center of the field of view, which creates a straight line motion pattern (*Target Orientation*, T = 1260 s). As soon as the *Orient to Target* intention activates (*Top Row*, T = 1260 s), the target orientation relaxes to the peak position of the motor gate field and thus corresponds to the horizontal position of the object.

The *Orient to Target* intention navigates the E-Puck toward the object. The *Brake Intention* activates when it is in front of the object. The *Brake Node* the stops the E-Puck, at which point the *LED Transient* is initiated (*Top Row, Brake, LED*, T = 1275 s).

After execution of the *LED-Transient*, the intention in actions are inhibited by the *CoS*. The *Prior-Intention* becomes inhibited once the *CoS Memory* activates (*Top Row*, 1280 = T = 1289 s). In this time period, the memory trace at the action level develops.

4.1.3 Belief Formation

The architecture forms a new belief if a sound is perceived right after an action has been performed. In that case, the perceived sound is sensed as being caused by the executed action. The *Outcome Detection* node detects this action outcome and controls the formation of a new belief via the associated *Gate* and *Learning Nodes* (see Section 3.7.2).

Figure 4.4 shows an example of an activation time course of relevant nodes for belief formation. The top diagram shows the time courses of nodes organizing the belief formation process. The two diagrams below show the activation time courses of *Belief* and *Commit Nodes*. Activation snapshots of *Outcome*, *Action* and *Color Role Nodes* are shown below the time courses.

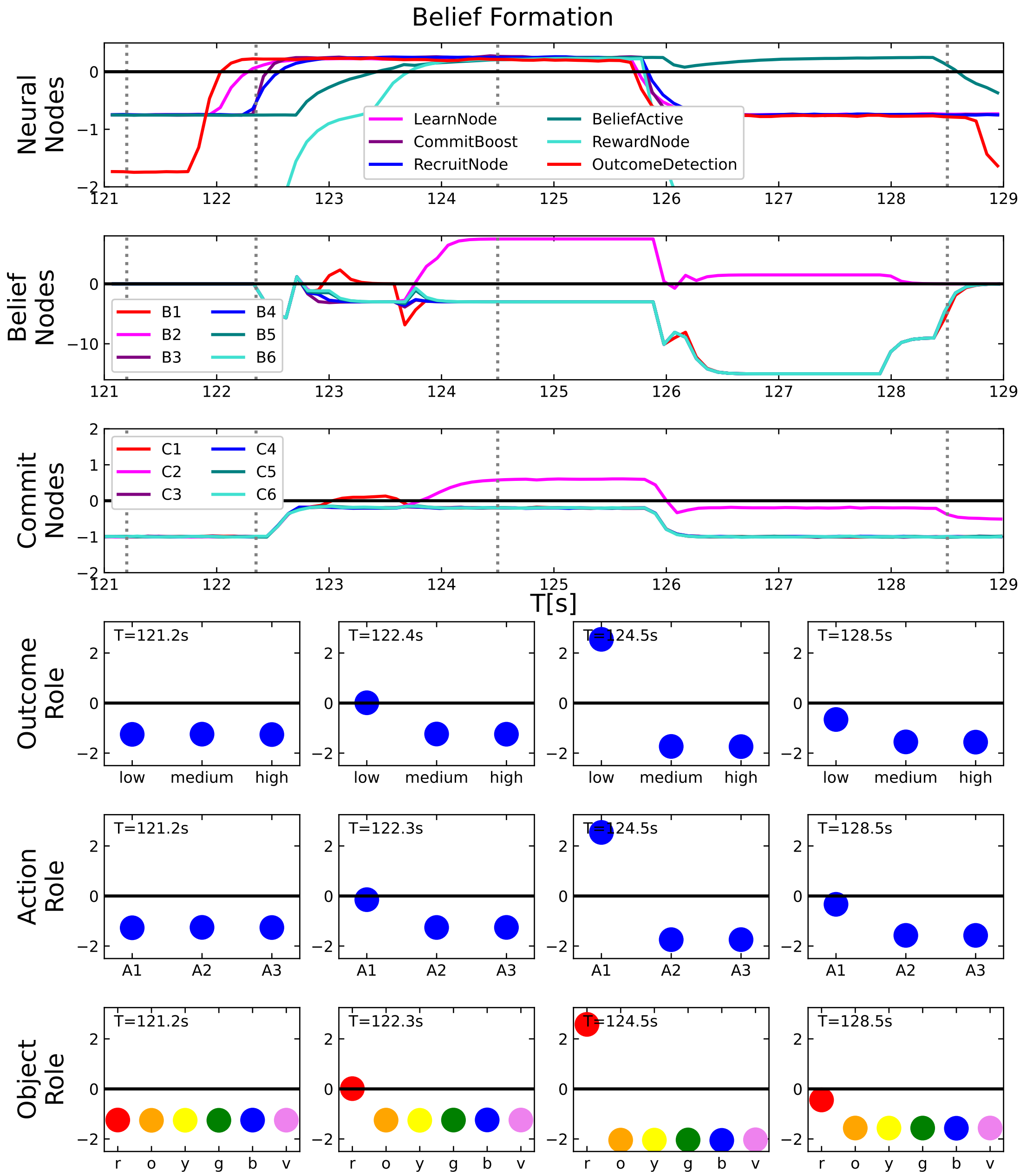

FIGURE 4.4: Activation time course and snapshots of neural nodes, related to belief formation.

At T = 121.2 s, the state of the *Belief System* is shown in its initial state before an outcome has been detected. No belief is active, no recall attempt is initiated and no sound is heard.

In the example shown, the E-Puck performs action 1 on a red object which results in a low-pitch sound. The *Outcome Detection Node* receives input from the *Sound*

Perception Node and the *Action Working Memory* nodes. These two inputs push the *Outcome Detection Node* above threshold, which signals that action 1 has caused the low-pitch sound (*Top Row*, 121.2 s <T < 122.4 s). The *Outcome Detection Node* activates the *Gate Nodes*, which in turn activate the the *Role Concept Nodes* (*Role Concept Nodes*, T = 122.4 s). In addition, the *Outcome Detection Node* initiates learning by activating the *Learning Node* (*Top Row*, T = 122.4 s).

The *Learning Node* boosts the *Commit Boost* and *Recruit Nodes*. They inhibit *Belief Nodes* already associated with a contingency by activating the corresponding *Commit Nodes* and then inducing a competitive selection decision from the pool of unassigned *Belief Nodes*. In the shown example, *Belief Node* B1 is already committed and *Belief Node* B2 is selected for belief formation (*Top Row, Belief Nodes, Commit Nodes*, 122.4 s < T < 124.5 s).

After *Belief Node* B2 has been selected, the *Reward Node* is activated. This sends a reward signal to the Hebbian connections between the *Belief* and *Role Nodes*, as well as to the plastic connections between *Commit* and the *Belief System Nodes*. The learning rate of the plastic connections is tuned such that the connection weights converge after a single learning episode (one-shot learning). The *Belief Node* B2 is then associated with the contingency that action 1 produces a low-pitch sound, when executed on a red object.

A learning episode terminates when the *Outcome Detection* node falls back below the detection threshold. This disables the *Learning*, *Recruit*, *Commit Boost* and *Reward Nodes*. The active *Commit Node* C2 is then deactivated and the *Belief Node* B2 also falls back to the default state (T = 128.5 s).

4.1.4 Opportunistic Activation

As described in Section 3.7.1, the architecture can reactivate beliefs associated with perceived objects. This is done by the *Contingency Recall Intention*. It is activated when a new object enters the field of view. If the *Recall Intention* successfully activates an associated belief, the active belief representation can opportunistically activate a goal state in the *Goal Selection* nodes (see Section 3.6.6).

Figure 4.5 shows an example of a recall episode initiated by the perception of a red object. The two upper diagrams show the time courses of the *Recall Intentions* and *Belief Nodes*. Below, activation snapshots of *Role* and *Goal Selection* nodes are shown at the four marked points in time.

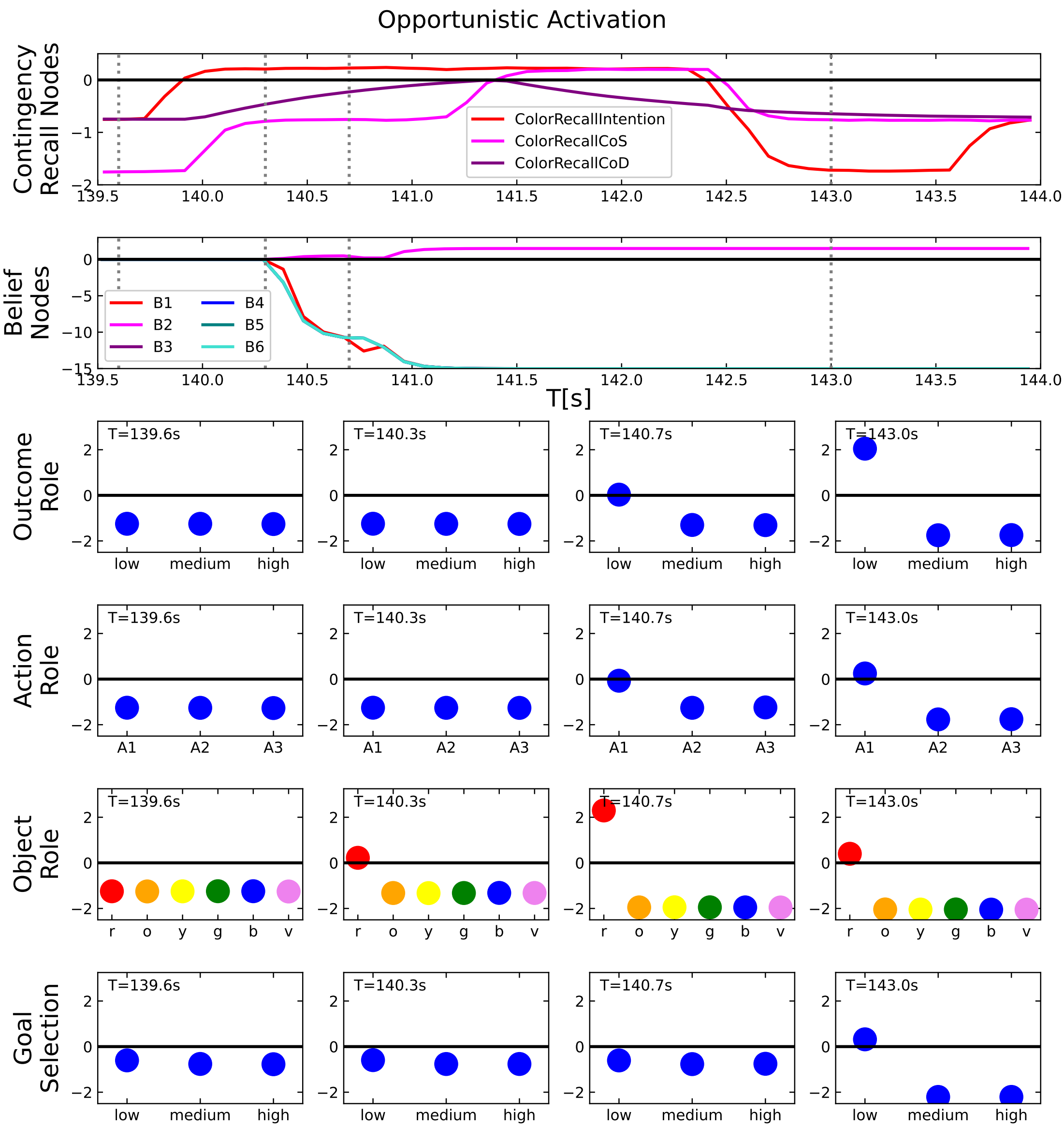

FIGURE 4.5: Activation time course and snapshots of neural nodes, related to opportunistic activation.

