FACULTY OF PSYCHOLOGY AND EDUCATIONAL SCIENCES

FLEXIBLE COMMUNICATION THROUGH NEURAL SYNCHRONIZATION

A COMPUTATIONAL MODELLING APPROACH

Word count: 19, 785

Pieter Verbeke
Student number:01206180

Supervisor(s): Prof. Dr. Tom Verguts

A dissertation submitted to Ghent University in partial fulfilment of the requirements for the degree of Master of experimental psychology

Academic year: 2016 - 2017

GHENT UNIVERSITY

Preface

After five years, I finish my education as experimental psychologist with this master thesis. Personally, I am quite proud of the result but I should acknowledge that this result would not have been possible without the great support of my supervisor, Tom Verguts. He guided me in my first steps into a completely new field. Because he was willing to meet with me every two weeks I could get a lot of feedback and I could continue working on a quite good pace. I learned a lot from him and therefore I would like to express a special thanks to professor Verguts. I am very excited to, in the future, continue working on this interesting project with him as my mentor.

I also want to express some gratitude to the members of professor Verguts' lab who showed interest in my work during my presentation in one of the lab meetings. Last but not least, I would also like to thank my girlfriend who had the patience to always listen to me when I was complaining or having a brain storm session aloud.

Abstract

We propose a computational model of cognitive control that combines the ideas of binding by synchronization (Gray & Singer, 1989) and learning through the Rescorla-Wagner rule. For the binding part, we implemented the idea of rapid phase synchronization and desynchronization through the principle of binding by random bursts (Verguts, 2017). While oscillatory synchronization leads to a fast association of two units, desynchronization causes a fast reset of this association. Additionally, we implemented an Actor-Critic structure in the model. Here, the idea of the Critic is adapted from the reward value prediction model (Silvetti, Seurinck, & Verguts, 2011) and the Actor is a combination of two prefrontal areas, the LFC and the caudal part of the ACC. We found that, because of the Actor-Critic structure, the model is able to determine on its own which processing areas have to be bound or unbound. By comparing the flexibility of models using binding by synchronization, learning or both, we illustrated how phase synchronization, on top of learning, can provide an additional way to very flexibly alter communication in the brain. We found that this additional flexibility is especially helpful when the learning rate is slow.

Keywords: cognitive control, phase synchronization, learning, computational modelling

Table of contents

Working memory

The human brain is capable of great things. Our brain accounts for abilities such as thinking, reasoning, planning; and more generally, quickly adapting to completely new situations. To do this, the brain must be able to keep certain information in store and process it, a capacity that is generally referred to as working memory.

A highly influential model of working memory is the one of Baddeley and Hitch (1974; Figure 1). These authors assume that working memory consists of three different components; (a) the visuospatial loop, (b) the phonological loop and (c) the central executive. While the main function of the visuospatial- and phonological loop is to keep information in store, the central executive has a more prominent role in their model. It is basically the central executive that puts the "working" in working memory. Over the years, the function of the central executive has received many names, such as executive control or cognitive control. However, no matter which name it received, it is generally agreed that the role of the central executive is threefold. That is, (a) binding and coordinating information from different sources, (b) monitoring selective attention and behavioral inhibition and (c) shifting between different task strategies. How the central executive does this exactly was not specified by Baddeley and Hitch (1974) but has been extensively studied in the last few years (for a review see Miller, 2013).

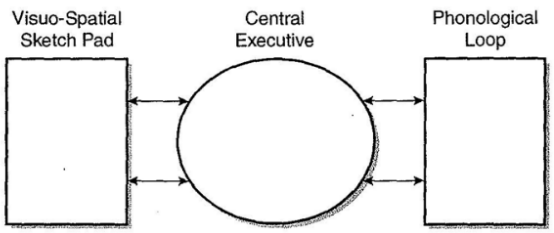

Figure 1. An overview of the working memory model of Baddeley and Hitch (1974).

Recent brain imaging studies point to the prefrontal cortex (PFC) as the strongest candidate to perform the role of central executive (e.g. Braver et al., 1997; Koechlin, Ody, & Kouneiher, 2003). Structurally, the PFC can be described as a collection of strongly interconnected areas that are in turn connected to a huge number of other cortical areas. Receiving input from almost every cortical area (Miller & Cohen, 2001), the PFC seems

ideal to bind pieces of information together. Functionally, Miller and Cohen (2001) proposed that the role of the PFC is to guide neural activity flow along task-relevant pathways in more posterior and/or subcortical areas. Thus, the PFC modulates general brain activity by sending biasing signals to other areas; inhibiting task-irrelevant neurons and exciting, or selecting, task-relevant neurons.

Computational models

Although brain imaging studies can give us useful information about all kinds of cognitive processes, their results do not provide us a complete, comprehensive view of all aspects of a cognitive process. As a complementary research strategy, researchers often propose a computational model. In such a computational model, a hypothetical description of a cognitive process is made. This model should ideally be able to connect with as much behavioral and neural data as possible and perhaps create some new hypotheses for future research. The better the model fits empirical data in a parsimonious manner, the higher the likelihood that the processes it describes, are close to reality. Hence, the main requirements of a good computational model are a good fit to empirical data and parsimony (measured, e.g., by its number of parameters).

Several computational models, describing working memory processes, have adopted the idea of the PFC as a modulator of neuronal activity. This idea of one area sending biasing signals to other brain structures has proven to be an effective way for solving tasks where high cognitive control is needed, such as for example a Stroop task (Cohen, Dunbar, & McClelland, 1990; Verguts & Notebaert, 2008). In these classical models, an activated input unit (neuron) typically sends its message (action potential) down its connections (axons and synapses), which have a specific strength or "weight", to all units to which it is connected. Those receiving units combine all different inputs they receive and again give a certain output activation. This output activation is generally described as

$$a_j = \sum a_i w_{ij} \quad (1),$$

where a_j represents the activation level of the output unit, a_i is the activation level of an input unit and w_{ij} indicates the weight value or synaptic strength from the input- to the

output unit. Learning is generally done by changing the weight strength. An increase in the weight value implies that, next time, the activation of the input unit will have more influence on the activation level of the output unit. Thus, the amount of information a unit receives is in these models reflected in the unit's activation level. Biologically, the activation of a unit, or neuron, corresponds to a neuron's firing rate. Therefore, we will further recall to this kind of model units as rate code neurons.

These classical computational models have provided a lot useful insights into cognitive processes but they only focus on the neuron's firing rate (e.g, Cohen et al., 1990; Holroyd & Coles, 2002; Verguts & Notebaert, 2008). Another characteristic of the firing of a neuron, which is in most models ignored, is the phase, or timing at which the neuron fires. In recent years, cumulating evidence indicates that relations in the phase of neuronal firing, especially phase synchronization, are also crucially involved in all kinds of cognitive processes (Cavanagh & Frank, 2014; Engel, Fries, & Singer, 2001; Fell & Axmacher, 2011). Thus, accounting for neural firing phase, with so-called phase code units, in a computational model might allow the model to connect with an even broader range of neuropsychological data. We elaborate on this important property in the next paragraph.

Neural synchronization

In recent years, a special interest has risen in the study of synchronized processes (Strogatz, 2003). A lot of synchronized phenomena seem to occur in nature. In East-Asia for example, there have been observations of fireflies blinking on and off simultaneously. Also, groups of birds flap their wings simultaneously. Additionally, the well-known incident on the opening days of the Millennium Bridge in London shows that, like metronomes, humans tend to synchronize their steps when walking on the same (unstable) platform. Synchronization has also been found to be important at more basic biological levels such as the working of heart cells and of course in brain cells (for a full review see Strogatz, 2003). Like fireflies blink on and off, neurons (brain cells) have the property to fire and then go back to their original state. When combined in groups, these neurons show a wave, or oscillation, over time which reaches its top when a lot of neurons in the group are firing and its trough when none of them are firing. It has been suggested that the level of synchronization between these oscillations might have important influences on cognitive

functioning (e.g., Fries, 2005, 2015; Gray & Singer, 1989). In the next section, we will describe recent theories about the function of oscillatory synchronization across different neuronal groups.

A first theory describing the importance of synchronization in neural firing is the Binding-By-Synchronization (BBS) hypothesis (Gray & Singer, 1989). This theory argues that neuronal firing phase provides a tag that binds those neurons that represent the same perceptual object. So, neurons that show synchronized firing patterns form a group representing one cognitive concept or perceptual object. This theory introduces the idea that information can be coded in the phase of neuronal firing. Interestingly, BBS could provide two advantages. First, it binds certain pieces of information together. Second, as was already proposed by Miller (2013), it can also increase the representational capacity of the brain since neurons can send different kinds of information depending on their phase of firing.

This idea was later on extended by Pascal Fries (2005, 2015), who recently described the communication-through-coherence (CTC) hypothesis. This hypothesis states that oscillatory synchronization between two different neuronal groups has important consequences for the efficiency of their communication. The CTC-framework relies on two basic assumptions. First, as described above, activated neuronal groups have the intrinsic property to oscillate. Second, these oscillations constitute rhythmic modulations in neuronal excitability that affect both the likelihood of firing output and the sensitivity to synaptic input. In order to fully comprehend this second assumption, consider two neuronal groups, each represented as a room. Suppose each room has a door that is constantly opening and closing, with the doors (and the rooms) facing each other. Obviously, you can only get out, or in, the room when the door is open. Moreover, when you must go from one room to the other, this will be easiest when the two doors are opening and closing in synchrony. This would mean that the doors are open and closed at the same time. So, just waiting until the moment they are both open will be sufficient to step from one room into the other.

Suppose now there is a small hallway between the two rooms. In this case, the optimal situation would be if the two doors are opening and closing at the same frequency but with a small delay between them which equals the time you need to cross the hallway.

Similarly, the CTC-theory implies that information transfer between two neuronal groups can only be effective if their oscillations are synchronized. Again, the optimal situation would be when there is a small delay in their phase of firing, equaling the time needed for neural information to travel down the axons and synapses (Bastos, Vezoli, & Fries, 2014). Thus, the CTC principle implies that even though anatomical connections between the two groups could remain intact at a fast time scale, the efficiency of communication can be modulated by changes in neural synchronization.

The CTC-principle also receives empirical support. In Figure 2 (Fries, 2009), we see a schematic illustration of spike recordings in neuronal groups of a cat's visual cortex (Womelsdorf et al., 2007). First, in Figure 2a, the general CTC-principle is illustrated. Two neuronal groups of a lower visual area and one group of a higher visual area are shown. Information should be transferred from the lower area to the higher area. However, according to the CTC-theory, this is only possible for group A because this one shows synchronized oscillations with C and B is not. The data, shown in Figure 2c, revealed that the power correlation between two groups, which indexed the similarity of their activation levels, depended on the phase relation of their oscillations. This indicates that the efficiency of the communication between two neuronal groups indeed depends on their phase relation. In Figure 2b, we see that, as expected, the power correlation was optimal when there was a small delay between the oscillations. In sum, the principle of CTC receives support from empirical data, showing that a neuronal group can only transfer its own activation to the other neuronal group when their relative phase is optimal.

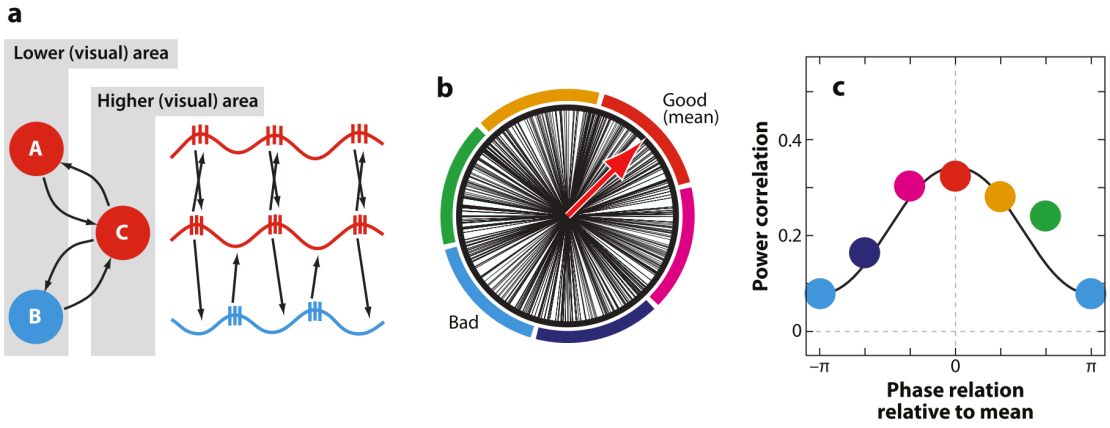

Figure 2 (Fries, 2009). An illustration of the CTC-principle with empirical data (Womelsdorf et al., 2007). *Figure 2a, shows a schematic overview of what was recorded. That is, oscillatory activity in a cat visual cortex in lower and higher visual areas. The CTC-principle states that only neuronal groups A and C can communicate effectively since their oscillations are synchronized. The lines in Figure 2b, show recorded relative phases for two neuronal groups across a time span of 250 ms. This data is divided into 6 bins which are given by the colors. Red represents the mean relative phase across the experiment. Figure 2c, shows the relation of power correlation and phase relation relative to the mean. It is shown that if relative phase is at the mean there is an optimal correlation of power between two neuronal groups.*

The added value of phase code

In the current study, we claim that a computational model might benefit from implementing phase code neurons in several ways. First, as already mentioned, phase code helps us to connect with a broader range of data (Verguts, 2017). Therefore, we could reach a deeper understanding of the underlying neural processes of behavior. Additionally, previous modelling studies, have demonstrated that oscillations, and the coherence between them, are computationally very powerful for modelling several brain areas and processes such as the olfactory bulb (Li & Hopfield, 1989), memory in the hippocampus (Borisyuk & Hoppensteadt, 1998; Scarpetta, Li, & Hertz, 2002) and visual attention (Jensen, Bonnefond, & VanRullen, 2012). Moreover, the model of Verguts (2017) already demonstrated the added value of phase code processes in a computational model of cognitive control.

Also theoretically, the BBS- (Gray & Singer, 1989) and CTC-frameworks (Fries, 2005, 2015) show attractive characteristics to describe cognitive control processes. In the previous section, we already discussed how the BBS-theory describes that neural synchronization can mechanistically account for binding (and coordinating) different pieces of information. Additionally, the CTC-theory demonstrated how neural synchronization can account for selective communication within the brain. A principle that might be very important in monitoring selective attention (Fell, Fernández, Klaver, Elger, & Fries, 2003). This means that only one aspect of the central executive functions remains to be discussed. Namely, flexibly shifting between different task strategies.

In order to flexibly shift between different task strategies, a person, or model, has to be able to flexibly unlearn and learn previous and new behavior. In the brain this is done

by altering the efficiency of communication between neuronal units. The efficiency of communication between two brain areas can be altered in two ways. First, the strength of the anatomical connection between two areas can change. This change is in general rather slow and gradual, making it ideal to detect profound regularities in the environment. Second, there can be a change in the time of neuronal firing, leading to synchronous or asynchronous oscillations in both areas. This change can occur very fast, allowing the brain to react to sudden changes in the environment, such as for example the need of a new task strategy. Because slow (anatomical) learning and fast (synchronization) learning rely on two different processes, both types of learning can happen independently. This is very convenient since it implies that a change in task strategy can occur without changes in anatomical connections. This means you can flexibly change between task strategies without any loss of information. The advantage of having a separate slow and fast learning system was already demonstrated by previous modelling studies (O'Reilly & Norman, 2002).

Although phase synchronization and anatomic changes are in general independent processes, it has also been suggested that phase synchronization can enhance plasticity of anatomical, synaptic connections (Axmacher, Mormann, Fernández, Elger, & Fell, 2006; Jutras & Buffalo, 2010). In some cases, phase synchronization processes might thus interact with learning. Specifically, a change in synchronization levels between two units and the consequences for the communication between these units might have important influences in Hebbian learning (Hebb, 1949). This type of learning is generally described by the statement; "if it fires together, it wires together", meaning that if two neurons are active at the same time, the connectivity between these neurons increases. The simplest implementation of Hebbian learning is

$$\Delta W_{ij} = \delta * a_i * a_j \qquad (2).$$

Here, ΔW_{ij} indicates the change in synaptic strength, δ is a learning rate parameter and a_i and a_j are respectively the in- and output neurons. Because, according to the CTC-theory (Fries, 2005, 2015), communication between synchronized units is more efficient than communication with desynchronized units, phase synchronization enhances the transfer of

activation levels from the input- to the output unit, causing the output unit to reach higher activation levels. This would lead to faster associative learning, because a stronger activation of the output unit would lead to a greater change in weight strength in the Hebbian learning rule (equation 2). The fact that learning performance is influenced by phase synchronization is also empirically supported by the finding that memory performance is better when there was more precise phase synchronization in the learning stage (Axmacher et al., 2006).

A last characteristic making it attractive to implement phase code in a computational model is the finding that phase code interactions are not limited to mere short range communication. Empirical research suggests that interactions between oscillatory signals in the brain are not limited to specific areas (Nigbur, Cohen, Ridderinkhof, & Stürmer, 2011; Womelsdorf, Johnston, Vinck, & Everling, 2010) or specific frequencies (Voloh, Valiante, Everling, & Womelsdorf, 2015). For instance, it has been suggested that a lower-frequency band rhythm, such as alpha (~ 8-12 Hz) or theta (~4-8 Hz), coming from an area like the PFC or the thalamus, modulates the strength of gamma-band (~30-70 Hz) synchronization in other cortical areas (Lisman & Jensen, 2013; Verguts, 2017). Thus, phase code processes seem to be present within and across all brain areas.

In sum, neural synchronization seems an attractive way to implement cognitive control in a computational model. It provides a computationally powerful mechanism for binding different pieces of information, it proposes a way for selective communication and it seems very flexible. Additionally, phase code processes seem not to be limited to specific frequencies or brain areas.

The model

A previous study of Verguts (2017) demonstrated how the implementation of phase code processes in a computational model allows to connect with a broad range of empirical findings, both neural and behavioral, during a Stroop task. However, this model was limited in several ways. First, the model was explicitly told what to do. That is, synchronizing oscillations in the units of the relevant task dimension with the response units and ignoring the irrelevant task dimension. The model did not learn itself which

dimensions of the stimuli were relevant. Second, the model did not illustrate the flexibility of phase code communication. It only used synchronization, which binds two units together and increases the efficiency of their communication but it was not able to unbind these units again. It did not desynchronize. Empirical findings indicate that also desynchronization might be important for cognitive processes (Rodriguez et al., 1999). In the study of Rodriguez and colleagues (1999), a period of strong desynchronization marked the transition from the moment of perception of an ambiguous visual stimulus to the execution of a motor response. Indirect evidence for the importance of desynchronization processes can also be derived from the fact that patients with Parkinson disease, which is often characterized by an extreme cognitive rigidity, show abnormally synchronized oscillatory activity (Hammond, Bergman, & Brown, 2007). So, beside synchronization also phase desynchronization might be important, in the sense that it can cause a reset, disrupting ongoing processes and allowing the initiation of other processes.

Hence, a first problem we addressed in the current study, was how to achieve synchronization and desynchronization of oscillations in different neuronal groups. Previous research suggested that synchronization between two oscillating units can be achieved by adding random but positively correlated bursts to the two units (Springer & Paulsson, 2006; Verguts, in press; Zhou, Chen, & Aihara, 2005). Similarly, we hypothesized that desynchronization could be accomplished by adding random, negatively correlated bursts. Thus, we argue that oscillations of two different units can be bound together by positively correlated bursts and unbound by negatively correlated bursts. An illustration of this synchronization by random bursts is given in Figure 3. In this figure, we see two vectors (black lines) which have a phase difference of $\pi/2$. Two positively correlated hits are given (green dashed vectors), leading to a reduced phase difference, and thus an increased synchronization. Two negatively correlated bursts on the other hand (dashed red vectors), lead to a greater phase difference or desynchronization.

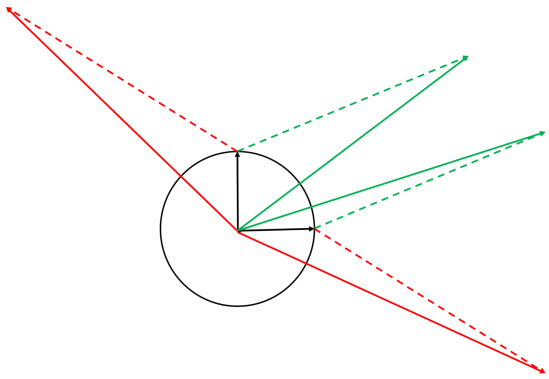

Figure 3. An illustration of the binding (and unbinding) by random bursts principle. *Two initial vectors (black lines) are desynchronized with a phase difference of π/2. When two positively correlated hits are given, (the dashed green lines) this leads to a decrease in the phase difference, or synchronization, between the resulting vectors (full green lines). When the two vectors receive negatively correlated hits (dashed red lines) this leads to an increase of the phase difference or desynchronization of the resulting new vectors (full red lines).*

A second problem we addressed is that the model (Verguts, 2017) was told what to synchronize. In order to let the model discover what to synchronize on its own, we used the principle of Reinforcement Learning (RL; for an introduction see Sutton & Barto, 1998). In RL, the model receives a reward signal when it is doing something right. In the absence of reward, the model will thus know it must change its behavior and when it receives a reward, it will reinforce its behavior. One way to implement this kind of learning in a computational model, is by introducing an Actor-Critic structure (e.g. Houk, Adams, & Barto, 1995). Here, the Critic uses the reward information to evaluate current behavior. This evaluation is then used by the Actor to adjust processes in the other parts of the model. The idea of the Critic is for the current model adapted from the reward value prediction model (RVPM; Silvetti et al., 2011). In the current model, the rostral part of the anterior cingulate cortex (ACC) functions as Critic by comparing reward expectations with actual received rewards. The Control module will consist of the lateral frontal cortex (LFC) and the caudal part of the ACC. These two areas will together give the burst in order to alter synchronization levels in the other parts of the model.

In the following sections, we will gradually build up a computational model in order to accomplish our goal of introducing a model of cognitive control that describes interactions between phase and rate code, shows a great flexibility in learning and adapting

to different task demands, and is able to connect with a rich amount of behavioral and neuronal data. Instead of only relying on synaptic weight changes to accomplish learning, the current model will describe how altering the level of synchronization between two oscillating neuronal groups can also alter communicative efficiency in the brain. Complete synchronization of two units should enhance information transfer between them while complete desynchronization should inhibit information transfer. Additionally, the model uses RL-principles to discover on its own what to learn.

In a first step, we addressed a first problem of the model of Verguts (2017). That is, the fact that the model could only synchronize but not desynchronize. In the second step, we demonstrate the CTC-principle by describing interactions between phase- and rate code neurons. In the third step, we address a second problem of the previous model of Verguts (2017), by implementing an Actor-Critic structure in the model. This allowed the model to learn what to synchronize or desynchronize on its own. In a fourth step, we introduced a reversal learning task (e.g. Clark, Cools, & Robbins, 2004; Izquierdo & Jentsch, 2012) to the model to demonstrate the flexibility of phase code learning. In the fifth and last step, we added basic anatomical weight learning to the model to demonstrate how the processes of anatomical learning and phase synchronization interact.

Simulation 1: Synchronizing and desynchronizing

Here, we tried to answer the question of how synchronization and desynchronization between two units could be accomplished. As mentioned before, we hypothesized that desynchronization could be accomplished by adding random negatively correlated bursts (see Figure 3). Thus, oscillations of two different units can be bound together by positively correlated bursts and unbound by negatively correlated bursts.

The model of Simulation 1, adopts the idea of Verguts (2017), by implementing the ideas of cross-frequency (Voloh et al., 2015) and cross-area coupling (Nigbur et al., 2011; Womelsdorf et al., 2010) to describe how two prefrontal brain areas, the ACC and LFC, work together to modulate synchronization in more posterior cortical processing areas. In this way, the model is also complementary with the notion of the PFC as modulator of neuronal activity in other processing areas (Miller & Cohen, 2001).

The model

The model in Simulation 1 consists of two modules; a Control and a Processing module. Its purpose was to explore how we could accomplish synchronization and desynchronization between two oscillating units. The Processing module contains two units which could, as an illustration, be presented as one stimulus (S_1) and one response (R_1). The Control module consists of the LFC and ACC which, as we will describe later, together determine how and when oscillations in the Processing module become synchronized or desynchronized. A schematic overview of this model is given in Figure 4.

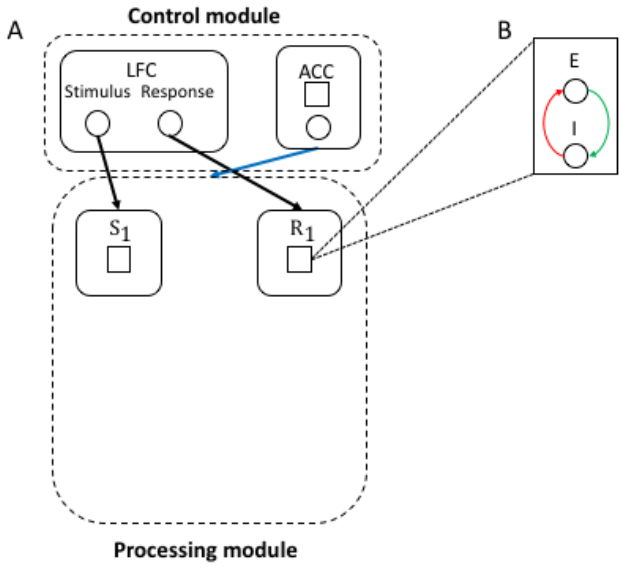

Figure 4. Model sketch. *In Figure 4A, an overview of the model is given. The Control module contains the lateral frontal cortex (LFC) and anterior cingulate cortex (ACC). The Processing module consists of two units. The S-unit could, as illustration, represent a stimulus and the R-unit a response. Circles represent rate code neurons and squares represent phase code units which consist of one excitatory-inhibitory neuronal pair as illustrated in Figure 4B. Here, E represents the excitatory neuron and I the inhibitory neuron. The red arrow thus indicates an inhibitory weight and the green arrow an excitatory weight.*

The Processing module.

Oscillations are in the current model represented by the interaction of a neuronal pair in which one neuron is excitatory (E_i) and one is inhibitory (I_i). Excitatory neurons are updated by

$$E_i(t+dt) = E_i(t) - C * I_i(t) - D * J(r > r_{min}) * E_i(t) + H_i(t) \qquad (3).$$

and inhibitory neurons are updated by

$$I_i(t + dt) = I_i(t) + C * E_i(t) - D * J(r > r_{min}) * I_i(t) \quad (4).$$

The two neurons are thus coupled by a parameter *C*. The strength of this coupling determines the frequency of the oscillations, $C/(2\pi)$ (Li & Hopfield, 1989; Verguts, 2017). In the Processing module, we aimed to represent oscillations in the gamma-frequency band (30-70 Hz). So, we chose a value for *C* which corresponds to a frequency of 40 Hz. The variable *t* refers to a certain timepoint and *dt* refers to a timestep of $3*10^{-4}$ sec. The radius ($r=E^2+I^2$) of the oscillations are attracted towards the value $r_{min}=1$. This is implemented by the term $D*J(r>r_{min})*E_i(t)$ in equation 3 and $D*J(r>r_{min})*I_i(t)$ in equation 4. Here, J(.) is an indicator function, returning 1 when the radius gets higher than the value of r_{min} and 0 otherwise. The damping parameter, *D*=.01, determines the strength of this attraction.

In sum, because of their coupling, activation in both neurons oscillates. The frequency of this oscillation is determined by the strength of the coupling. When the radius becomes higher than r_{min}, oscillations will receive an extra inhibitory signal which will dampen its amplitude towards r_{min} again. The excitatory neurons of the Processing module additionally receive a burst, or hit term $H_i(t)$,

$$H_i(t) = LFC_i * ACC(t) * U(t) \quad (5).$$

This equation implies that the LFC and ACC together determine the burst signal that is received by the excitatory phase code neurons of the Processing module. The variable *U(t)* is a standardized-Gausian variable. So, the hit signal that is determined by the ACC and LFC is multiplied with a random number sampled from the standard normal distribution with mean 0 and variance 1. This creates some random fluctuation in the hits even if the basic signal remains the same. Nevertheless, when the basic signal remains the same, different hits will correlate over time. This correlation should lead to synchronization or desynchronization.

The Control module.

As the name states, the Control module controls information transfer in the Processing module. Like the model of Verguts (2017), the Control module consists of two parts. In the current model, these two parts are the LFC and the caudal part of the ACC.

The ACC.