At T = 139.6 s all shown nodes are below activation threshold. The recall intention is activated after a red object enters the field of view and is attentionally selected. The *Color Transient Detection* then activates the *Contingency Recall Intention*, which in turn activates the red *Role Color* node by means of boosting the *Color Gate* nodes (*Color Recall*, *Color Role*, T = 140.3 s).

The active *Role Color* node activates the *Belief Node* B2, which is associated with the previously learned contingency (see above). This in turn activates the "action1" and "low-pitch" *Action* and *Outcome Role* nodes (T >= 140.7 s). The *Outcome Role* nodes project their content to the *Goal Selection* nodes. After the recall of the contingency, the active "low-pitch" *Outcome Role* node activates the "low-pitch" goal state in the *Goal Selection* nodes (*Goal Selection*, T = 143 s).

An active belief is self-sustaining. Even if the *Recall Intention* terminates, the *Role Concept* nodes and the *Belief Node* remain active. Accordingly, the opportunistically induced goal state remains active until it is inhibited by its *CoS Node* or a new recall attempt deactivates the belief (*Belief*, T > 143 s).

4.1.5 Outcome-Oriented Behavior

A goal can be activated opportunistically, as shown above, or by boosts from the *Supervisor Command* nodes. Outcome oriented action is achieved by attempting to recall an action strategy, that results in the desired outcome. Through this mechanism, goal states at the outcome level determine action (see Section 3.6.6).

Figure 4.6 and Figure 4.7 show an example of how an action plan is activated by the *Strategy Recall Intention*. The upper two diagrams of Figure 4.6 show the activation time course of the *Recall Intention* and *Belief Nodes*. The four rows below show activation snapshots of the *Role Concept* and *Goal Selection* nodes at 4 characteristic points in time. The top two rows of Figure 4.7 show again the time course and snapshots of *Recall Intention* and *Goal Selection* nodes. This is done for easier comparisons between Figures 4.6 and 4.7. The two lower rows show activation snapshots of the working memory *Strategy* nodes and the *Action Selection* nodes of the *Prior-Intention*.

At T = 2137.2 s, none of the *Goal Selection* nodes are active. After a while, the "high-pitch" goal state is activated by one of the *Supervisor Command* nodes. The newly active "high-pitch" *Goal Selection* node is detected by the corresponding *Goal Transient Detection* node, which in turn activate the *Strategy Recall Intention* and thus triggers a recall attempt (Figure 4.6, *Goal Selection*, *Strategy Recall*, T = 2138 s).

The *Strategy Recall Intention* reactivates the *Belief Node* B1, which is associated with the previously learned belief that a "high-pitch" sound can be generated by performing action 3 on a blue object (Figure 4.6, *Belief*, *Role Concepts*, T = 2141 s).

Since the active "high-pitch" *Outcome Role* node matches the active *Goal Selection* node, the *Color Role* and *Action Role* nodes activate the two-dimensional *Strategy* nodes, which represents the recalled action plan in the two-dimensional action/color-space (Figure 4.7, *Strategy*, T = 2141 s).

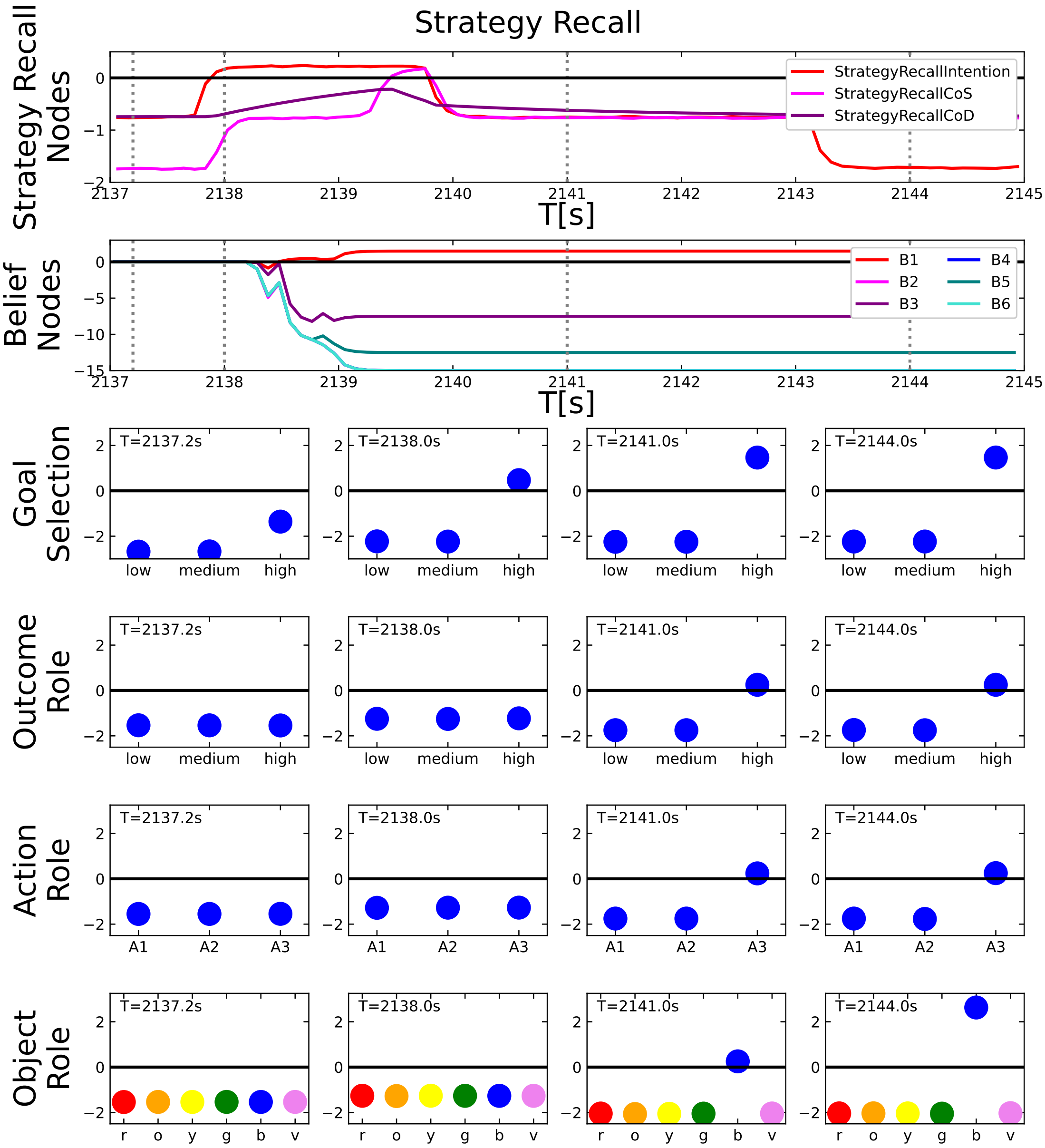

FIGURE 4.6: Activation time course and snapshots of neural nodes, related to strategy recall.

The *Strategy* nodes preshape the *Action Selection* nodes. When a blue object enters the field of view, the ridge input from the *Perception Color Concept* nodes and from the *Strategy* nodes overlap, resulting in a selection decision of the action sequence matching the recalled strategy (Figure 4.7, *Action-Selection*, T = 2144 s).

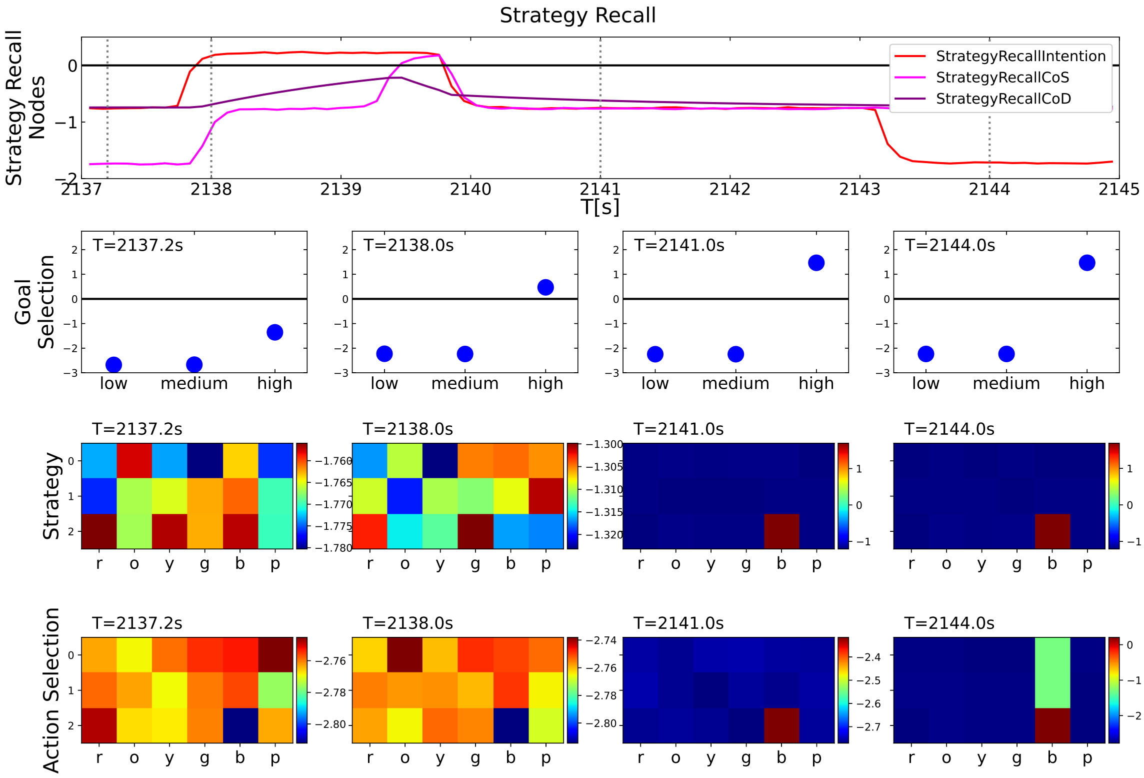

FIGURE 4.7: Activation time course and snapshots of neural fields and nodes, related to outcome oriented action.

4.1.6 Feature Guiding

The action sequence demonstrated above is object-oriented, in the sense that the content of the *Orient to Target* and *Brake* intentions depend on the attentionally selected object. The mechanism of attentional selection described in Section 3.5.2 is modulated by a simplified form of feature guidance, that amounts to a top-down preshape of the *Color Attention* field coming from the *Strategy* nodes.

Figure 4.8 shows example activation snapshots of fields and nodes together with camera images, from a time window of test phase 3. The *Supervisor Command* nodes have already activated the "high-pitch" *Goal Node* at the beginning of the shown time window, and the architecture has already been able to generate an action plan through the process demonstrated in Section 4.1.5 (*Goal-Selection*, *Strategy*, T = 2080 s).

At T = 2085 s, the supervisor places a green distractor object in the field of view of the E-Puck. This leads to an opportunistic activation of the "medium-pitch" *Goal-Selection* node through the process shown in figure 4.5. However, the "high-pitch" goal remains stable with respect to the distractor input. Accordingly, the selected

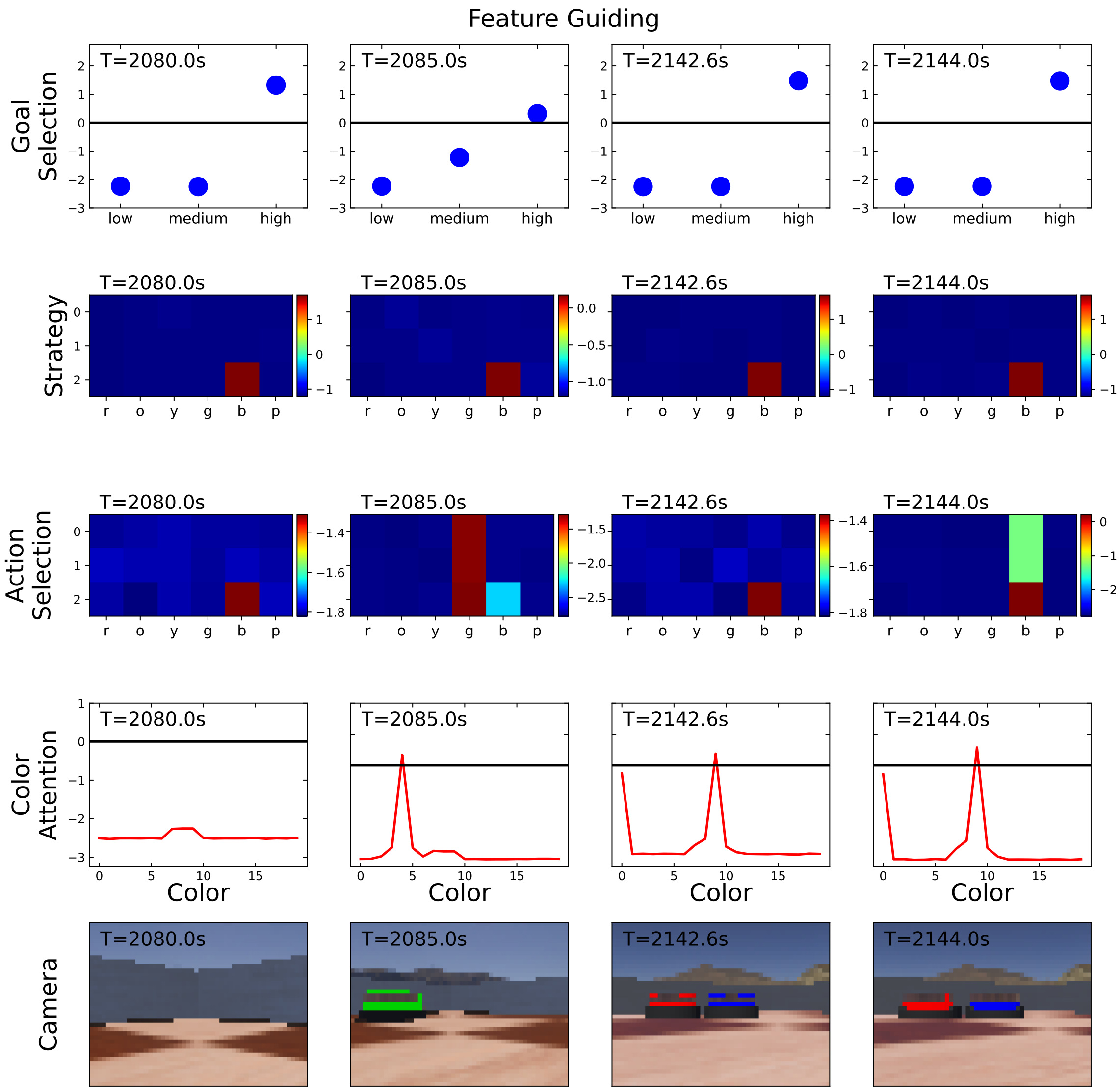

FIGURE 4.8: Camera Images and activation snapshots of neural fields and nodes, related to feature guiding and ignoring of distractors.

Strategy remains unchanged. Opportunistic input from the *Perception Color Concept* nodes and from the *Strategy* field do not overlap in the *Action Selection* field. Therefore, there is no supra-threshold activation in the *Action-Selection* field. The object is ignored (*Goal-Selection*, *Strategy*, *Action-Selection*, T = 2085 s).

In the next snapshot (T = 2142.6 s), the supervisor simultaneously places a blue and a red object in the E-Pucks field of view. The red object is a distractor. The blue object, on the other hand, can be used in combination with action 3, to create the desired "high-pitch" sound.

The *Strategy* field reflects this action plan and gives a ridge input to the *Color-Attention* field. This leads to the blue input winning in the competitive attentional selection over the red input (*Strategy*, *Color-Attention*, T = 2142.6 s).

For the *Action-Selection* field, the overlap of the ridge color input and the strategy leads to a detection instability, which initiates the selected action to produce the desired "high-pitch" sound (*Action-Selection*, T = 2144 s).

4.1.7 Opportunistic Strategy

If several action-object combinations lead to the same sound, the architecture opportunistically switches its strategy if opportunistic activation from a perceived object matches the current goal.

Figure 4.9 shows activation snapshots from another exemplary time course of test phase 1. In this phase, the "low-pitch" goal was activated by the supervisor and the architecture recalled the action plan "Action1 on red object" to achieve this goal (*Goal-Selection*, *Strategy* T = 1836 s).

In this example, the supervisor places a yellow object in the E-Pucks field of view (*Camera*, T = 1840 s). The yellow object also produces a "low-pitch" sound, when action 3 is performed on it. Via opportunistic activation, the perception of the yellow object activates the corresponding belief representation in the *Belief System* (as in Figure 4.5).

The architecture detects via the *Goal-Recall Match* nodes that it can also reach its current "low-pitch" goal with the yellow object. As a result, the *Strategy* field receives input from the *Color Role* and *Action Role* nodes via the *Strategy Gate* nodes (see Section 3.6.5). This input destabilises the old strategy and leads to the formation of a new action plan involving the yellow object (*Strategy*, T = 1840.8 s). The new action plan matches the perceived yellow object, activating the *Action-Selection* field and initiating the goal-directed action (*Action-Selection*, T = 1842.5 s).

4.2 Learning Phase

During learning phase, objects of different colours are distributed to random places in the arena. The E-Puck is given the "explore" *Supervisor Command* to perform various actions on objects it encounters. As described in Section 3.6.7, action selection is modulated by a weak inhibitory memory trace, while opportunistic activation of