The modelled ACC consists of one phase code pair (E_{ACC}, I_{ACC}) and a rate code neuron (ACC). The phase code neurons obey the same updating rules as given by equation 3 and 4. In the ACC, r_{min}=.8 and the value of C is changed such that oscillations are at a 5Hz (theta-)frequency. The ACC does not receive a hit, so H_{ACC}=0. The activation in the rate code neuron of the ACC is given by

$$ACC(t) = Be\left(\frac{1}{1+e^{(-10(E_{ACC}(t)-.95)}}\right) \quad (6).$$

This equation represents a Bernoulli process which is 1 with a certain probability. The probability, $\frac{1}{1+e^{(-10(E_{ACC}(t)-.95)}}$ is a sigmoid function which has its greatest value when the E_{ACC}(t) is near its top and its amplitude is sufficiently strong.

In sum, activation in the phase code neuronal pair (E_{ACC}, I_{ACC}) oscillates. Crucially, every time the oscillation of the E-neuron reaches its top the probability of a hit becomes high. Since oscillations in the ACC have a theta rhythm, the ACC gives rhythmic (at approximately a 5 Hz frequency) hits to the oscillations in the Processing module. So, the ACC determines the 'when' of the hit signal. As long as the value of the LFC-unit does not change, these hit signals remain correlated.

In previous studies, high theta-power in the ACC has been linked to high cognitive control (e.g., Cavanagh & Frank, 2014; Womelsdorf et al., 2010). This is implemented here by the fact that when theta-power in the ACC is high, more bursts will be given. This means that synchronization or desynchronization will be more effective when the amplitude of the oscillations in the ACC is high. When theta-power is low, no hits will be given.

The LFC.

The LFC consists of two rate code neurons that send an eligibility signal to the Processing module units with a value of $LFC_i \in \{0,1,-1\}$. Units that receive a signal (1 or -1)

will receive a hit, $H_i(t)$. Units that do not receive the signal (LFC_i=0) will not be eligible to receive the hit. LFC_i has no time index t because it is constant across the entire trial. We argue that in units that receive the same signal, a hit will lead to synchronization because the hits are positively correlated over time. In units receiving an opposite signal (1 and -1), a hit will lead to desynchronization because their hits are negatively correlated over time. Importantly, this means that in transition from synchronization to desynchronization between two areas only the eligibility signal to one of the areas should be changed. The signal the other area receives can remain the same. The LFC determines thus the 'what' of the hit signal. This is consistent with findings indicating that the LFC holds the representation of task demands (Braver et al., 1997; Mac Donald, Cohen, Stenger, & Carter, 2000).

Method

10 replications of 30 trials of 2500 timesteps each were simulated. In the first 10 trials, $LFC_{S_1}=LFC_{R_1}=0$. So, the hit signal, H, was zero during all 10 trials. In the following 10 trials $LFC_{S_1}=LFC_{R_1}=1$. In these trials, the units in the Processing module did receive hits. Because the LFC signals for both Processing units, S_1 and R_1, were the same, they received positively correlated hits each time the Bernoulli process in equation 6 returned 1. The combination of hits across time implements the principle of binding by random bursts (Springer & Paulsson, 2006; Verguts, 2017; Zhou et al., 2005). In the last 10 trials, we set $LFC_{S_1}=1$ and $LFC_{R_1}=$-1. Because of the opposite sign of the LFC values, the Processing units received negatively correlated hits. We hypothesized that this would lead to desynchronization of the oscillations in the two Processing units. Starting points of the phase code neurons, in both the Processing- and Control module, were values between 0 and 1, randomly sampled from the uniform distribution. Each neuron, excitatory and inhibitory, of a phase code pair had the same starting value. For unit S_1, this starting point was multiplied by -1 to make sure that oscillations in S_1 and R_1 were not synchronized from the start. For all trials following the first trial, the oscillations started at the same point they ended in the previous trial.

As a measure of phase synchronization between two oscillations, we used the dissimilarity function (Rosenblum, Pikovsky, & Kurths, 1997) at phase lag zero. This function is given by

$$S_{E_1E_2} = \frac{\langle (E_2(t) - E_1(t))^2 \rangle}{\sqrt{\langle E_1^2(t) \rangle \langle E_2^2(t) \rangle}} \quad (7),$$

where the brackets ⟨ ⟩ indicate the average over time. This dissimilarity function reaches its minimum zero when the two signals (E_1, E_2) are completely synchronized over time and has no defined upper bound. The function was computed across all time points within one trial. This was done for all trials separately and then averaged over the 10 replications.

Results

To demonstrate the consequences of the three different LFC signal combinations, we have plotted oscillations in all excitatory phase code units for trial 1, 11 and 21 of the first replication in Figure 5A-C. In trial 1, oscillations start at a random point but are attracted towards a frequency and amplitude determined by the parameters C and r_{min} in equation 3. Since the LFC signals are zero, no hits are given. In trial 11, the two units in the Processing module receive positively correlated bursts, leading to an increase in synchronization. In trial 21, our hypothesis is confirmed by the observation that the two oscillations become desynchronized again because of the opposite LFC signals. Figure 5A-C also clearly demonstrate that the occurrence of hits to oscillations in the Processing module depends on the phase of the excitatory unit in the ACC.

The results of phase synchronization analyses with the dissimilarity function are given in Figure 5D. The plot shows that synchronization or desynchronization between two units can be accomplished very fast and this obtained state will not change as long as the LFC values remain the same. Because the hit signal depends on a probability process in the ACC (equation 6) and is subject to random noise, there can be some variation in how fast synchronization or desynchronization is accomplished.

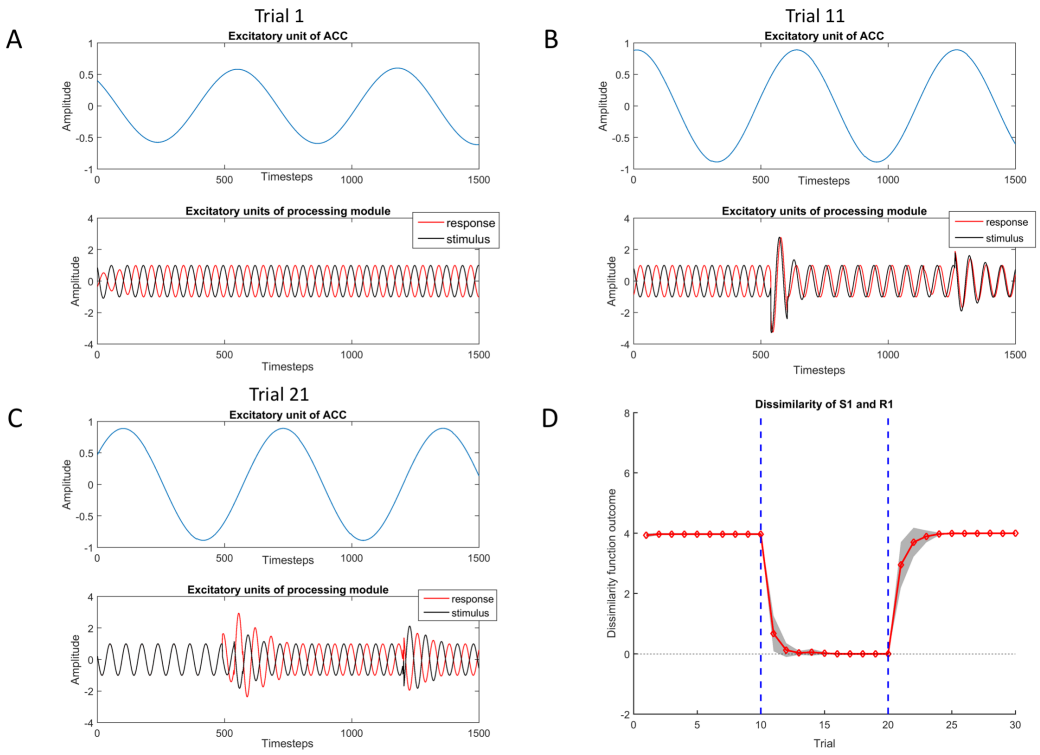

Figure 5: Simulation 1. Overview of synchronization and desynchronization. *A-C show the excitatory phase code neurons across 1500 timesteps of one trial in the first replication. In blue, we see the excitatory unit of the ACC. In the lower box, we see in red the excitatory unit of S_1 and in black the excitatory unit of R_1. Figure 5A gives an overview of trial 1, in which no unit receives any hit. B presents trial 11, in which both units of the Processing module receive positive hits, leading to synchronization. Figure 5C shows what happens in trial 21, in which S_1 receives positive hits and R_1 receives negative hits, leading to desynchronization between their two oscillations. Figure 5D, presents the outcome of the dissimilarity function for oscillations in the excitatory units of S_1 and R_1. The red line shows the average over 10 replications and the shade indicates the 95% confidence interval (mean ± 2*standard error). The vertical blue dashed lines indicate a change in the combination of LFC signals. During the first 10 trials, the oscillations are desynchronized. At trial 11, there is synchronization which is kept until trial 21 where there is desynchronization.*

Discussion

In Simulation 1, we first demonstrated the basic binding by random bursts principle (Springer & Paulsson, 2006; Verguts, in press; Zhou et al., 2005). That is, adding positively correlated bursts to two oscillating units leads to synchronization. Adding negatively correlated bursts on the other hand causes desynchronization. These bursts are in our model given by the Control module, where the ACC determines when the bursts can occur and the LFC determines which units can receive bursts. Thus, prefrontal areas guide the processes in the more posterior cortical processing areas (Miller & Cohen, 2001).

Simulation 2: Communication through coherence

In Simulation 2, we added rate code neurons to the Processing module of our model. In this way, we implement actual communication between our units and we can implement the main assumption of the CTC-principle (Fries, 2005, 2015), which states that the level of synchronization determines the efficiency of communication between two units. Importantly, this change also allows us to link oscillatory activity to actual behavior. As was suggested by previous research (Rodriguez et al., 1999), the current study assumes that desynchronization accomplishes a transition between two different cognitive states. A complete desynchronization should, according to the CTC-framework, make all communication ineffective. It causes a reset, leading to very fast unbinding of previous associations. Thus, in terms of our model, we could hypothesize that if we have two oscillating units, for example one representing a stimulus and the other a response, synchronization will lead to always choosing that response over another when the stimulus is presented. Desynchronization on the other hand, should lead to fast unbinding of this stimulus-response association and making that response the least likely one when the stimulus is presented. To explore this principle of change in response preference, we added an extra R-unit in the Processing module which should act as competing response.

The model

In Simulation 2, the previous model was extended in several ways. First, an extra R-unit was added to the Processing module, introducing the idea of competition between two responses. Hence, also one neuron was added in the LFC, holding the eligibility signal for the new Processing unit. Additionally, each unit in the Processing module gained one rate code neuron. These neurons were connected by constant weights as presented in Figure 6. Finally, a new module was used which is called the Integrator module since it integrates the processes that happen in the Processing module into only two rate code neurons. As shown in Figure 6, these Integrator units have connections with response units in the Processing module and they exert lateral inhibition. The Integrator module implements a competitive accumulator model (Usher & McClelland, 2001) which allowed us to not only record responses but also response times (RT).

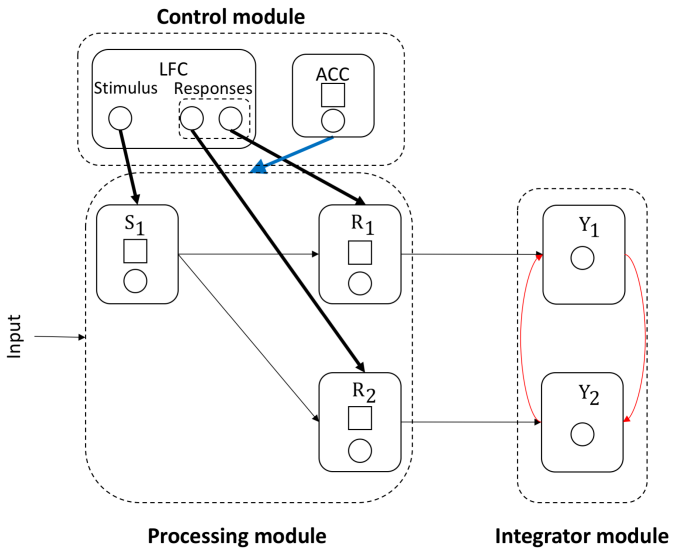

Figure 6. A schematic overview of the model we used for Simulation 2. *Excitatory weights are presented in green and inhibitory weights in red. Again, circles represent rate code neurons and squares represent phase code units.*

The Processing module.

In the adapted model, the phase code neurons obey the same update rules as described in equations 3 and 4. All parameters also remain the same. The only change in the Processing module is that each unit now receives one additional rate code neuron (x_i). This neuron obeys

$$dx_i(t) = -\eta x_i(t) + \left(\boldsymbol{W}\boldsymbol{x}(t) + Z_i(t)\right) f\left(E_i(t)\right) + \sigma_{noise} N(t) \quad (8).$$

In this equation, η=*.9*, which leads to fast activation decay in absence of input. This input can either be internal (**Wx**) or external (Z). The matrix **W** represents the connections between the rate code neurons. These connections are visually presented in Figure 6. For unconnected neurons, the value W_{ij} is 0 but for connected neurons W_{ij} =1. In the current simulation, Z was always 1 for the S-unit. The internal and external stimulation are multiplied by a sigmoid increasing function

$$f\left(E_i(t)\right) = \frac{1}{1+e^{(-5(E_i(t)-\theta))}} \quad (9).$$

This is a key function since it implements the communication-through-coherence (CTC) principle (Fries, 2005, 2015). The working of this function can be explained in terms of the metaphor of the rooms with constantly opening and closing doors. Here, when the door is open, is determined by the oscillation in the excitatory unit. The sigmoid function will stay at 0, or the door will remain completely closed, as long as the oscillation in the E-neuron is below zero. When the E-neuron reaches zero the door will start to go open and reach 0.5 when the E-neuron reaches θ. This means that as long as the oscillation in E is below θ, less than half of the input is actually received by the neuron. Equation 9 will go towards its asymptote 1 as the oscillation reaches its top. Thus, the parameter θ determines how fast the door opens. A high value means that the door will open slowly, therefore the time window in which effective communication can take place is reduced, and the need for synchronization is increased. For a full demonstration of the influence of this parameter, see Verguts (2017). Finally, some random noise is added to the activation function where $N(t)$ is a standard-normal Gaussian and σ_{noise}=.05.