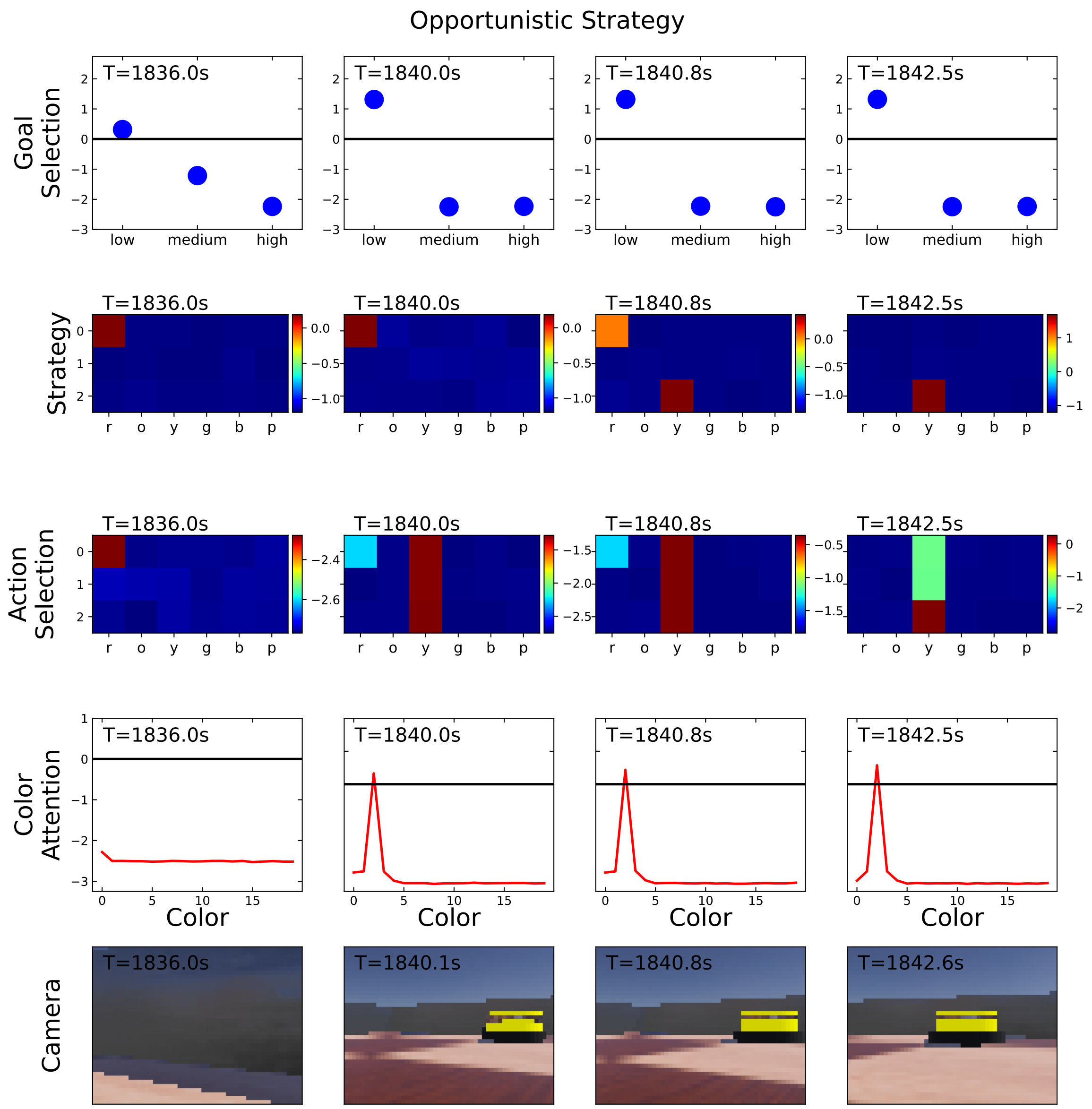

FIGURE 4.9: Camera Images and Activation snapshots of neural nodes, related to the opportunistic switching of strategies.

goal states may lead to outcome-oriented action. According to Section 3.2.1, the E-Puck is expected to first explore the action/color-space. As the architecture learns action-outcome contingencies, opportunistic activation should lead to exploitative behavior. However, as the goal states habituate the E-Puck to switch back to exploratory behaviour.

4.2.1 Exploration

In the learning phase, the *Prior-Intention* is activated by the "explore" *Supervisor Command* node. In addition, the command gives a homogeneous boost to the *Action-Selection* nodes, allowing them to activate by opportunistic input from the *Perception Color Concept* nodes. After an action is executed, and while the *CoS Node* of the *Prior-intention* is active, activation is built up in the inhibitory memory trace.

Figure 4.10 shows an example of two action selection decisions. The time courses of the *Intention Nodes* comprising the *Prior-Intention* are shown in the upper diagram. The rows below show the *Action-Selection* nodes and the memory trace at different points time.

In the examples shown, a red object enters the field of view of the E-Puck at two different points in time (T = 28 s, T = 106 s). In the first action episode, opportunistic ridge input from the "red" *Perception Color Concept* node leads to competitive activation of action 3 (*Action-Selection*, T = 38 s). After successful execution of the selected action, activation in the memory trace builds up (*Inhibition of Return*, T = 44 s).

In the next action episode, opportunistic ridge input from the "red" *Perception Color Concept* node again leads to an action selection. The previously executed action 3 and 2 are slightly inhibited by the memory trace, which gives action 1 an advantage in competitive action selection (*Action-Selection*, T = 106 s). After action 1 is performed, it is also inhibited for future action selections (*Memory Trace*, T = 109 s, T = 128 s). Through this mechanisms, the E-Puck explores the action/color-space through multiple action selections that are modulated by an inhibitory memory trace that keeps track of past actions.

4.2.2 Goal Habituation

As shown in Section 4.1.4, opportunistic activation of goal states can lead to exploitative behaviour in the learning phase. If a desired sound has been successfully produced several times, the inhibitory memory trace at the goal level starts to inhibit corresponding *Goal-Selection* node. The sound is then no longer considered desirable. It is therefore expected that after the E-Puck learns a contingency, it will not longer explore action/color-space for the corresponding color, but instead engage in opportunistic goal oriented behavior, until the corresponding goal state habituates, at which point the E-Puck should revert back to exploratory behavior.

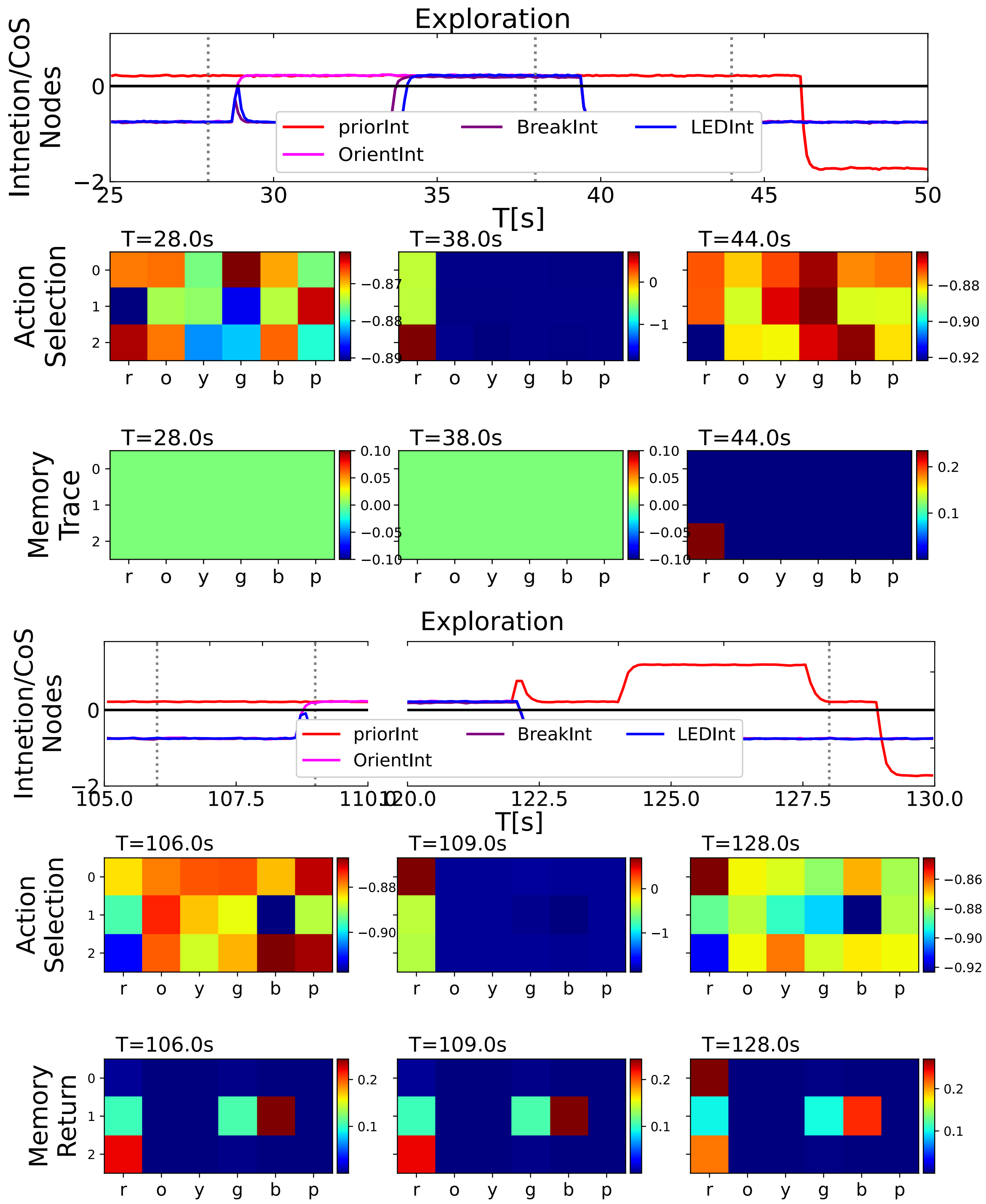

FIGURE 4.10: Activation time course and snapshots of neural fields and nodes, related to the exploration of the color/action-space.

Figure 4.11 shows *Goal-Selection* and *Action-Selection* nodes, together with the excitatory and inhibitory *Memory Traces* at the goal level, for two consecutive action-episodes.

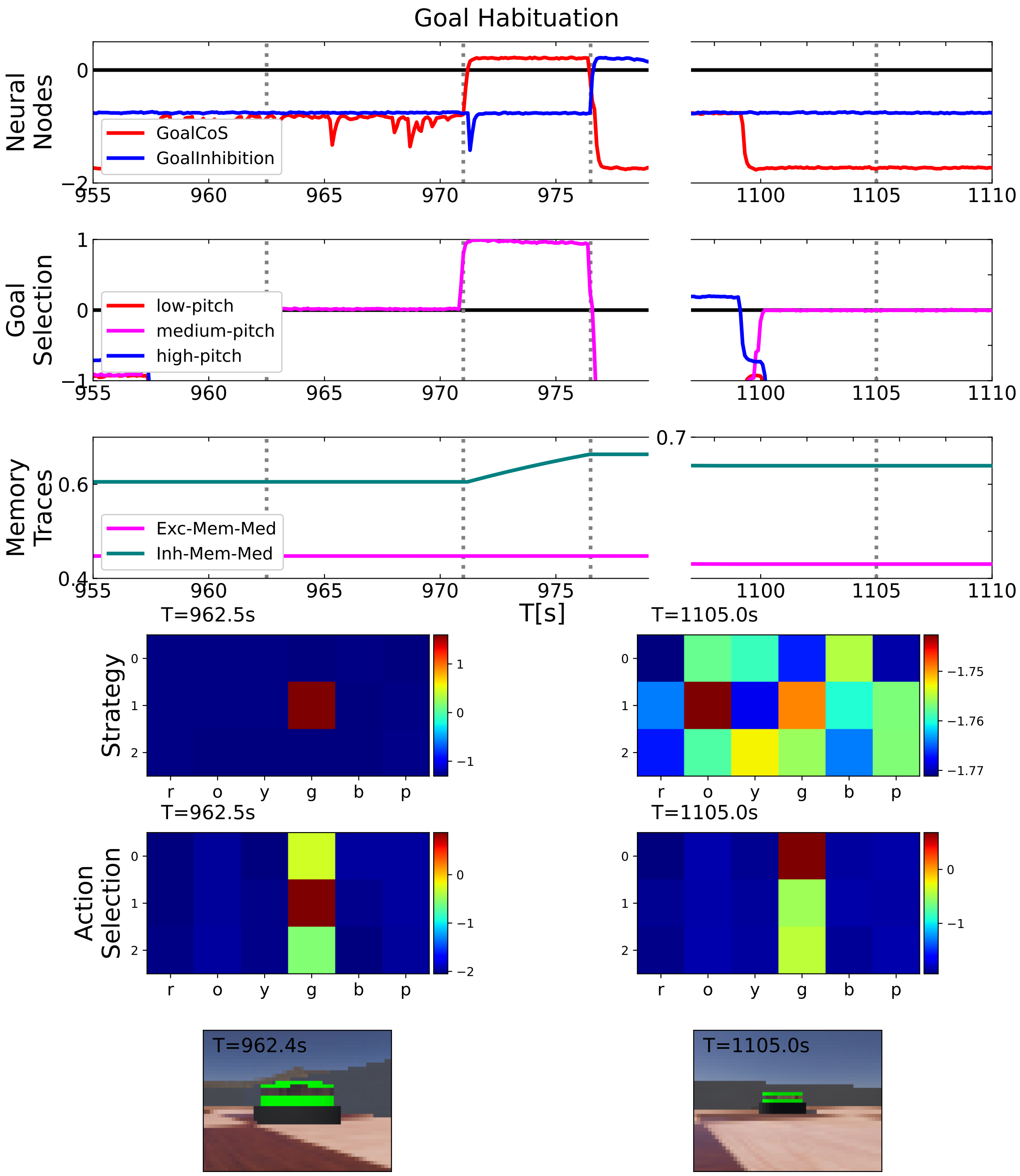

FIGURE 4.11: Activation time course and snapshots of neural fields and nodes, showing how goals become habituated against opportunistic input.

In the first action episode, opportunistic activation of the perceived green object is just sufficient activate the "medium-pitch" *Goal-Selection* node (*Goal-Selection*, T = 962.5 s). The recalled strategy leads to the selection of action 2, which produces the desired "medium-pitch" sound (*Action-Selection*, T = 962.5 s). The inhibitory *Memory Trace* builds up activation after each goal reaching episode (the excitatory memory trace is already saturated) (*Memory Trace*, 971 s < T < 976 s), which is controlled by the *CoS* and *CoD* nodes at the *Goal* level.

In the second action episode (T = 1105 s), the inhibitory *Memory Trace* is just strong enough that the "medium-pitch" *Goal Selection* node is not able to initiate a recall episode (T = 1105 s). Accordingly, the architecture does not form a strategy representation and switches back to exploratory behaviour. Due to the inhibitory memory trace at the action-level, the *Action-Selection* field selects action 1 (*Action-Selection*, T = 1105 s).

4.2.3 Exploratory Behavior

The two processes shown in Section 4.2.1 and Section 4.2.2 overlap in the learning phase. The initial exploratory behaviour is replaced by exploitative behaviour once a contingency has been learned that allows the E-Puck to produce a desirable sound. Figure 4.12 shows the behaviour of the E-Puck over the learning phase in terms of actions performed and sounds produced. The two upper diagrams show the time course of the executed actions and the generated sounds. The colour of objects on which actions were performed are represented by the marker color. Below the time courses is a histogram of the action/color space. The histogram shows the number of times an action/object combination was executed over the course of the learning phase. Actions that produce a sound according to table 3.1 are indicated by a circle.

The time courses show that the E-Puck switches to exploitative behavior in relation to an object, once an action producing a sound was discovered for that object. This is pronounced for the green object. After first carrying out the actions 3 and 1, once action 2 has been produced (which plays a sound), the E-Puck sticks to this action for 5 action episodes in a row.

The histogram shows that actions are favored for which the target object produces a sound. But in this example, you can also see that this is not the case for the yellow object. Since the "low-pitch" sound can be produced with both red and yellow objects, the corresponding goal habituates more quickly in relation to the object, resulting in a faster return to exploratory behavior.

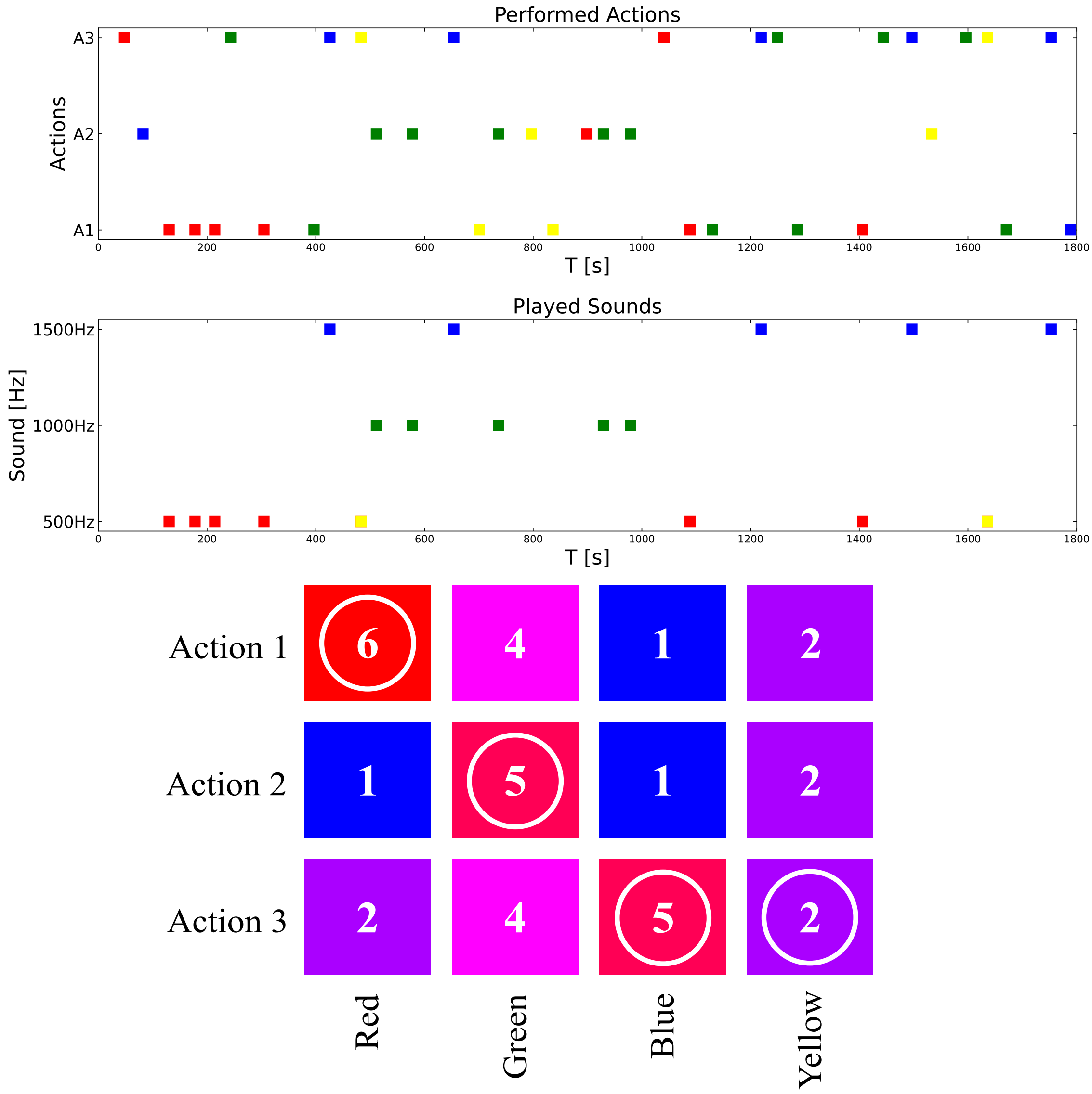

FIGURE 4.12: Overview of performed actions and produced sounds in the learning phase.

4.3 Goal Selection Phase

In the goal selection phase, a *Goal-Selection* node is activated by a homogeneous boost from the *Supervisor Command* nodes. The goal is selected by competitive selection. The selection decisions are modulated by an excitatory and an inhibitory *Memory Traces*, that operate on different time scales and keep track of past goal reaching episodes. After each selection decision, a corresponding goal object is presented to the E-Puck, allowing it to play the desired sound. As described in Section 3.6.6, the excitatory *Memory Traces* develop faster than the negative ones. Due to the resulting dynamics, selection decisions should first display familiarity preference, followed by later novelty preference (see Section 3.2.3).

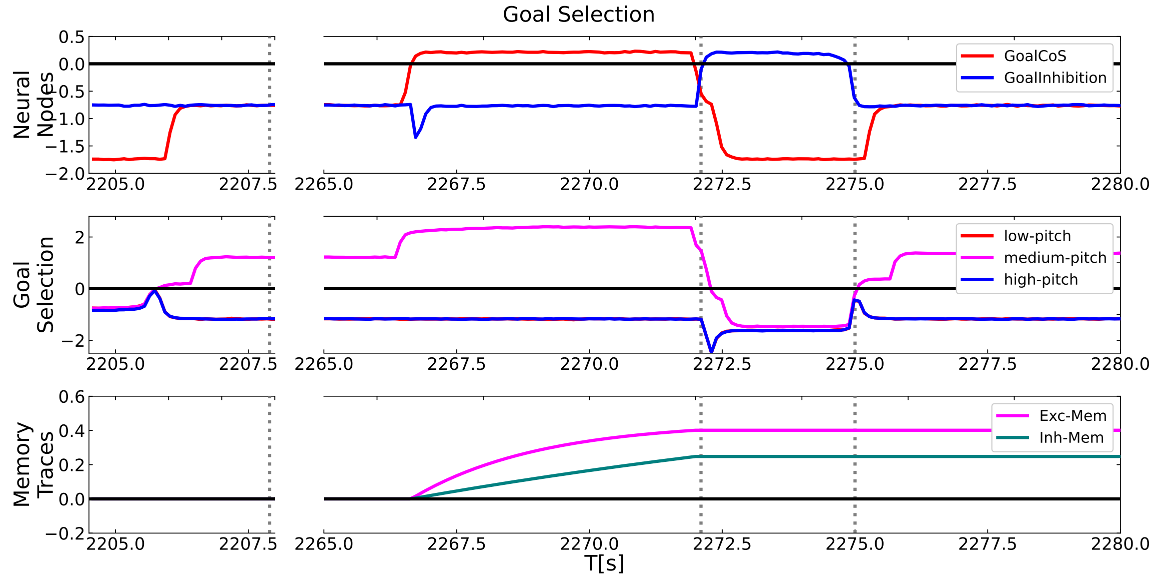

FIGURE 4.13: Activation time course of neural nodes, related to the selection dynamics of goal selection decisions.