The Integrator module.

The Integrator module consists of two units (Y_1, Y_2) that contain only rate code neurons (y) of which the update rules are given by

$$dy_i(t) = \boldsymbol{A} * \boldsymbol{x}(t) + \boldsymbol{A_{inh}} * \boldsymbol{y}(t) + \sigma_{noise} N(t) \qquad (10).$$

In this equation, **A** is a matrix representing the connections between the neurons in the Integrator module and the rate code neurons in the Processing module. These connections are visually represented in Figure 6. Here, unconnected neurons have A_{ij}=0 and connected neurons have A_{ij}=.3. **x** contains the activation levels of the rate code neurons in the Processing module. Thus, the Integrator neurons receive input from the units in the Processing module. Additionally, the matrix $\mathbf{A_{inh}}$ implements lateral inhibition between the two Integrator units. Here, diagonal elements are zero and off-diagonal elements are -.1. Because of the lateral inhibition, there is competition between the Integrator units. During this competition, one neuron will reach the fixed threshold of 75 while the activation in the other neuron will decay towards zero. When the activation of a neuron reaches the threshold, it has won the competition and the given response and RT are recorded. We set a

lower bound of zero for the activation levels in the Integrator units. So, they cannot become negative. Finally, there is again some random noise added.

The Control module

In the Control module, nothing was changed with respect to Simulation 1.

Method

As in Simulation 1, we wanted to test the different combinations of LFC signals. Therefore, we chose to again perform 10 replications, each containing 30 trials, in which for the first 10 trials $LFC_{S_1}=LFC_{R_1}=0$, in the following 10 trials $LFC_{S_1}=LFC_{R_1}=1$ and in the last 10 trials we set $LFC_{S_1}=1$ and $LFC_{R_1}=$-1. The LFC value for the new unit R_2 was always zero. This time we did not simulate 2500 time steps per trial but the trial was stopped when a response occurred. For future reference, when Y_1 has reached the threshold first, we will say the model has given response 1 and when Y_2 has reached the threshold we will say the model has given response 2. Again, starting points for each phase code pair were values between 0 and 1, randomly sampled from the uniform distribution but for unit S_1 this was multiplied by -1. For all trials following the first trial, the oscillations started at the same point they ended in the previous trial.

Results

Figure 7 shows that equation 9 has successfully implemented the CTC-principle (Fries, 2005, 2015). In this figure, activation of both the rate code and excitatory phase code neurons of units S_1 and R_1 are plotted across a given trial. As in Simulation 1, we see that for the first trial the phase code neurons are desynchronized. The rate code neuron of the S_1-unit receives a constant external input (*Z*) but can only receive this input when the door is open. This is illustrated by the fact that activation levels of the rate code neurons follow oscillations in the excitatory phase code neurons. Because oscillations in S_1 and R_1 are completely desynchronized, the 'doors' of S_1 and R_1 are open at different times. Therefore, the output of unit S_1 is not received by unit R_1. Thus, activation of R_1 contains only random noise. However, at trial 11 the oscillations become synchronized. Therefore, the doors of both units are open at the same time and the rate code neuron of unit R_1 is able to receive the output of unit S_1. In sum, synchronization of phase code neurons leads, in our

model, to effective communication between their respective rate code neurons. Desynchronization on the other hand leads to ineffective communication.

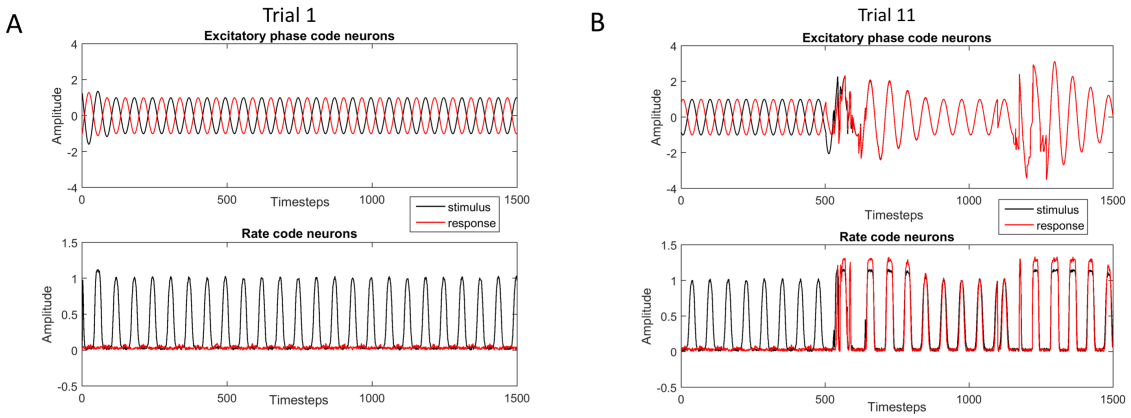

Figure 7. Simulation 2. An overview of the CTC-principle (Fries, 2005, 2015). *This figure shows plots of activation in the rate code (lower box) and excitatory phase code neurons (upper box) of the Processing units S_1 (in black) and R_1 (in red). Figure 7A gives an overview of 1500 timesteps in trial 1 of the first replication. Figure 7B gives an overview of 1500 timesteps in trial 11 of the first replication.*

A new factor in the current simulation, is that there is a new unit, R_2, in the Processing module which never receives a hit. Since the other two units do receive hits, the degree of synchronization between R_2 and the others will not stay the same. On the other hand, there is no way to say whether a hit will lead to more, or less synchronization. Because the hit signals that the two units, for example S_1 and R_2, receive are completely uncorrelated the process of synchronization or desynchronization is completely random. That this is the case is shown in figures 8A and B. In these figures, again results are shown for the dissimilarity function of each trial averaged across replications. The dissimilarity of oscillations in the excitatory phase code neurons of S_1 and R_1 is plotted (Figure 8A) as well as the dissimilarity in S_1 and R_2 (Figure 8B). The figure shows that for S_1 and R_1 everything was the same as in Simulation 1. No change in the first 10 trials, synchronization in the following 10 trials and desynchronization in the last 10 trials. The dissimilarity between S_1 and R_2 on the other hand, seems indeed to be completely random. In the first 10 trials, no hits were given, so their phase relation did not change. However,

starting at trial 11 the dissimilarity function seems to just change randomly. This is illustrated by the large confidence intervals.

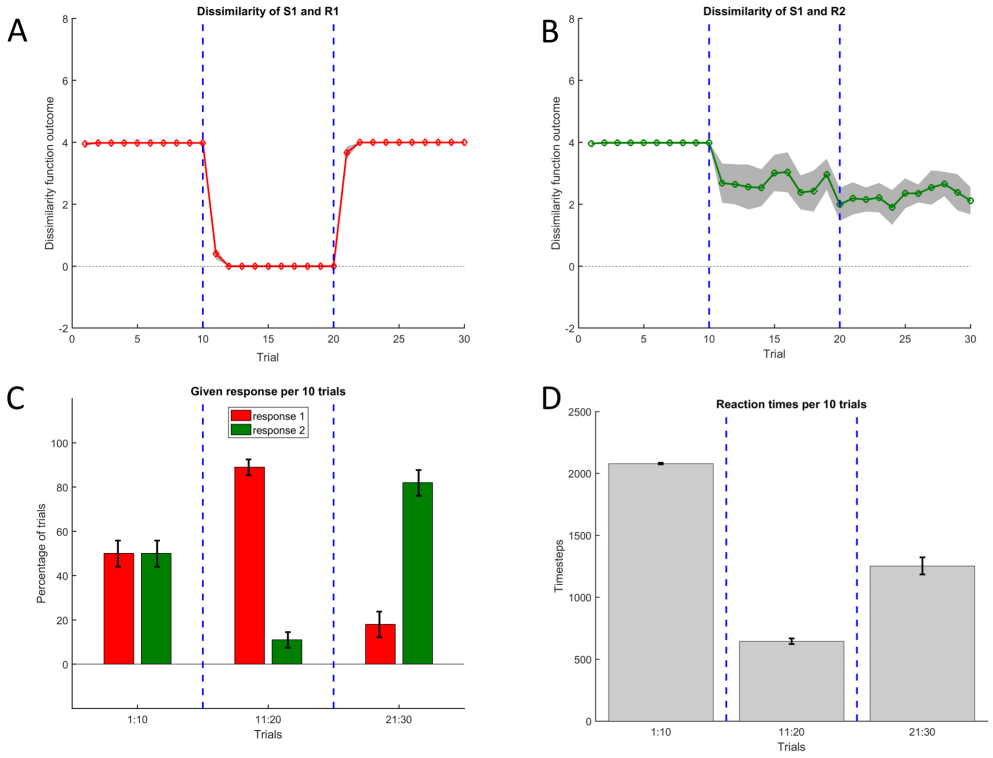

Figure 8. Data from Simulation 2. *A and B show plots of the outcome of the dissimilarity function for oscillations in the excitatory units for each trial. Figure 8A shows the dissimilarity of units S_1 and R_1 and B shows the dissimilarity of S_1 and R_2. The line presents the average over 10 replications and the shade indicates the 95% confidence interval (mean ± 2*standard error). The blue, vertical dashed lines indicate a change in the combination of LFC signals. Figure 8C shows the percentage of trials in which a certain response was given averaged across 10 replications and plotted per 10 trials. Red bars represent the occurrence of response 1 and green bars represent response 2. Error bars indicate the 95% confidence intervals (mean ± 2*standard error). Figure 8D presents the response times of the model averaged across 10 replications and means plotted per 10 trials. Error bars indicate the 95% confidence intervals (mean ± 2*standard error).*

As mentioned before, synchronization between units enhances their communication and desynchronization leads to a reset of this communication. Therefore, synchronizing R_1 with S_1 should lead to more responding with response 1 than with response 2. Desynchronizing R_1 with S_1 on the other hand, should lead to less responding with response 1 and thus more response 2. That this is the case is shown in Figure 8 C, where we show

the percentage of trials in which a certain response was given, averaged per 10 trials. In the first 10 trials, the model did not synchronize or desynchronize. Since the weights between the rate code neurons, which are denoted in **W** of equation 8, for S_1 and R_1 have the same strength as the ones between S_1 and R_2, the response that wins the competition is determined randomly. In trials 11 until 20, the model synchronized oscillations in R_1 and S_1. Therefore, communication between them should be more effective and Y_1 should win the competition from Y_2. In Figure 8C we see indeed that response 1 was given more often than response 2 in trials 11-20. In trials 21-30, there was desynchronization between S_1 and R_1. This led to a reset of their association and therefore choosing more the other response.

Of course, synchronization, and therefore more effective communication, does not only determine which response is given but also how fast the Integrator unit reaches the threshold. As can be seen in Figure 8D, it takes more time steps to find a winning response when both R-units are completely desynchronized with the S-unit then when one of the R-units has achieved complete synchronization with the S-unit. When there is no clear synchronization or desynchronization, the response times are somewhere in the middle.

Discussion

In Simulation 2, we have successfully implemented and demonstrated the CTC-principle (Fries, 2005, 2015). We illustrated this in Figure 7. In this figure, one can see that information transfer between two units is only possible if their oscillations are synchronized. When there is complete synchronization, the door for input of the receiving neuron is open when the sending unit releases its output. An increase in synchronization between two units thus improves their communication. Desynchronization on the other hand can rapidly make this communication ineffective again. The effects of synchronization and desynchronization are also reflected in the given responses and response times in the sense that the model generally responded with the response-unit that showed the strongest synchronization with the stimulus-unit. As also observed in empirical data (Womelsdorf, Fries, Mitra, & Desimone, 2006), the level of gamma-synchronization predicted the RT of the model. In sum, Simulation 2 has demonstrated that phase synchronization is an effective way to implement biased competition in a computational model.

Simulation 3: Finding out what to synchronize or desynchronize

Until now, the model was always told which units it had to synchronize or desynchronize (i.e., as in Verguts, 2017). Simulation 3 went beyond this earlier work, and implemented a way for the model to explore on its own which units should be synchronized or desynchronized. The model should thus be able to evaluate its own performance and use this evaluation to adjust where needed. This can be done by implementing an Actor-Critic structure (e.g. Houk, Adams, & Barto, 1995) in the model. Here, the Actor uses an evaluation of the Critic to adjust processes in the rest of the model. In the current model, making adjustments means synchronizing or desynchronizing two units in the Processing module. Since synchronization is determined by the Control module, the Control module will function as the Actor of our model. To implement a Critic, a new module is needed. For this purpose, we use the reward value prediction model (RVPM; Silvetti et al., 2011). In the RVPM, the rostral part of the ACC functions as Critic by comparing reward expectations with actual received rewards.

In sum, in the current model we implement a Critic module which is adapted from the RVPM (Silvetti et al., 2011). This Critic module will evaluate the performance of the model and send this evaluation to the Control module. The Control module will then on its turn give hits to the units in the Processing module in order to reinforce or revise current behavior.

The model

In the new model, presented in Figure 9, one main change was made relative to Simulation 2. A Critic module was added, consisting of 3 rate code neurons. Two neurons represent signed prediction errors, which compute the discrepancy between the actual received reward and the estimation of the model. These neurons thus hold information about how far the model is removed from its optimal state. The third neuron computes the expected reward given a certain response. This unit, *V*, is connected to the units in the Integrator module because these units code for which response is given. Additionally, the Critic is connected to the Control module because the evaluative information computed by the Critic should be used in the Control module to act. This connection received a

modulating signal from the Integrator module in order to inform the Control module about the current state of the model.