Figure 4.13 shows an example activation time course of two successive selection decisions. The upper diagram shows the *CoS* node of the selected goal in addition to the *Transient Detection* of the *CoS Memory*, which inhibits the *Goal Selection* nodes. The middle diagram shows the activation time course of the *Goal-Selection* nodes. The bottom diagram shows the development of the excitatory and inhibitory *Memory Traces*.

At the beginning of the shown time course, the *Selection Signal* of the *Supervisor Command* nodes leads to a competitive selection decision, in which the "medium-pitch" sound is initially selected as the goal (*Goal-Selection*, $T < 2207.9$ s).

After the selection decision, the supervisor presents the E-Puck with a matching green object. The *Memory Traces* develop while the *CoS* at the goal level is active. As soon as the E-Puck has produced the desired "medium-pitch" sound, the corresponding *Memory Traces* build up activation (*Memory Traces*, 2207.9 s $< T < 2272.1$ s).

Since the excitatory *Memory Traces* develop faster than the inhibitory ones, the modulated resting level of the "medium-pitch" *Goal-Selection* node subsequently increases slightly compared to the other *Goal-Selection* nodes (*Goal-Selection*, $2272.1 < T < 2275$ s).

In the next selection decision, the "medium-pitch" *Goal-Selection* node has a competitive selection advantage over other *Goal-Selection* nodes. This leads to the same "medium-pitch" goal being selected again (T > 2275 s). This amounts to a form of familiarity preference in goal selection, as desired goals which have been achieved in the past tend to be selected again.

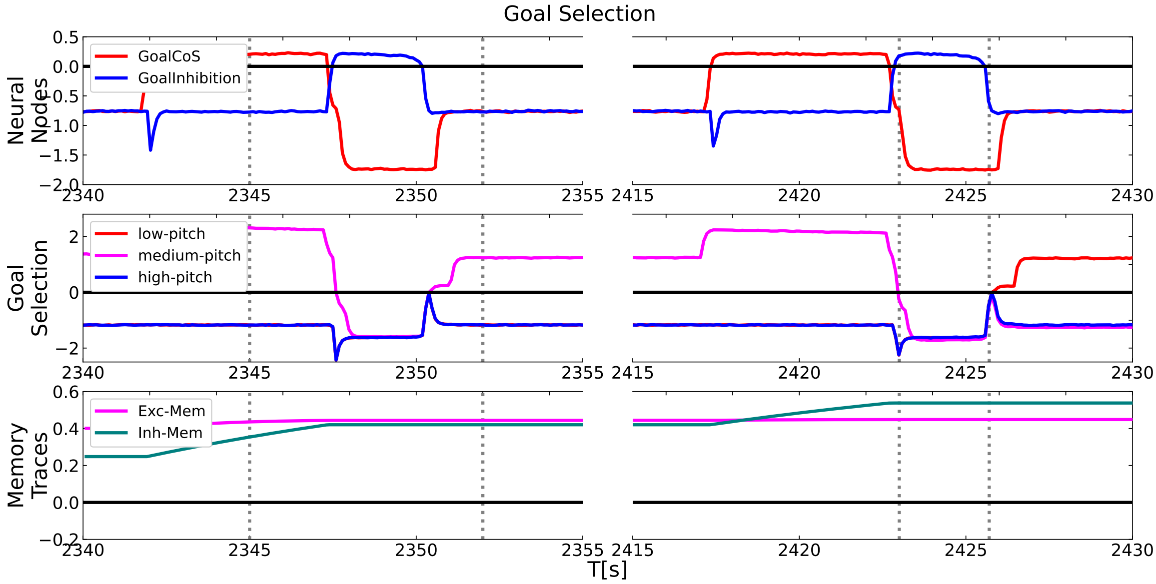

FIGURE 4.14: Activation time course of neural nodes, related to the selection dynamics of goal selection decisions.

Figure 4.14 shows the activation time courses of two consecutive selection decisions at a later point in time. In this case, the "medium-pitch" excitatory *Memory Trace* is already almost saturated. Initially, the "medium-pitch" *Goal-Selection* node is again selected by competitive selection. The excitatory *Memory Trace* still has a greater influence here (*Goal-Selection*, 2345 < T < 2352 s).

After the generation of the "medium-pitch" sound, the inhibitory *Memory Trace* exceeds the excitatory one, so that the resting level of the "medium-pitch" *Goal-Selection* node is below that of the other *Goal-Selection* nodes (*Goal Selection*, *Memory Traces*, 2423 s < T < 2426 s).

In the following selection decision, the "medium-pitch" *Goal-Selection* node is at a disadvantage. In the shown example, the "low-pitch" *Goal-Selection* node is selected (2426 s < T). This amounts to a form of novelty preference in goal selection, as desired goals eventually become habituated leading to a selection of a different goal.

Figure 4.15 shows the course of goal selection decisions in the goal selection phase. It can be seen that the same selection decision of the "medium pitch" goal is made several times in succession. This can be seen as a familiarity preference

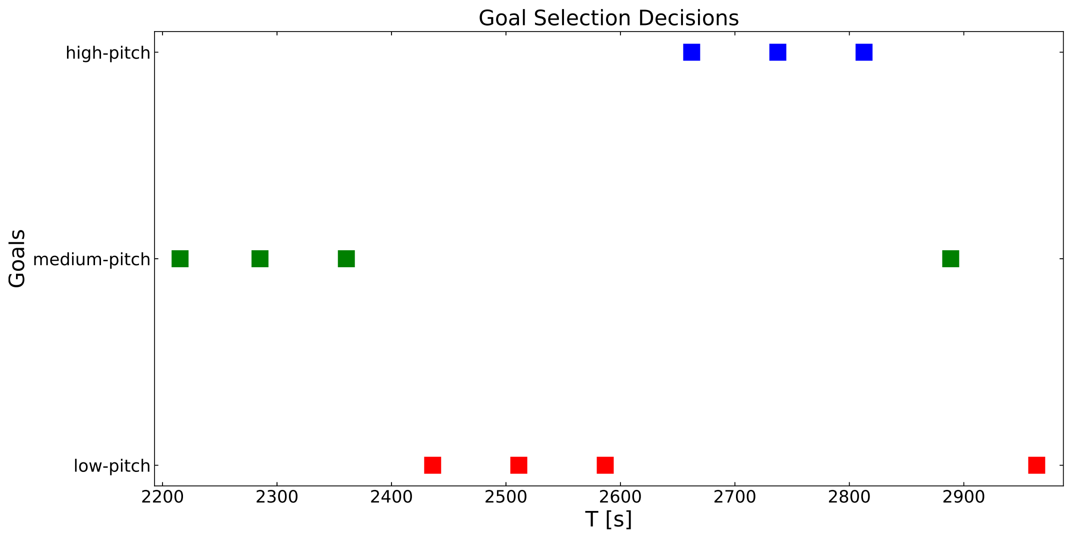

FIGURE 4.15: Time course showing goal selection decisions in the goal selection phase.

generated by the excitatory *Memory Trace*. After three episodes of producing the "medium-pitch" sound, a change in selection behaviour occurs. This habituation in selection causes the "low-pitch" and eventually the "high-pitch" *Goal-Selection* node to be selected. This habituation in selection decisions is produced by the modulation through the inhibitory *Memory Traces*. The progression of initial stabilization of selection behaviour followed by eventual habituation is consistent with past work, in which a pair of excitatory and inhibitory *Memory Traces* at the action-level was able to account for early familiarity preference and later habituation in motor behavior.

4.4 Goal Switching Phase

In the goal switching phase, the stability of a selected goal is probed through opportunistic activation by distractor objects. As in the goal selection phase, the *Supervisor Command* nodes give a *Selection Signal* that leads to a competitive selection decision in the *Goal-Selection* nodes. However, unlike in the goal selection phase, the *Goal Selection* nodes are not inhibited after *CoS* at the goal-level. The E-Puck executes a self-selected goal multiple times in succession.

As described in Section 3.2.4, the E-Puck is presented with an alternating sequence of target and distractor objects. The faster excitatory *Memory Trace* should stabilise the goal state against opportunistic input from distractors for the first goal reaching attempts. After several goal reaching episodes the inhibitory *Memory Trace* should build up enough inhibition that opportunistic distractor input destabilizes the active *Goal-Selection* node, leading to opportunistic goal switching (see Section 3.6.6).

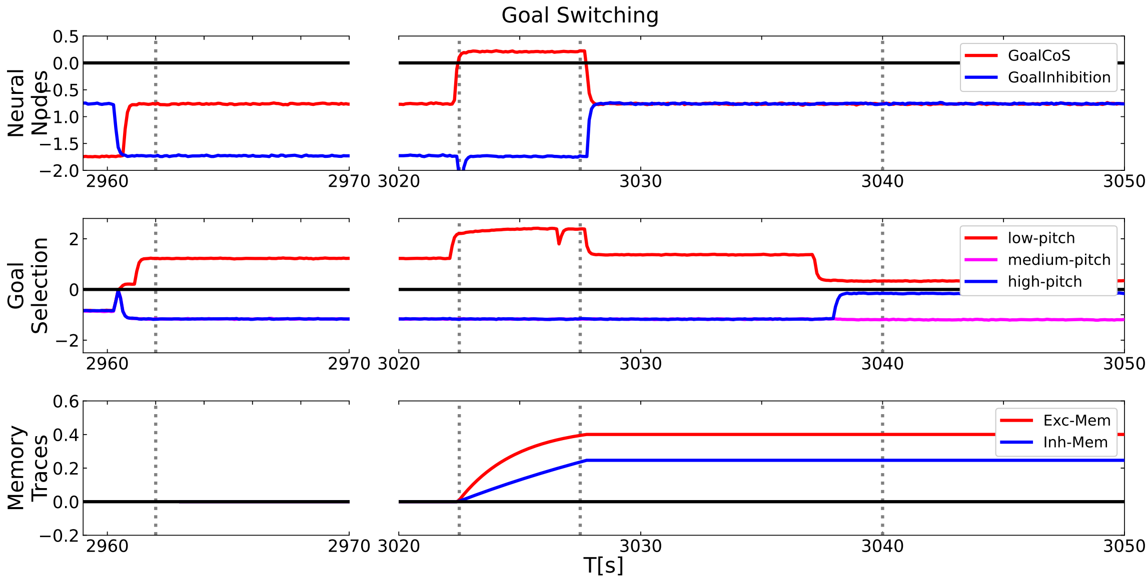

FIGURE 4.16: Activation time course of neural nodes, demonstrating goal stabilization through excitatory memory traces.

Figure 4.16 shows an activation time course of the *Goal-Selection* nodes in a goal reaching episode and the subsequent probing of goal stability by opportunistic distractor input. The upper diagram shows the activation of the *CoS Node* of the goal-level. The middle diagram shows the activation of the *Goal-Selection* nodes. The development of the *Memory Traces* is shown below.

In the beginning of the goal switching phase, the selection signal from the *Supervisor Commands* gives a homogeneous boost to the *Goal-Selection* nodes and thus leads to a competitive selection decision. In the example shown, the "low-pitch" *Goal-Selection* node was activated ($T = 2962\,s$).

After successfully producing the desired sound (signified by the *CoS Node*), the excitatory and inhibitory *Memory Traces* build up activation. The faster time scale of the excitatory *Memory Trace* causes the "low-pitch" *Goal-Selection* node to be boosted relative to the other *Goal-Selection* nodes, stabilising it against distractor input ($3022.5\,s < T < 3027.5\,s$).

After producing the "low-pitch" sound, the supervisor presents the E-Puck with a blue distractor object. The perception of this object leads to the opportunistic activation of the "high-pitch" *Outcome Role* node via the *Contingency Recall Intention*. Input of the *Outcome Role* node to the "high-pitch" *Goal-Selection* node, lifts it close to the detection threshold. At the same time, the "low-pitch" *Goal-Selection* node operates close to the reverse detection instability. In this example, stabilization from the excitatory *Memory Trace* is sufficient to stabilize the "low-pitch" *Goal-Selection* node against opportunistic activation of the "high-pitch" *Goal-Selection* node T > 3040 s).

After probing the goal stability, the next goal reaching episode is initiated by presenting the E-Puck with the next red target object. The mechanism described above is how excitatory *Memory Traces* stabilize active goal representations, whenever the desired goal has been achieved. Figure 4.17 shows the activation time course of the *Goal-Selection* nodes, at a later point in time. Here, the excitatory "low-pitch" *Memory Trace* is already saturated. Further goal reaching episodes do not lead to a stabilization of the "low-pitch" goal state.

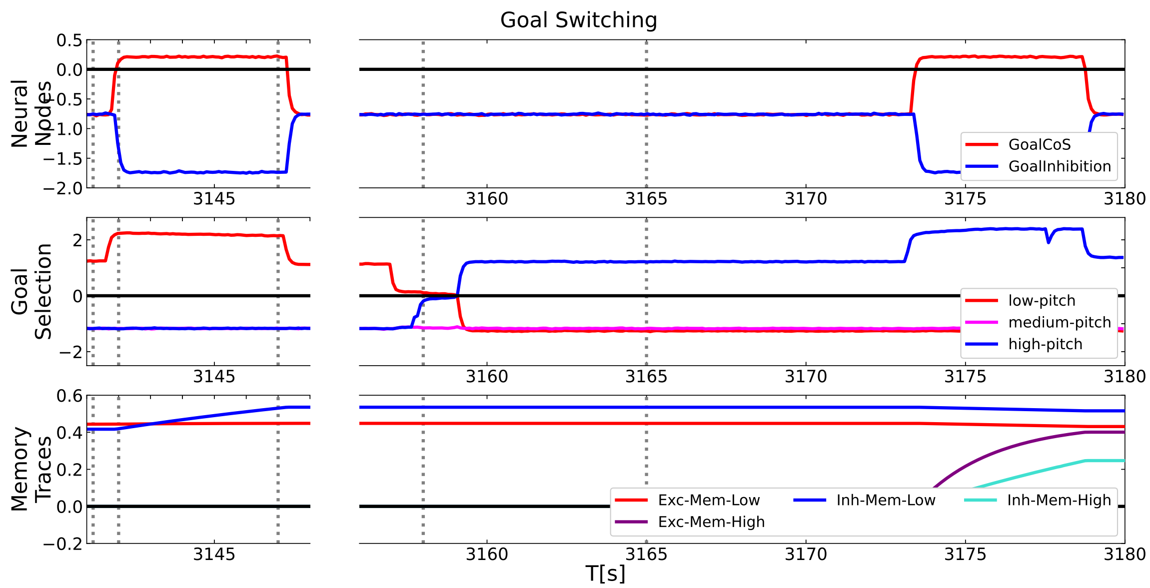

FIGURE 4.17: Activation time course of neural nodes, demonstrating goal habituation through inhibitory memory traces.

In the example shown, the "low-pitch" *Goal-Selection* node is still active (*Goal-Selection*, T = 3141.2 s). The supervisor again first presents the red target object to the E-Puck. After a successful goal reaching episode (marked by the *CoS Node*), the *Memory Traces* build up activation, whereby the inhibiting *Memory Trace* now increases significantly more than the already saturated excitatory one. (*Memory Traces*, 3141.2 s< T < 3142 s).

After this, the E-Puck is again presented with a blue distractor object, which again leads to the opportunistic activation of the "high-pitch" *Outcome Role* node. The input to the "high-pitch" *Goal-Selection* node lifts it back close to the detection instability, while the "low-pitch" *Goal-Selection* node operates close to the reverse detection instability (*Goal-Selection*, T = 3142 s).

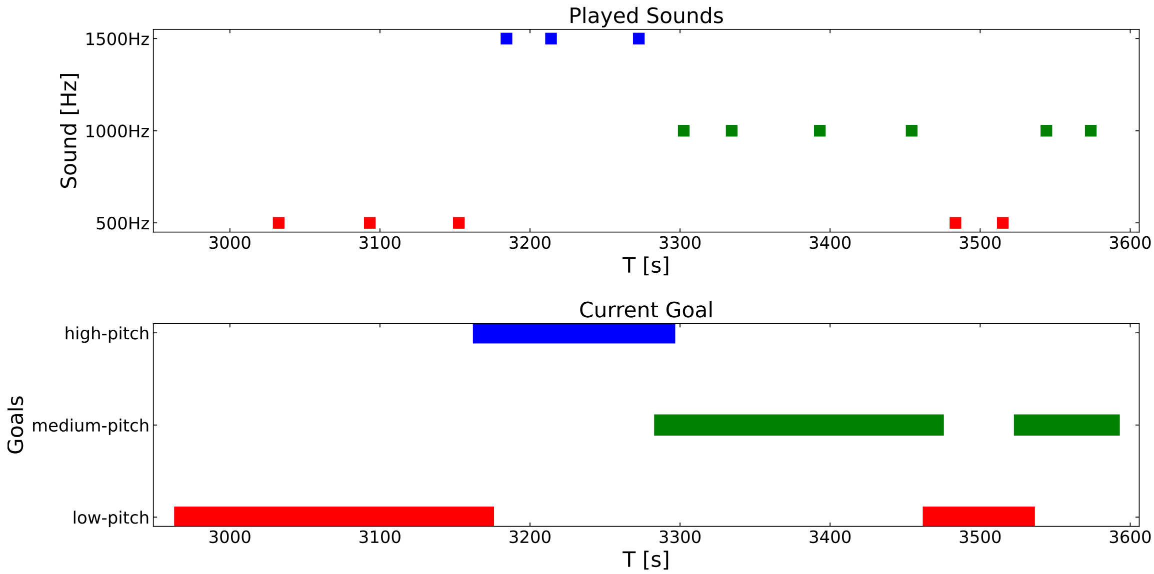

FIGURE 4.18: Active goals during the goal switching phase, showing time course of goal destabilization's.

The inhibitory *Memory Trace* is strong enough so that the "high-pitch" node can destabilize the "low-pitch" *Goal-Selection* node. Global inhibition from the "high-pitch" node pushes the "low-pitch" node through the reverse detection instability, causing the "high-pitch" *Goal-Selection* node to reach detection threshold (*Goal-Selection*, 31412 s< T < 3165 s). The newly selected "high-pitch" goal is stabilized by the excitatory *Memory Trace* (*Memory Trace*, T > 3165 s) until the inhibiting *Memory Trace* is again strong enough for the "high-pitch" *Goal-Selection* node to be destabilized.

Together, these examples show that the combination of excitatory and inhibitory *Memory Traces* initially lead to the stabilization of desired goals as they are being achieved, even if opportunistic input would favour a different goal. After repeated production of the desired sounds, the dynamics of the two *Memory Traces* eventually lead to the habituation of a goal.

Figure 4.18 shows the time course of active *Goal-Selection* nodes together with the time course of produced sounds. It can be seen that a goal remains stable over several goal reaching episodes and is finally destabilized by opportunistic input. The progression shown demonstrates that a pair of *Memory Traces* generically lead to stabilization of goals against opportunistic activation, while at the same time allowing exploitative behavior as a result of opportunistic input, when a current goal is sufficiently inhibited.

Chapter 5

Conclusion

In summary, this thesis provides a neurally plausible DFT model of Ideomotor Theory, that enables a simulated robot to act and learn in its environment. The association of actions with their outcomes is achieved by autonomously forming beliefs about action-outcome contingencies via Hebbian learning. Neural representations of desired outcomes can drive ideomotor action by inverting previously learned action outcome contingencies. The proposed model is capable of autonomously selecting and stabilizing its own goals against opportunistic activation. This thesis suggests a model on the dynamics of goals that is in line with previous DFT accounts of perseveration and habituation in motor behavior (Aerdker, Feng, and Schöner, 2022), and enables both stabilization of active goals and opportunistic behavior. The dynamics of goals is investigated in a small toy experiment loosely based on the task switching paradigm (Arrington and Logan, 2004). The experiment is able to demonstrate that signatures such as early familiarity preference and later novelty preference can arise at the level of goals, through memory traces in excitatory and inhibitory layers.

As a model of Ideomotor Theory, this model can be seen as an attempt to give a more comprehensive account than what earlier computational models provided (Herbort and Butz, 2012). The implementation in a neurally embodied agent demonstrates how the postulated inversion of associated action and outcome representations can actually give an embodied agent the possibility to act in a goal-oriented and opportunistic fashion.