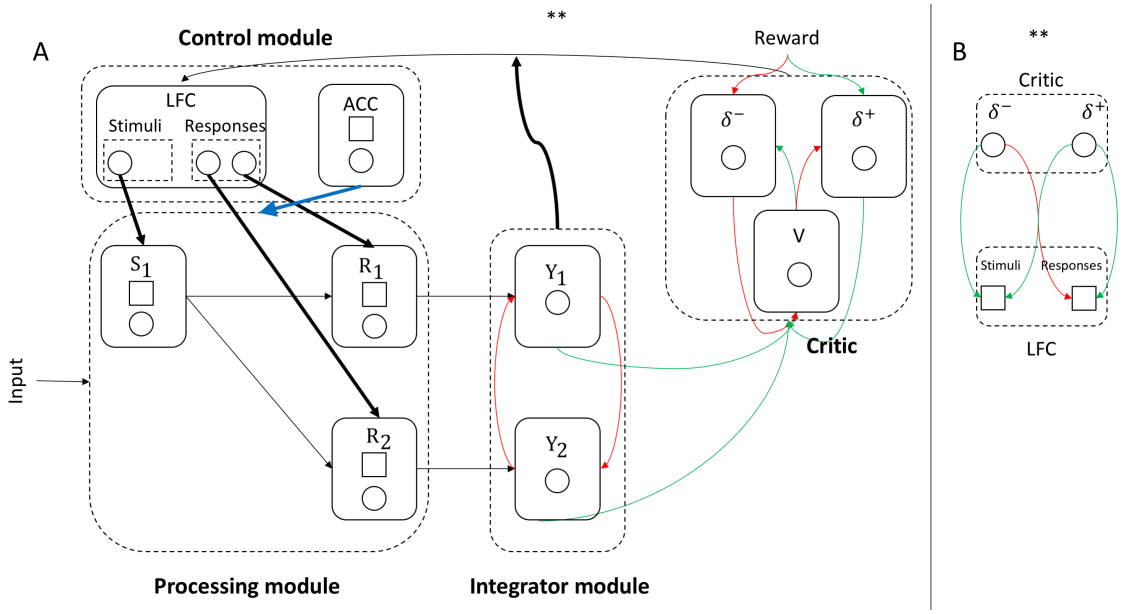

Figure 9. Model sketch. Figure 9*A shows a schematic overview of the model that was used for Simulation 3. Green arrows indicate excitatory weights and red arrows indicate inhibitory weights. Again, circles represent rate code neurons and squares represent phase code units. Connections between units in the Critic and the Control module (**) are specified in B.*

The Processing module.

In the Processing module, all update rules and parameters were the same as in Simulation 2.

The Integrator module.

Also in the Integrator module, all update rules and parameters were the same as in Simulation 2.

The Critic.

As Critic, we implemented the RVPM (Silvetti et al., 2011). Here, there is one expected reward unit, V, which holds an estimation of the reward the model will receive given a certain response. This estimation is made by

$$V = n * \boldsymbol{E}' * \boldsymbol{y}(T) \qquad (11).$$

In this equation, $\boldsymbol{E}$ is a vector representing the connections between the Integrator module units and the value unit as presented in Figure 9A and $\boldsymbol{y(T)}$ is a vector containing the values of the rate code neurons in both Integrator units at timestep T. Here, T is the response time of the model. So, information transfer between the Integrator module and Critic happens only once a response is given. This could be modelled more explicitly by letting these connections be modulated by a reafferent feedback signal, indicating that a response occurred. The term $n=1/75$ is a normalization constant which controls for the fact that the end-value of an Integrator unit is always 75. V, as well as all other units in the Critic, are rate code neurons which have no time index because they only take on one value per trial.

Thus, information about the value of a certain response is coded in the $\boldsymbol{E}$ vector, representing the connections between the Integrator module and the Critic. These weights are updated by the RVPM learning rule (which is a variant of the Hebbian learning rule), and is given by

$$\Delta E_i = \alpha * V * y_i(T) * (\delta^+ - \delta^-) \quad (12).$$

Here, α is a learning parameter with a value of .1. We chose to assign a starting value of .5 to all these weights in the first trial.

The Critic also contains two prediction error units, one positive prediction error unit and one negative prediction error unit. The negative prediction error for each trial is computed by

$$\delta^- = \max(0, Rew - V) \quad (13)$$

and the positive prediction error is given by

$$\delta^+ = \max(0, V - Rew) \quad (14).$$

In these equations, *Rew* stands for the received reward and V is the outcome of equation 11, reflecting the reward the model predicted to receive (see Silvetti et al., 2011, for more details).

The Control module

In the Control module, nothing changes for the ACC relative to Simulation 2. Crucially, for the LFC units, the eligibility signals are no longer arbitrarily assigned but they are now computed by the model itself. This computation is performed by multiplying input from the Critic module, reflecting the performance of the model, with the input from the Integrator module, reflecting current behavior. The activation of the LFC neurons connected with the responses, or R-units, can be described by the Hebbian-like activation rule,

$$LFC_i = \varepsilon * (\delta^+ - \delta^-) * n * y_i\,(T) \qquad (15)$$

in which ε = 3 represents the weight strengths between units in the Critic and the Control module. n = 1/75 is again a normalization constant. As illustrated in Figure 9A and B, The LFC units receive input from the two prediction-error-units in the Critic but this input is modulated by the *y(T)* term. This term codes for the winning response, in the sense that it is zero for the losing response and 75 for the winning response. The respective LFC neuron will thus have an activation of zero when the corresponding response was not given in the previous trial. It will take on positive values if the corresponding response elicited a positive prediction error and negative values if the response elicited a negative prediction error.

The activation of the LFC neurons connected with the S-units can be described by

$$LFC_j = \varepsilon * (\delta^+ + \delta^-) \qquad (16).$$

This equation differs from equation 15 in two ways. First, there is no modulating signal. This is not needed because there is, in the current model, only one S-unit which is active in all trials. Second, as illustrated by the weights in Figure 9B, the prediction errors are added instead of subtracted. Hence, the LFC signal for the stimulus units is always positive. ε is also in this equation a constant with value 3.

The combination of LFC signals given by equations 15 and 16 will lead to synchronization of the stimulus and response units when the response led to a positive prediction error in the previous trial. When the response elicited a negative prediction error,

the LFC signals for stimulus and response will have opposite signs, which will lead to desynchronization. When there is no prediction error, no hits are given because both equations will be zero. Of course, this would also mean the model reached an optimal state.

Method

In the current simulation, we chose to do 10 replications of 40 trials in which for the first 20 trials response 1 was rewarded (*Rew*=1) and in the last 20 trials response 2 was rewarded. In this way, the model had to first bind two units and then unbind them. Therefore, both synchronization and desynchronization will be needed. Again, a trial stopped when a response occurred and the starting points for each excitatory phase code unit were values between 0 and 1, randomly sampled from the uniform distribution but for unit S_1 this was multiplied by -1. For all trials following the first trial, the oscillations started at the same point they ended in the previous trial.

Results

For the current simulation, we first checked whether the model was successful in learning what response was rewarding. Here, this is reflected by the weight values represented in the vector $\boldsymbol{E}$ of equation 11. In the current simulation, response 1 was rewarded during the first 20 trials and response 2 during the last 20 trials. Hence, the weight value corresponding to response 1 should increase during the first 20 trials and the weight value corresponding to response 2 should decrease. At trial 21 this trend should be inversed. That this is the case is shown in Figure 10A.

As shown in Figure 10A, the weight-strength for the not rewarding response does not keep declining after the change in task demands. This is because the model is implemented in such a way that the expected reward for a certain response can only change when the respective response occurred. Given the flexibility of our model, response 1 is not given any more after it was not rewarded anymore. The absence of a reward caused a negative prediction error which led to immediate desynchronization. Therefore, response 1 cannot win the competition anymore. That the model is really this flexible is demonstrated by Figure 10B. Although the model can only know at the end of trial 21 that response 1 is not rewarding anymore it still only gives an average of 3 wrong responses after trial 20.

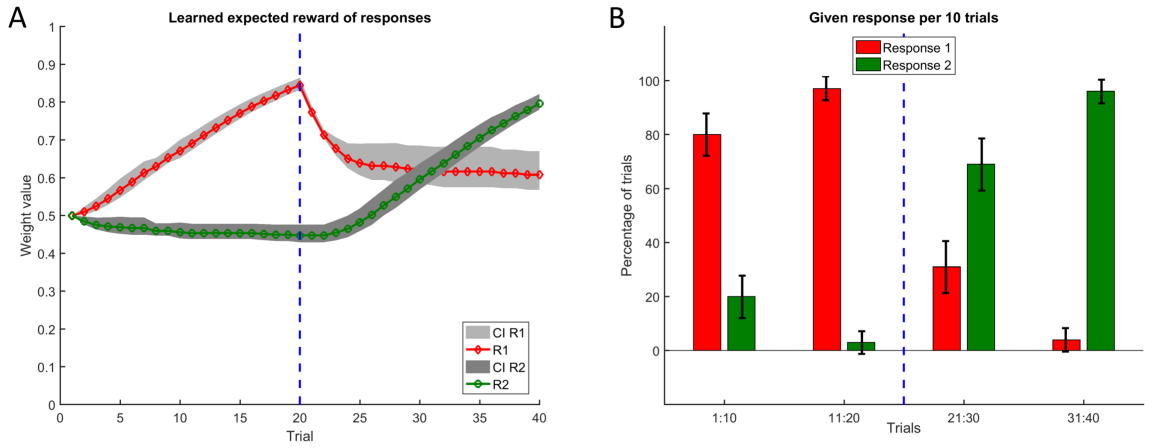

Figure 10. *Simulation 3. Figure 10A: A plot of the change in expected reward of the responses. The weight strengths between Integrator- and Critic module indicate the expected-reward the model has assigned to a certain response. The red line shows the average weight strength with Y_1, coding for the expected reward of response 1 and the green line represents the average weight strength with Y_2, coding for the expected reward of response 2. The shades indicate the 95% confidence intervals (mean ± 2*standard error). Figure 10B: Percentage of trials in which a response was given averaged across 10 replications and plotted per 10 trials. Red bars represent the occurrence of response 1 and green bars represent response 2. Error bars indicate the 95% confidence intervals (mean ± 2*standard error). The blue vertical dashed lines indicate the change in task demands.*

Thus, the model succeeds in finding out which response is more rewarding and is able to use this information in order to give the most rewarding response. Additionally, the model has proven to be very flexible in adapting to changes in these response-reward associations. As illustrated in Figure 11A and B, this flexibility is in line with the flexibility of synchronization and desynchronization. In the plots of the dissimilarity function outcome in Figure 11, it is shown that during the first trials the model achieves synchronization for the S- and R_1-unit while oscillations in the R_2-unit remain desynchronized with S_1. At trial 20, the model observes that response 1 is no longer rewarded. Therefore, it decides to rapidly desynchronize R_1 with S_1. This implies that response 1 is not given anymore. By giving response 2, the model experiences that this response is now rewarded and therefore reinforces that behavior by synchronizing the corresponding R-unit with the stimulus-unit.

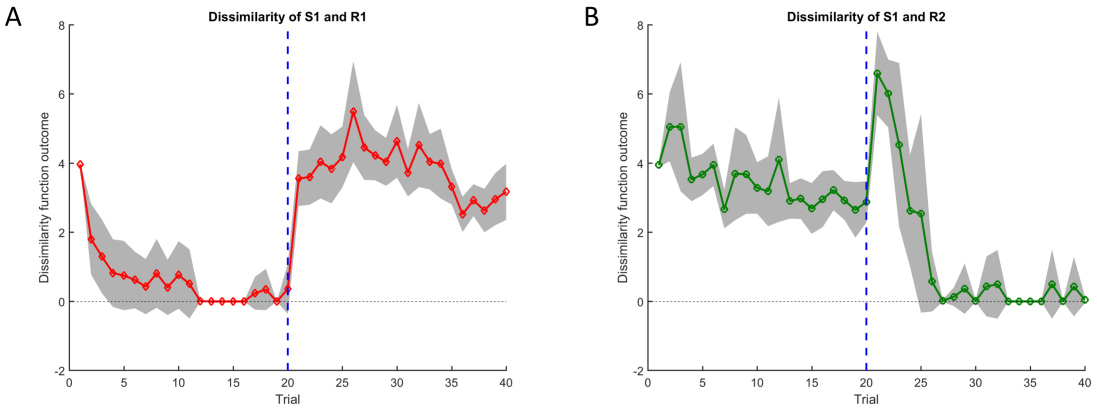

Figure 11. Simulation 3, Plot of the outcome of the dissimilarity function for oscillations in the excitatory neurons for each trial. *Figure 11A shows the dissimilarity of units S_1 and R_1 and Figure 11B shows the dissimilarity of S_1 and R_2. The line presents the average over 10 replications and the shade indicates the 95% confidence interval (mean ± 2*standard error). The vertical blue dashed lines indicate the change in task demands.*

Perhaps a bit remarkable is that right after the change in response-reward association there is a high dissimilarity of oscillations in S_1 and R_2. This can be explained by the nature of the hits. As described by equation 5, hits are a combination of the rate code unit in the ACC, the LFC units and some random noise. The influence of the noise is small and the value of the ACC is determined by a Bernoulli process. Therefore, it is either one or zero. Thus, the size of the hit is solely determined by the value of the LFC unit. Equation 15 and 16 describe how the value of the LFC unit is on its turn determined by the size of the prediction errors. In the current simulation, the model had first learned to assign a high reward expectation to response 1 (see Figure 10A). Suddenly, at trial 21, this response is not rewarded anymore which leads to a huge prediction error and therefore also a huge hit-signal to oscillations in both, the S- and R_1-unit. This hit caused on its turn huge changes in their oscillations in both amplitude and frequency (see Figure 5B and C for an illustration of the hit consequences). Then again, the R_2-unit does not receive a hit because response 2 was not given in that trial. Therefore, oscillations of the R_2-unit just keep unchanged while the frequency of oscillations in S_1 are significantly altered, leading to a high dissimilarity.

The important factor is that oscillations in S_1 and R_1 get the same (but random) size of hit but with opposite signs, therefore their frequency during the hits remains the

same but they become completely desynchronized. This means that the output of S_1 can never be received by R_1. However, even though it will be far less efficient, it might be possible to exchange some information through doors that are opening and closing with a different frequency. Thus, although the dissimilarity function indicates a higher dissimilarity, this might be an artifact of the received hits instead of a real measure of effective communication. As soon as response 2 is given, or the prediction error is reduced, this artifact will disappear.

Discussion

In Simulation 3, we implemented an Actor-Critic structure in the model. By the addition of the Critic, the model is now able to evaluate its own performance by comparing expected rewards to actual received rewards. This evaluation is then used by the Control module, which functions as the Actor, to send synchronizing or desynchronizing hits to the units of the Processing module. Results show that the model is indeed able to learn which response is most rewarding and it uses this information to synchronize rewarding responses and desynchronize unrewarding responses. When the response-reward association is changed, the model shows a great flexibility in adjusting its own performance to obtain the maximum total reward.