As a model of the neural dynamics of goals, this model can be seen as an integration of the DFT account of Intentionality (Tekulve and Schöner, 2020) with previous DFT models accounting for stabilization and inhibition signatures found in motor and perceptual experiments (Aerdker, Feng, and Schöner, 2022). Neural fields accounting for such signatures were either grounded directly into the sensorimotor surface (e.g. for motor primitives) or (as a shortcut) were postulated to be grounded eventually. By integrating them into the hierarchical world-to-mind structure, it is possible to model the neural dynamics of more conceptual actions and abstract goals, while still grounding them in a sensorimotor surface.

Of course, in a hierarchical setting like this, it is not easy to determine at which level signatures of habituation or perseveration should arise. They may arise at the level of goal, strategy, action, or even at multiple levels. To gain an insight into the dynamics of goals, experimental data probing the dynamics of goals, strategies and actions may be beneficial to test if the chosen approach is feasible.

A clear limitation of the model is that it can only choose from a very limited set of goals. These goals and the set of actions possible to achieve them are predefined by the scenario and the task. How humans set themselves a task and structure their cognitive resources, such as attention, memory, or reasoning in its service is still far beyond the capabilities of the presented model. Cognitive abilities like these are linked to a more general cognitive control (Botvinick and Braver, 2015), which lies outside of what neural dynamic DFT models can currently capture.

An often held view is that these cognitive resources are directed towards problems that require our conscious control, while already automated actions can proceed without conscious awareness (Peterson, 2002). This is related to the concepts of flow and intrinsic motivation, in which humans focus their cognitive resources on the acquisition of skills that lie just outside their current capabilities (Oudeyer and Kaplan, 2007). A possible avenue of investigation could explore how actions that initially require active top-down control (e.g. by recalling beliefs) can be automated and subsequently used for more abstract goals. For DFT, a top-down habit formation mechanism could be appropriate (Kazerounian et al., 2012). This could open the door for DFT models with a more open ended development (Colas et al., 2020; Oudeyer, Kaplan, and Hafner, 2007).

Bibliography

Aerdker, Sophie, Jing Feng, and Gregor Schöner (2022). "Habituation and Dishabituation in Motor Behavior: Experiment and Neural Dynamic Model". In: *Frontiers in Psychology* 13, pp. 1–8. DOI: 10.3389/fpsyg.2022.717669.

Arrington, Catherine M. and Gordan D. Logan (2004). "The cost a voluntary task switch". In: *Psychological Science* 15, pp. 610–615. DOI: https://doi.org/10.1111/j.0956-7976.2004.00728.x.

Bicho, E., P. Mallet, and G. Schöner (1998). "Using attractor dynamics to control autonomous vehicle motion". In: *IECON '98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.98CH36200)*. Vol. 2, 1176–1181 vol.2. DOI: 10.1109/IECON.1998.724266.

Botvinick, Matthew and Todd Braver (2015). "Motivation and Cognitive Control: From Behavior to Neural Mechanism". In: *Annual Review of Psychology* 66, pp. 83–113. DOI: 10.1146/annurev-psych-010814-015044.

Butz, Martin V., Oliver Sigaud, and Pierre Gérard (2003). *Anticipatory Behavior in Adaptive Learning Systems Foundations, Theories, and Systems*. Berlin: Springer. ISBN: 3-540-40429-5.

Colas, Cédric et al. (2020). "Intrinsically Motivated Goal-Conditioned Reinforcement Learning: a Short Survey". In: *CoRR* abs/2012.09830.

Dineva, Evelina and Gregor Schöner (2018). "How infants' reaches reveal principles of sensorimotor decision making". In: *Connection Science* 30.1, pp. 53–80. DOI: 10.1080/09540091.2017.1405382. eprint: https://doi.org/10.1080/09540091.2017.1405382. URL: https://doi.org/10.1080/09540091.2017.1405382.

Erlhagen, Wolfram and Gregor Schöner (2002). "Dynamic field theory of movement preparation". In: *Psychological Review* 109, pp. 545–572. DOI: https://doi.org/10.1037/0033-295X.109.3.545.

Grieben, Raul and Gregor Schöner (2021). "A neural dynamic process model of combined bottom-up and top-down guidance in triple conjunction visual search". In: *Proceedings of the 43nd Annual Conference of the Cognitive Science Society*. Ed. by Tecumseh Fitch et al. Cognitive Science Society.

Hebb, Donald Olding (2005). *The organization of behavior: A neuropsychological theory*. Psychology Press.

Herbort, Oliver and Martin V. Butz (Nov. 2012). “Too Good to be True? Ideomotor Theory from a Computational Perspective”. In: *Frontiers in Psychology* 3. DOI: 10.3389/fpsyg.2012.00494.

Hommel, Bernhard et al. (2001). “The Theory of Event Coding (TEC): A framework for perception and action planning”. In: *Behavioral and Brain Sciences* 24, pp. 849 –837. DOI: https://doi.org/10.1017/S0140525X01000103.

Jarecki, Jana B., Jolene H. Tan, and Mirjam A. Jenny (July 2020). “A framework for cognitive process models”. In: *Psychonomic Bulletin Review* 27, pp. 1218–1229. DOI: https://doi.org/10.3758/s13423-020-01747-2.

Kazerounian, Sohrob et al. (Oct. 2012). “Autonomous Reinforcement of Behavioral Sequences in Neural Dynamics”. In: *Proceedings of the International Joint Conference on Neural Networks*. DOI: 10.1109/IJCNN.2013.6706877.

Kornblum, S., T. Hasbroucq, and A. Osman (1990). “Dimensional overlap: Cognitive basis for stimulus-response compatibility–A model and taxonomy”. In: *Psychological Review* 97, 253–270. DOI: https://doi.org/10.1037/0033-295X.97.2.253.

Kunde, Wilfried, Joachim Hoffmann, and Philipp Zellmann (2002). “The impact of anticipated action effects on action planning”. In: *Acta Psychologica* 109, pp. 137–155. DOI: https://doi.org/10.1016/S0001-6918(01)00053-1.

Lomp, Oliver et al. (2016). “Developing Dynamic Field Theory Architectures for Embodied Cognitive Systems with cedar”. In: *Frontiers in Neurorobotics* 10. ISSN: 1662-5218. DOI: 10.3389/fnbot.2016.00014. URL: https://www.frontiersin.org/article/10.3389/fnbot.2016.00014.

Olivier, Michel (2004). “Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation”. In: *International Journal of Advanced Robotic Systems* 1.1, p. 5. DOI: 10.5772/5618. eprint: https://doi.org/10.5772/5618. URL: https://doi.org/10.5772/5618.

Oudeyer, Pierre-Yves and Frederic Kaplan (2007). “What is Intrinsic Motivation? A typology of Computational Approaches.” In: *Frontiers in neurorobotics* 16.2. DOI: 10.3389/neuro.12.006.2007.

Oudeyer, Pierre-Yves, Frederic Kaplan, and Verena V. Hafner (2007). “Intrinsic Motivation Systems for Autonomous Mental Development”. In: *IEEE Transactions on Evolutionary Computation* 11.2, pp. 265–286. DOI: 10.1109/TEVC.2006.890271.

Papies, Esther and Lawrence Barsalou (Dec. 2014). "Grounding desire and motivated behavior: A theoretical framework and review of empirical evidence". In.

Perone, Sammy and John P. Spencer (2013). "Autonomy in Action: Linking the Act of Looking to Memory Formation in Infancy via Dynamic Neural Fields". In: *Cognitive Science* 37, pp. 1–60. DOI: 10.1111/cogs.12010.

Peterson, Jordan B (2002). *Maps of meaning: The architecture of belief*. 1st ed. Routledge. ISBN: 978-0415922227.

Richter, Mathis, Yulia Sandamirskaya, and Gregor Schöner (2012). "A robotic architecture for action selection and behavioral organization inspired by human cognition". In: vol. IEEE, pp. 2457–2464.

Sandamirskaya, Yulia, Mathis Richter, and Gregor Schöner (Sept. 2011). "A neural-dynamic architecture for behavioral organization of an embodied agent". In: vol. 2, pp. 1 –7. DOI: 10.1109/DEVLRN.2011.6037353.

Schöner, Gregor (2020). "The Dynamics of Neural Populations Capture the Laws of the Mind". In: *Topics in Cognitive Science* 12, pp. 1257–1271. DOI: 10.1111/tops.12453.

Schöner, Gregor, John Spencer, and Institut für Neuroinformatik DFT Research Group (2015). *Dynamik Thinking: A Primer on Dynamic Field Theory*. Oxford University Press.

Schöner, Gregor and Esther Thelen (2006). "Using Dynamic Field Theory to Rethink Infant Habituation". In: *Psychological Review* 113, pp. 273–299. DOI: 10.1037/0033-295X.113.2.273.

Searle, John R. (1983). *Intentionality: An Essay in the Philosophy of Mind*. Cambridge University Press. DOI: 10.1017/CBO9781139173452.

Stock, Armin and Claudia Stock (May 2004). "A short history of ideo-motor action". In: *Psychological research* 68, pp. 176–88. DOI: 10.1007/s00426-003-0154-5.

Swindale, Nicholas V. (2000). "How Many Maps are there in Visual Cortex?" In: *Cerebral Cortex* 10, pp. 633–643. DOI: 10.1093/cercor/10.7.633.

Tekulve, Jan and Gregor Schöner (Aug. 2020). "A neural dynamic network drives an intentional agent that autonomously learns beliefs in continuous time". In: *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1–1. DOI: 10.1109/TCDS.2020.3013768.

Tekülve, Jan (2021). "A Neural Process Model of Intentionality Implemented on an Autonomous Robot". PhD thesis. Ruhr-Universität Bochum.

Tekülve, Jan et al. (2019). "Autonomous Sequence Generation for a Neural Dynamic Robot: Scene Perception, Serial Order, and Object-Oriented Movement". In: *Frontiers in Neurorobotics* 13. ISSN: 1662-5218. DOI: 10.3389/fnbot.2019.00095. URL: https://www.frontiersin.org/article/10.3389/fnbot.2019.00095.

Thelen, Esther and Gregor Schöner (2001). "The dynamics of embodiment: A field theory of infant perseverative reachings". In: *Behavioral and Brain Sciences* 24, pp. 1–86.

Vinyals, O. et al. (2019). "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: *Nature* 575, 350–354. DOI: https://doi.org/10.1038/s41586-019-1724-z.