The flexible change in synchronization levels between two neurons is here caused by changes in the LFC signal. This LFC signal changes on a trial-by-trial basis in response to an error signal sent by the Critic. This makes the phase synchronization process very ‘one-shot’, in the sense that one trial can make the difference between a fully synchronizing hit and a fully desynchronizing hit. However, although a change in the error signal also leads to a change in the LFC signal, this does not always lead to a significant change in synchronization levels. As long as both areas receive correlated, positive hits they remain synchronized. Only when there is a change in the sign of the LFC signal this hit will lead to significant changes in the synchronization levels between two oscillating units. Thus, although the hit signal is constantly changing, the synchronization levels remain quite robust.

Simulation 4: A reversal learning task

In previous simulations, the task was very easy. In each simulation, there was only one stimulus and the only thing the model had to do was selecting the right response. Thus, which stimulus was presented did not matter; the model just had to learn a response-reward association. In the current simulation, we went one step further by adding a second stimulus. In this way, the model now had to learn about (higher order) stimulus-response-reward associations. More specifically, the model in Simulation 4 had to perform a reversal learning task (e.g. Clark, Cools, & Robbins, 2004; Izquierdo & Jentsch, 2012). The task for the model was to learn that given stimulus 1, response 1 will be rewarded while response 2 will be rewarded when stimulus 2 is presented. After some time, these rules are reversed, associating stimulus 1 with response 2 and stimulus 2 with response 1. In this way, we could test the model's flexibility in both binding and unbinding of higher-order associations.

The model

In comparison to the previous model, the current model, shown in Figure 12, was changed in two ways. First, a new S-unit was added in the Processing module. Hence, also a corresponding stimulus-neuron was added in the LFC. Second, the Integrator module contained four instead of two units. This is because reward information is for the current task not only coded by which response is given but by which stimulus-response combination is present. As in the study of Holroyd and Coles (2002), the current model uses units that integrate information of stimulus-response associations. Since there are four stimulus-response combinations, there are four Integrator units.

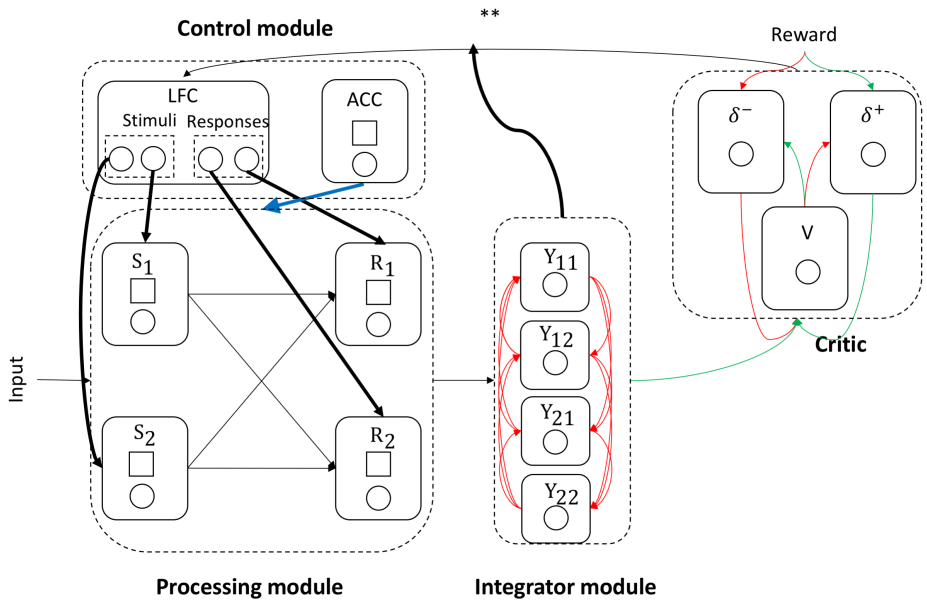

Figure 12. A schematic overview of the model that was used for Simulation 4. *Green arrows indicate excitatory weights and red arrows indicate inhibitory weights. Again, circles represent rate code neurons and squares represent phase code units. Connections between units in the Critic and the Control module are the same as specified in figure 9B.*

The Processing module

In the Processing module of the new model, only one change was made. We added one stimulus-unit, S_2, which, like all other units, consists of a phase code pair and a rate code neuron. The rate code neuron of this unit is connected to the rate code neurons of both response units. The value of these weights is the same as for the unit S_1. This means the matrix $\boldsymbol{W}$ in equation 8 is extended to a 4 by 4 matrix.

The Integrator module

The Integrator module now contains four rate code neurons. These neurons integrate the information of stimulus-response pairs. This integration is done by implementing the following activation rule,

$$dy_{ij}(t) = A * R_i(t) + U * S_j + \boldsymbol{A_{inh}y}(t) + \sigma_{noise}N(t) \qquad (17).$$

In this equation, a weighted sum is made of activation in both the response- and stimulus-unit. As in previous simulations, the Integrator units exert lateral inhibition and a noise

term was added. The weight with the response unit, *A*, had a value of .2. The weight for the stimulus unit, *U*, had a value of .05. In comparison to previous simulations, the noise σ_{noise} was reduced to .025. This was also the case for the rate code neurons in the Processing module. $\boldsymbol{A_{inh}}$ was still a matrix in which diagonal elements are zero and off-diagonal elements have a value of -.1. However, here $\boldsymbol{A_{inh}}$ was extended to a 4 by 4 matrix. The parameters were set in this way in order to let the model still reach the response threshold of 75 at approximately the same time as in previous simulations. When one of the Y_{1j}-units reached the threshold first, response 1 was given and when one of the Y_{2j}-units reached the threshold, response 2 was given. Because of the lateral inhibition, only one of the units reached the threshold while activation of all other Integrator units decayed to zero.

The Critic

For the Critic, nothing changed with respect to Simulation 3 but the value unit was now connected to four Integrator units instead of one. Therefore, the Critic did not compute the expected reward given a certain response, but the expected reward given a certain stimulus-response pair.

The Control module

In the Control module, one rate code neuron was added which corresponded to the new S-unit in the Processing module. In Simulation 3, we described how the LFC response signals, given by equation 15, were modulated by the winning Integrator unit. This was done to only make the given response eligible to receive a hit. However, in the current model which response is given is not determined by one Integrator unit but by two. Therefore, equation 15 is slightly changed. The new rule is given by

$$LFC_i = \varepsilon * (\delta^+ - \delta^-) * n * \sum_{j=1}^{J} y_{ij}(T) \qquad (18).$$

Because of the lateral inhibition, only one Integrator unit has an end-value different from zero. Therefore, the sum term just returns the winning Integrator unit.

Because of the addition of an extra stimulus unit, the LFC signals corresponding to the stimuli also need modulation in order to only make the previously active stimulus-unit eligible to receive a hit. Therefore, the new rule for the LFC stimulus neurons is given by

$$LFC_j = \varepsilon * (\delta^+ - \delta^-) * n * \sum_{i=1}^{I} y_{ij}(T) \qquad (19).$$

For the ACC unit, nothing has changed with respect to previous simulations.

Method

The process in the current simulation was the same as in Simulation 3. 10 replications of 40 trials were performed. In the first 20 trials, response 1 was rewarded (*Rew*=1) in presence of stimulus 1 and in the presence of stimulus 2, response 2 was rewarded. At trial 21 this rule was reversed, implementing a change in task demands. So, in the last 20 trials response 2 was rewarded in the presence of stimulus 1 and in the presence of stimulus 2, response 1 was rewarded. Again, the model had to first bind and then unbind units. The stimuli were presented in an alternating way, S_1 being presented during all odd trials and S_2 during all even trials. Again, a trial stopped when a response occurred. The starting points for each excitatory phase code unit were values between 0 and 1, randomly sampled from the uniform distribution but for both S-units this was multiplied by -1. For all trials following the first trial, the oscillations started at the same point they ended in the previous trial.

Results

In the current simulation, we explored the performance of the model in a slightly more complex task. In particular, we tested whether phase synchronization is also able to solve a task with stimulus-response-reward associations instead of response-reward associations. Therefore, the model should first achieve the correct levels of synchronization between stimulus- and response-units. As shown in Figure 13, the model indeed succeeds in achieving synchronization between S_1 and R_1 and between S_2 and R_2 during the first 20 trials. The other S-R combinations remained desynchronized during the first 20 trials. At trial 21, the situation was reversed. Then, the model successfully unbound the previously associated units and bound new ones. One slight remark that can be made is that during the last 20 trials the model is less successful in achieving complete synchronization.

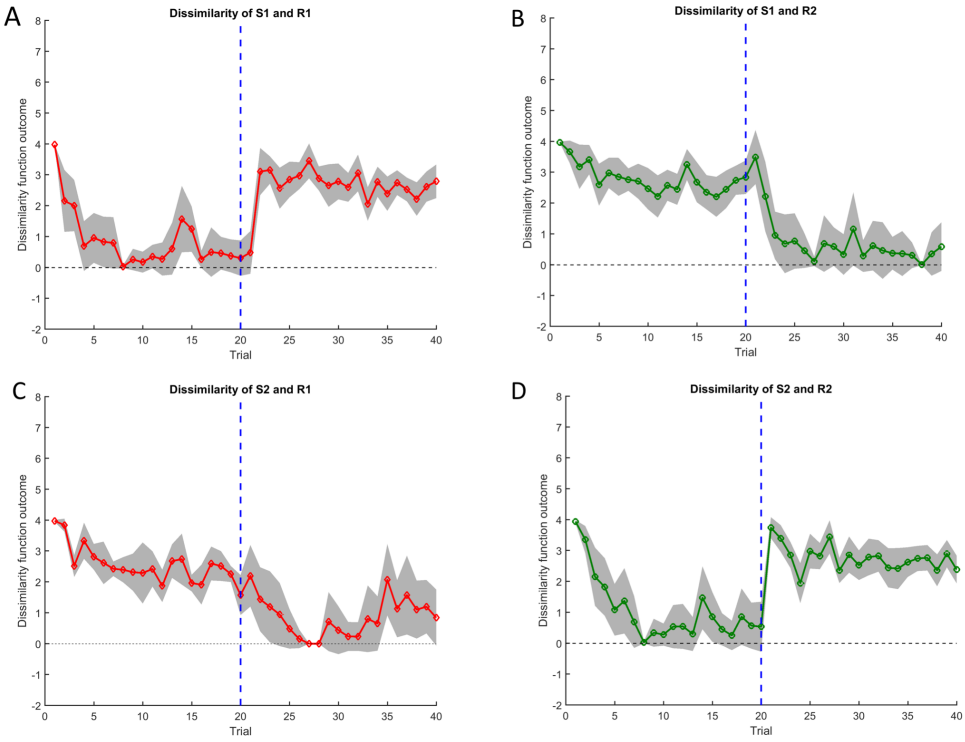

Figure 13. *Simulation 4.* Plot of the outcome of the dissimilarity function of oscillations in the excitatory neurons for each trial. *Figure 13A shows the dissimilarity of units S_1 and R_1; B of S_2 and R_2; C of S_2 and R_1 and Figure 13D shows the dissimilarity of S_1 and R_1. The line presents the average over 10 replications and the shade indicates the 95% confidence interval (mean ± 2*standard error). The blue dashed lines indicate the change in task demands.*

The fact that synchronization processes were successful is also reflected in the accuracy of the model. In Figure 14, it is shown that the model performed quite well. Remarkably, accuracy is highest in the first 10 trials after the change in task demands. The model seems thus to be indeed very flexible. The fact that synchronization itself was less successful after the change in task demands is reflected in the accuracy data by a very broad confidence interval in the last 10 trials.

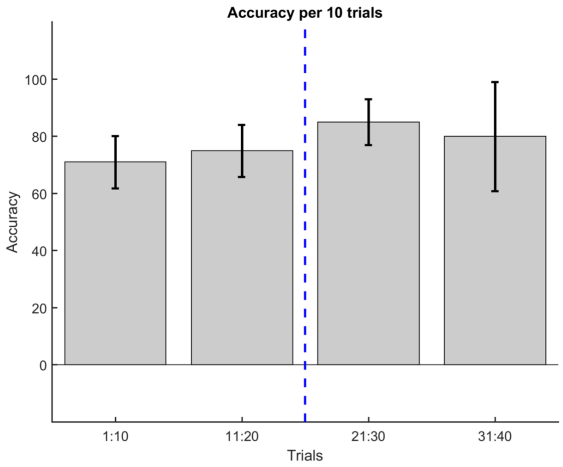

Figure 14. Simulation 4. Accuracy percentage of the model averaged across replications and means plotted per 10 trials. *Error bars show the 95% confidence intervals (mean ± 2*standard error). The vertical blue dashed line indicates the change in task demands.*

Discussion

In Simulation 4, we demonstrated how the model still shows a great flexibility in dealing with higher order associations. By adapting the idea of Holroyd and Coles (2002) to implement units that compute the value of stimulus-response pairs, the model can deal with higher order associations. The model performs quite well on the more complex task but seems to have some problems in the last trials. In these trials, the model does not achieve full synchronization for the correct stimulus-response pairs. This is reflected in the variability of accuracy across replications during those trials. Besides these troubles, accuracy was highest in the first 10 trials after the change in task demands. This indicates that especially unbinding by desynchronization might be very effective. In the first trials after task demands had changed, desynchronization is the main process. First, the previous associations have to be undone or unlearned and then new associations can be made. As can be seen in Figure 13, desynchronization is very effective. At first, synchronization seems to be also quite effective. However, in the last trials it looks as if it does not stick to its obtained optimal state. So, in the current simulation phase synchronization did not show a good robustness.

In the current simulation, the absence of robustness might be mainly due to the fact that phase synchronization is here implemented in a one-shot manner. The LFC values, determining the hit are overwritten every trial. In the current model, one might think about the LFC signal as a sort of binary learning signal. Binary, because there are only two objectives; synchronization or desynchronization. As long as both LFC values remain positive, a change in the signal will not lead to a change in synchronization levels. While in learning, the weight values will change gradually towards the objective, a change in objective leads very fast to big changes in synchronization levels. This makes it more flexible but also less robust.

Additionally, only one stimulus-unit and one response-unit can receive hits every trial. Crucially, hits to, for example, units S_1 and R_1 do not only have consequences for their synchronization levels but also for synchronization between S_1 and R_2, and S_2 and R_1. In sum, in contrast to typical learning algorithms, synchronization is very fast instead of gradual and the relation one unit has to another is not independent of its relationship with a third unit. This implies that phase synchronization shows a lower robustness than general anatomical learning.

Simulation 5: The added value of phase code

So far, making stimulus-response associations was solely done by the achievement of phase synchronization. However, the current paper aims to investigate whether phase synchronization could be a useful supplementary way for flexibly altering communication in the brain on top of weight learning. In the previous simulations weight learning was only done for weights in the Critic and the weights between the Critic and Integrator module. The connections between these units were not subject to synchronization processes. Therefore, we will in the current simulation implement learning in the Processing module.

On the one hand, neuronal synchronization or desynchronization should help the slow learning weights to increase flexibility. On the other hand, Simulation 4 has indicated that, especially in the way that it is implemented here, phase synchronization might not be as robust as anatomical learning. To explore the interactions between the two kinds of learning, we will compare the performance of three models.

The first model has been described in Simulation 4; this model only used phase synchronization and desynchronization to change behavior. Therefore, we will further refer to this model as the Phase code model. In Simulation 5.1, we will expand the model of Simulation 4 by implementing weight learning between the stimulus- and response units in the Processing module. This model combines both phase code processes and rate code anatomical learning. Therefore, we will call this model the Phase-rate code model. The third model, which is presented in Figure 15, will be a very basic rate code model which only uses anatomical weight learning. This model will further on be referred to as the Rate code learning model. Additionally, we explored in the last two models different rates of anatomical weight learning.

Simulation 5.1: The Phase-rate code model

In the current simulation, the Phase code model of Simulation 4 was changed in only one respect. Anatomical weight learning was implemented for the connections in the Processing module.

The model

The structure of the model was not changed with respect to the previous simulation, it still corresponds to the model presented in Figure 12. The only difference is that connections between the stimulus- and response units in the Processing modules are now subject to learning.

The Processing module.

For the current simulation, the strength of connections within the Processing module will change on every trial. This means that the weights, between S- and R-units, represented by the matrix **W** in equation 8 are subject to a learning rule. For this purpose, the delta rule (Widrow & Hoff, 1960) was used. This rule is given by

$$\Delta W_{ij} = \beta * \left(Objective - max\left(x_j\right)\right) * max\ (x_i) \qquad (20)$$

In this equation, β is a learning parameter determining the speed at which the weights learn. The *Objective* implements a supervised learning feedback; as a result, the weights will keep changing until the objective is reached by the output unit. Here, x_j is the output unit, in the current model the rate code neuron of a response unit, and x_i is the input unit, in the current

model the rate code neuron of a stimulus unit. Because activation levels of these rate code neurons oscillate, we use the maximum activation within a trial. All weights started with an initial value of .5. Apart from this update rule, nothing was changed in the Processing module.

Other modules.

In comparison to the model in Simulation 4, no changes were made in any of the other modules.

Method

In the current simulation, we again performed 10 replications of 40 trials. In the first 20 trials, response 1 was rewarded (*Rew*=1) in presence of stimulus 1 and in the presence of stimulus 2 response 2 was rewarded. At trial 21, this rule was reversed, implementing the change in task demands. For stimulus-response associations that were rewarded, the *Objective* of the learning rule in equation 20 was 1 and for unrewarded associations this *Objective* was 0. Hence, also these objectives were reversed at trial 21. The stimuli were presented in an alternating way, S_1 received external stimulation (Z=1) during all odd trials and S_2 during all even trials. Again, a trial was stopped when a response occurred. Again, the starting points for each excitatory phase code unit were randomly sampled values between 0 and 1 but for both S-units this was multiplied by -1 in order to have desynchronization between stimulus- and response units at the start. For all trials following the first trial, the oscillations started at the same point they ended in the previous trial. This process was repeated with three different values of the learning rate (β= .025, .05, and .1, respectively).

Results

Figure 16A-C show the accuracy data of the Phase-rate code model with the different learning rates. We see in the accuracy percentage of trials 21-30 that, for all learning rates, accuracy dramatically declined after the switch in task demands. However, with a fast learning rate (β=.1), the model achieves a quite good accuracy in the last 10 trials, while with a slow learning rate (β=.025), accuracy stays low in these trials. The performance of the model with an intermediate learning (β=*.05*) rate lies in between.

Simulation 5.2: The Rate-code learning model

The rate-code learning model holds 4 neurons, representing 2 stimuli and 2 responses. The activation for the stimulus units just equaled the external input they received, $S_i=Z_i$. The response-units obeyed the activation function given by equation 1. Similar as in the phase- and rate-code model, all weights had an initial value of .5 and were updated by the rule given by equation 20. The recorded response corresponded to the response unit with the highest activation.

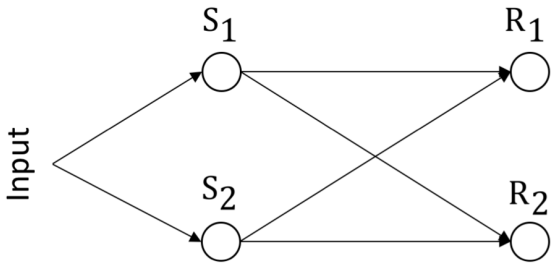

Figure 15. Simulation 5, Sketch of the rate code model.

Method

For the Rate code learning model, we also simulated 40 trials. In these trials, the objective of the learning rule and the external stimulation for the stimulus units were subject to the same manipulations as in Simulation 5.1. Again, this was done for the three different learning rates.

Results

In Figure 16 D-F, we see that the Rate code learning model performs really well in the first 20 trials. It only makes one error at the first trial because all weights start with the same value and the response determination is therefore random. However, when the change in task demands occurs at trial 21, we see that it takes at least 10 trials for the model to unlearn the previous association and learn the new one. When the learning rate is slow (β=.025), or intermediate (β=.05), it even takes longer. On the other hand, we see, for the model with the fast learning rate (β=.1), that when the new task rule is learned, the model again achieves 100% accuracy (Figure 16F). This demonstrates that although weight learning might be slow, it is really robust.

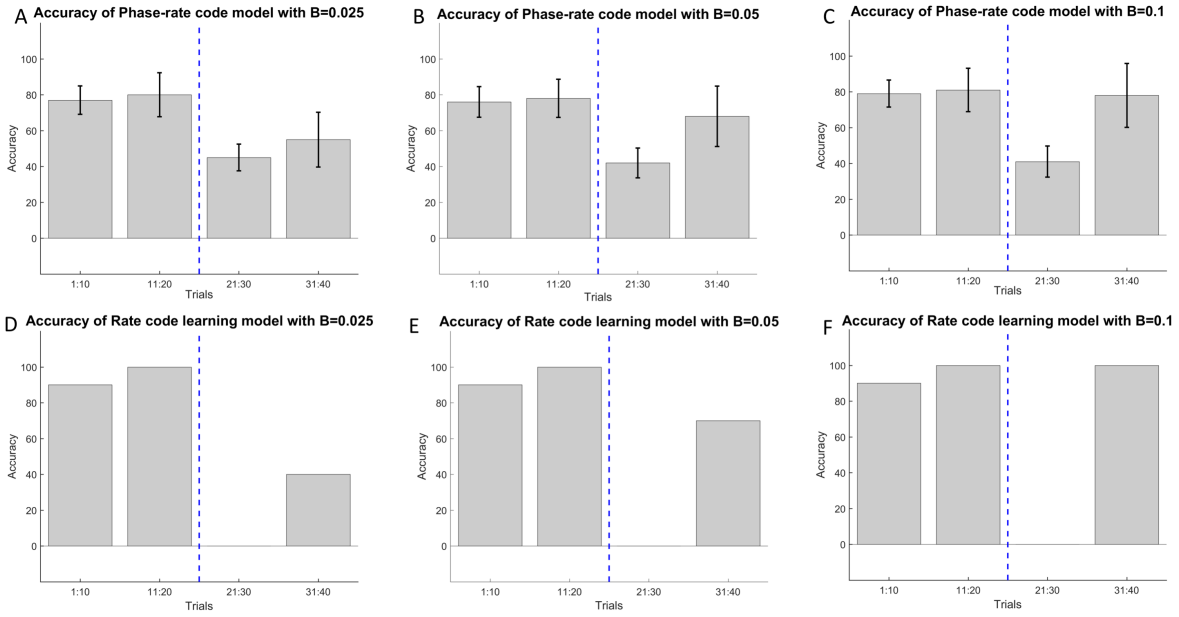

Figure 16, Results of Simulation 5. *A-C: Mean accuracy of the Phase-rate code model (Simulation 5.1) averaged across replications and means plotted per 10 trials. Error bars show the 95% confidence intervals (mean ± 2*standard error). Figure 16A shows the accuracy when the learning rate was slow (β=.025), B shows the performance with an intermediate learning rate (β=.05) and Figure 16C shows accuracy with a fast learning rate (β=.1). D-F: Mean accuracy of the Rate code learning model (Simulation 5.2), means plotted per 10 trials. Figure 16D shows the accuracy when the learning rate was slow (β=.025), E shows the performance with an intermediate learning rate (β=.05) and Figure 16F shows accuracy with a fast learning rate (β=.1). The vertical blue dashed line indicates the change in task demands.*

Model comparison

When we compare the performance of the models, we see (in Figure 14 and 16) that the model in Simulation 4, which only used phase synchronization, is definitely the most flexible one. Indeed, right after switching between task strategies (trials 21-30), it obtained the highest accuracy of all three models. The Rate code learning model (Simulation 5.2), is the least flexible one. Therefore, we can conclude that phase synchronization increases the flexibility of the model. However, a model that implements only phase synchronization, has its disadvantages too. In particular, it never seems to be able to reach 100% accuracy. Also, the confidence intervals in Figure 16A-C, and Figure 14 show that there were significant variations in accuracy levels when the model

implemented phase code units. This variation is not present in models using only anatomical learning. Thus, a model that only implements anatomical weight learning, shows a more robust performance.

A comparison of the total accuracy of the models is given in Figure 17. Here, we see that, for all learning rates, the total accuracy of the Rate code learning model in Simulation 5.2 is lower than the accuracy of the Phase code model in Simulation 4. When learning rate is slow ($\beta=.025$) the difference in total even reaches 20.25% (77.75%-57.5%). Considering the accuracy-levels per 10 trials in Figure 16D and Figure 14, we see that this difference is made in the trials following the switch in task demands. While the Phase code model achieves approximately the same accuracy in all trial bins, the rate code model suffers from a severe drop in accuracy after the task-rule switch.

The implementation of anatomical weight learning into the Phase code model in Simulation 5.1 reduced flexibility in the model. Also here, accuracy was reduced in the first 10 trials after the change in task demands. Still, the Phase-rate code model showed a better flexibility than the Rate code learning model of Simulation 5.2. This better flexibility is also reflected in total accuracy rates. The Phase-rate code model of Simulation 5.1 still shows a better general performance than the purely Rate code learning model. However, when anatomical learning is fast, the Rate code learning model shows an equally good general performance as the fast Phase-rate code model. This is because the Rate code learning model shows a more robust learning performance, which compensates for the lack in flexibility. Thus, combining phase code and anatomical learning in one model makes the model more flexible than a purely Rate code learning model. Especially when the learning rate is slow. In terms of robustness however, there is no significant indication that the robustness that is present in the Rate code learning model is in some way adopted by the Phase-rate code model. Hence, the Phase code model in Simulation 4 still shows a better general performance than the Phase-rate code model in Simulation 5.1.

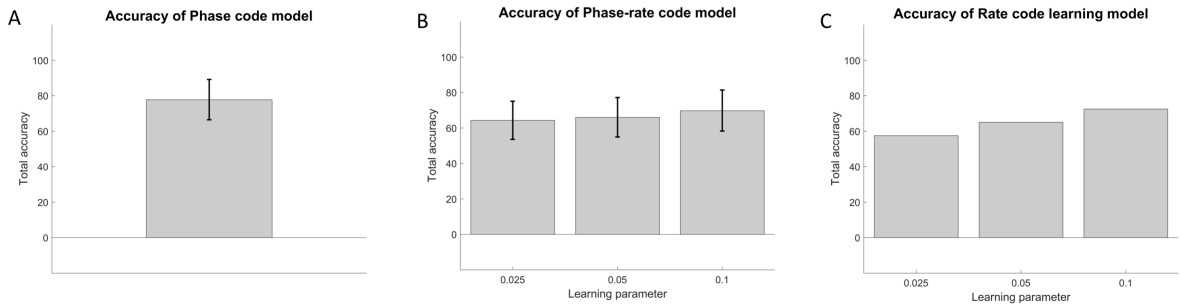

Figure 17. Comparisons of total accuracy of the models. *Figure 17A shows the total accuracy of the Phase code model of Simulation 4. B, shows the total accuracy for all learning rates of the Phase-rate code model in Simulation 5.1. Figure 17C shows the total accuracy for all learning rates of the Rate code learning model in Simulation 5.2.*

Discussion

In the current simulation, we compared two different processes to alter communication in the brain. That is, we compared a model that only used phase synchronization and a model that only used anatomical weight learning to a model that used both processes. Here, anatomical learning was implemented in the form of a delta rule (equation 20). Additionally, the learning rate was manipulated across 3 replications. The data revealed that the Phase code model in Simulation 4 is far more flexible than anatomical learning in the Phase-rate code and the Rate code learning models of Simulation 5.1 and 5.2. Implementing anatomical learning into a Phase code model, as in the Phase-rate code model of Simulation 5.1, makes the model less flexible but still more flexible than the Rate code learning model in Simulation 5.2. So, on one hand the Phase code model is more flexible than the Rate code learning model. On the other hand, the performance of the Rate code learning model is far more robust than the performance of the Phase code model in Simulation 4. In contrast to the Phase code model, the Rate code learning model is able to reach a 100% accuracy in certain trial bins. In the current study, the better performance of the Phase code model is mainly due to the first 10 trials after the change in task demands. If there would be more trials without another change in task demands the Rate code learning model would make up for the loss it made in that trial bin and perform equally well or even better than the Phase code model. This implies that phase code is especially helpful when there is a constant need to change between task strategies. When

anatomical learning is implemented in a Phase code model, as in Simulation 5.1, there are no significant improvements in the robustness of the models' performance.

In sum, results support our claim that neuronal synchronization can impose a way for flexible, effective communication on top of the anatomical communication structure. Then again, the communication through coherence principle (Fries, 2005, 2015), or at least in the way it is implemented in the current model, seems to lead to a decrease in the robustness of cognitive performance even when it is combined with anatomical learning.

General discussion

The current paper adopted a computational modelling approach to demonstrate how the CTC-principle (Fries, 2005, 2015) describes an additional way to flexibly alter communication in the brain that is complementary with general anatomical learning principles. As a start, we adopted the basic principles of the model of Verguts (2017), which described phase code processes in the Stroop task. A first extension of the model of Verguts (2017) was made in Simulation 1. Here, we explored how to achieve desynchronization between two oscillating units. Following the principle of binding by random bursts (Springer & Paulsson, 2006; Verguts, 2017; Zhou et al., 2005), we hypothesized that, like giving positively correlated bursts to two units can synchronize their oscillations, giving negatively correlated bursts might lead to desynchronization. Results confirmed our hypothesis. In sum, in Simulation 1 we have extended previous findings, in the sense that we demonstrated how the addition of random bursts to two oscillating units can not only lead to an increase in synchronization but also to desynchronization.

In our model, the units in the Processing module showed oscillations in the gamma-frequency. We chose this frequency because oscillations in the gamma range have been linked to several cognitive processing functions such as the binding of visual features (Gray & Singer, 1989), selective attention (Womelsdorf & Fries, 2007) and working memory load (Howard et al., 2003). Additionally, empirical findings indicate that gamma-synchronization has important consequences for both, short-range (e.g., Gray & Singer, 1989) and long range communication between neuronal groups (Gregoriou, Gotts, Zhou, & Desimone, 2009; Womelsdorf et al., 2007). In our model, gamma-oscillations were

modulated by a theta frequency-oscillation that originated in the ACC. In this way, the model is also consistent with a large amount of previous research suggesting a strong link between theta- and gamma-frequency, where faster gamma frequencies are embedded in and modulated by slower theta-oscillations (Canolty et al., 2009; Jensen & Colgin, 2007; Lisman & Jensen, 2013; Voloh et al., 2015). Moreover, the model also connected with empirical findings suggesting a strong importance of theta-oscillations in cognitive control processes (e.g., Cavanagh & Frank, 2014; Nigbur et al., 2011).

In Simulation 2, we demonstrated how synchronization and desynchronization are effective methods to alter the efficiency of neural communication. While synchronization of two neural oscillations can rapidly bind two units, desynchronization can lead to a fast unbinding. This was shown by implementing the CTC principles in the model. Here, we multiplied all input to a rate code neuron with a sigmoid increasing function (equation 9) which used the oscillation in the phase code units as one of the parameters. Because of this function, the oscillation in the excitatory phase code unit functioned as a constantly opening and closing door. When the oscillation was at its top, the door was fully open and all input could be received but when the oscillation was below zero, the door was fully closed, therefore no input could come in and no output could go out. Consistent with previous findings (Womelsdorf et al., 2006), this enhanced communication between units with synchronized oscillations led in our model to faster response times. By also introducing a second response-unit we demonstrated how phase synchronization is an effective way to implement biased competition in a computational model. Because synchronization was accomplished by bursts given by the ACC (in interaction with LFC), the model is in line with previous work suggesting that prefrontal areas exert biasing signals to other processing areas (Miller & Cohen, 2001).

Together, Simulation 1 and 2 illustrated how phase code can account for two of the main functions of the central executive in the working memory model of Baddeley and Hitch (1974). That is, (a) binding and coordinating different sources of information and (b) monitoring selective attention and behavioral inhibition. The first is accomplished by the binding by random burst principle and the latter is accomplished by making communication

in the brain selective; enhancing communication between task-relevant units and inhibiting communication between task-irrelevant units.

In Simulation 3, we addressed a second limitation of the model of Verguts (2017). That is, the model had no learning algorithm to find out what units it should bind. Therefore, we implemented an Actor-Critic structure into the model. In this way, the model could discover on its own what it must synchronize or desynchronize. Here, the Critic was adopted from the RVPM (Silvetti et al., 2011) and the Actor was a combination of two prefrontal brain areas, the ACC and LFC, which gave bursts to units in the more posterior processing areas. The ACC and LFC worked together in such a way that the model is consistent with the finding of high theta-power in the ACC during tasks requiring high cognitive control (e.g., Cavanagh & Frank, 2014; Womelsdorf et al., 2010) and with the notion of LFC as holder of task demand representations (Braver et al., 1997; Mac Donald et al., 2000). Since, in the RVPM (Silvetti et al., 2011), the Critic corresponds to the rostral part of the ACC and this Critic sends in the current model its information to the LFC, the model can also connect with theories and findings suggesting a strong link between the LFC and ACC during cognitive control tasks (e.g., Cohen, Botvinick, & Carter, 2000; Kondo, Osaka, & Osaka, 2004). To our knowledge, this is the first time that a computational model combined phase code processes with general rate code learning principles.

In Simulation 4, we went one step further by testing the model's performance on a reversal learning task (e.g. Clark et al., 2004; Izquierdo & Jentsch, 2012). In this task, the model must learn stimulus-response-reward associations. Suddenly, the task requires these associations to be reversed. Therefore, the model should be as flexible as it can, to first detect this change and then unbind the previously associated units and bind new ones. Results revealed that the model is indeed very flexible in responding to this change in the environment. This implies that phase code can also connect with the third aspect of the central executive functions; flexibly shifting between different task strategies.

Finally, in Simulation 5, we implemented anatomical learning in the model of Simulation 4 and constructed a basic Rate code learning model. In this way, we were able to compare anatomical learning with phase synchronization processes. The comparison

revealed that on one hand, phase synchronization is far more flexible than anatomical learning. The combination of both learning processes leads to a semi-flexible model. On the other hand, anatomical learning leads to a much more robust performance of the model but this robustness seems to almost disappear when both learning types are combined. In sum, both, rate code and phase code learning, have their advantages and disadvantages. While the main advantage of a Phase code model is that it is very flexible, the advantage of the rate code model is that its learning performance is more robust. Although the combination of the two processes (as implemented in the Phase-rate code model) leads to an increased flexibility compared to the Rate code learning model, it does not lead to an increased robustness compared to the Phase code model.

When overall accuracy was compared, the Phase code model of Simulation 4 showed the best overall performance on the reversal learning task and the Rate code learning model showed the worst performance. Still, for these comparisons some considerations might be made. First, the simulated task had a low number of trials. A more robust model, such as the Rate code learning model, will show a better performance when there are more trials between the changes of task demands. This implies that phase synchronization is especially helpful when the model must make regular task switches. Second, the Rate code learning model in Simulation 5.2 is a very basic model. Obviously, there are a lot of rate code models that show a greater flexibility (e.g., Holroyd & Coles, 2002; Reilly & Frank, 2006; Verguts & Notebaert, 2008). Therefore, one could argue that we overestimated the flexibility advantage of the Phase code model compared to the rate code model. Third, we might also have overestimated the robustness advantage of anatomical learning. The absence of robustness in behavior when using phase synchronization might, in part, be due to how we implemented this in our model. In the model, phase synchronization was done in response to a Hebbian error signal that was overwritten every trial. This stands in contrast with the very gradual, supervised learning mechanism we implemented for the anatomical learning. No matter how a connection between two areas is made, a gradual learning towards one defined goal will probably be more robust than a trial-by-trial learning based on a constantly changing error signal. Additionally, in the current model, the phase relation between two units is not independent

of the relation between those units and a third unit. This is most clearly demonstrated in Simulation 2 where Figure 8B shows that, although the R_2-unit did not receive hits, the synchronization levels of oscillations in S_1 and R_2 were constantly changing. This is because S_1 did receive hits in order to be bound with R_1. This also leads to a reduced robustness for the model. However, when we look at the data of Simulation 3, we see in Figure 11A and B that during trials 11-20 and trials 31-40 there was a significant difference in synchronization levels of the two response units with the stimulus unit. No matter how the model processes are implemented, in classical rate code models this difference in anatomical weights would always lead to a 100% accuracy. In Figure 10B we see however that in neither of these trial bins the Phase code model reached an accuracy of 100 %. This shows that no matter how the process of synchronization itself is implemented, the performance will probably never reach the same robustness as classical rate code models.

Limitations

Of course, this study suffers some limitations. First, we might think about a more thorough exploration of the model. We argued for example that the robustness of anatomical learning disappears when it is combined with phase code processes. However, the performance of a Phase code model can also be enhanced by increasing the value of the θ parameter in equation 9 (Verguts, 2017). It might be useful to explore different values of this parameter in the model of Simulation 5.1. In this way, we could do a search for the model that combines phase- and rate code processes, shows the greatest flexibility and most robust performance.

Second, in our model all gamma-oscillations had the same frequency. Brain signals are rather noisy and the gamma-frequency range is rather broad (at least 30-70 Hz). Therefore, the occurrence of two neuronal groups showing oscillations at the exact same frequency might be unrealistic. Intuitively, one could argue that a difference in frequency would have significant consequences for the communication between two neuronal groups. Yet, in the model of Verguts (2017) it is shown that, as long as the prefrontal areas keep sending hit signals, a small difference in frequency (2%) has no dramatic consequences for their communication. Also, previous modelling work has demonstrated how two oscillators can learn to achieve the exact same frequency by means of Hebbian Learning (Righetti,

Buchli, & Ijspeert, 2006). In our model this would mean that we use a Hebbian learning rule for the parameter *C* in equations 3 and 4. Thus, two neuronal groups might be able to reduce their frequency difference by means of learning. They can learn to oscillate on the same frequency. When the frequency difference is small enough, this does not have problematic consequences for their communication.

Third, although the model is in line with several theoretical and anatomical principles, it did not connect with the link between the occurrence of prediction errors and the error-related negativity signal (ERN; e.g., Holroyd & Coles, 2002). More specifically, it has been suggested that the ERN arises from a phase-resetting and amplitude-enhancement of theta-activity in the ACC (Trujillo & Allen, 2007). In the current model, we simulated theta-activity in the ACC but prediction errors did not lead to changes in these oscillations.

Fourth, although the current model describes oscillatory dynamics in the Processing module, the connectivity between and within the other modules remained purely anatomical. Research has suggested that phase relations between oscillations in the ACC and LFC might also be important for cognitive control. In a study of Cavanagh and colleagues (Cavanagh, Cohen, & Allen, 2009), the degree of synchronization between theta-oscillations in these two areas predicted the degree of ACC power on the next trial and the degree of posterror reaction time slowing.

Another principle we mentioned but did not implement in the model is the idea of communication through coherence with small inter-areal delays (Bastos et al., 2014). As we argued before, communication between two areas should be optimal when there is a small delay between their oscillations corresponding to the time the signal of the sending neuron needs to travel down the synapses and axons to the receiving neuron. In our model, we chose to ignore this idea and modelled the optimal situation as the situation with zero phase lag.

Another limitation of the model is that we combined anatomical learning and phase synchronization but did not demonstrate how phase synchronization can increase synaptic plasticity in Hebbian learning. Here, we chose to implement anatomical learning by means of the delta-rule (Widrow & Hoff, 1960) because we did not want anatomical learning to be influenced by phase synchronization. In this way, we could make a clear

comparison of the models. Then again, it might be interesting to also computationally explore the influence of phase synchronization on anatomical learning itself.

Lastly, the current study focused mainly on the demonstration of certain theoretical principles but did not focus on connecting with empirical data. This is a limitation since connecting to empirical data is one of the main requirements of a good computational model. Additionally, we argued that the implementation of phase code should allow us to connect to a very broad range of data but failed to demonstrate this here. Nevertheless, the current study succeeded to demonstrate several theoretical principles in a computational and comprehensive way.

Future directions

As mentioned in the previous section, the current study has still some issues that fall beyond the scope of this master thesis but might be interesting to address in the future. For example, we might first consider a more thorough exploration of all parameters. Second, we should attempt to connect with empirical neuropsychological data. Third, although we here combined phase synchronization with anatomical learning we did not explore the influence of phase synchronization on anatomical learning itself. Fourth, we might consider a stronger elaboration of the data-analysis. In the current model, we only focused on behavioral data and the dissimilarity function as a measure of synchrony. In order to connect the model's performance with empirical findings, we should in the future give a more detailed representation of the data.

Furthermore, the field of phase code computational models is still understudied in psychology. As demonstrated by Verguts (2017), the implementation of phase code process allows a model to connect with a broad range of neural and behavioral data. Therefore, it might certainly be interesting for future research to continue in this direction. The current model ignored several aspects of phase code processes that can be addressed in the future. First, implementing the idea of communication through coherence with small inter-areal delays might be interesting in the future since it makes it possible to increase the selectivity of communication, especially when there are a lot of units. Second, communication might also be selective in terms of the frequency of the oscillation. As demonstrated in previous research, it is possible to employ learning algorithms for the parameter C in equations 3 and

4 (Righetti et al., 2006; Scarpetta et al., 2002). In this way, a unit can learn to oscillate at a given frequency, making communication with that unit selective only for other units oscillating at the same frequency. Another option might be to continue in the direction of the current study and combine phase code processes with other learning principles of classical rate code models. For instance, we might make the task more difficult by making it a probabilistic reversal learning task. In this way, the model should use an algorithm to determine whether to further exploit one stimulus-response association or explore different options. This exploitation-exploration trade-off has in previous research been linked to the neuromodulators norepinephrine and acetylcholine (Aston-Jones & Cohen, 2005). So, we might expand the model to relate to an even broader range of empirical data. Additionally, future work might consider our finding that a combination of phase- and rate code processes in our model did not lead to the best performing model. Here, the Phase-rate code model did not show an increased robustness in comparison to the Phase code model. Therefore, future research might consider a hierarchical, higher order learning process that determines whether the model should rely on either the phase code processes or the rate code processes in order to show the best performance.

Conclusion

The current study proposed a computational model of cognitive control that combined phase- and rate code processes. The model demonstrated that phase synchronization and desynchronization can impose an additional way to flexibly change communication in the brain on top of anatomical learning. First, the study extended the previous finding of binding by random bursts (Springer & Paulsson, 2006; Verguts, 2017; Zhou et al., 2005), in the sense that we also demonstrated how random bursts can lead to desynchronization. Second, the current study is, in our knowledge, the first to combine both phase code and rate code learning principles. More specifically, the current model implemented Reinforcement Learning through an Actor-Critic structure (e.g. Houk et al., 1995) and supervised learning through the delta-rule (Widrow & Hoff, 1960). The model also explored interactions between phase synchronization and anatomical learning. Comparisons of the different models revealed that on the one hand, the Phase code model is indeed more flexible than the Rate code learning model. On the other hand, the Rate code

learning model showed a more robust performance. Combining rate- and phase code processes can compensate for the reduced flexibility of the Rate code learning model but not for the reduced robustness in the performance of the Phase code model.

In sum, the current study succeeded in demonstrating how the implementation of phase code processes in a computational model gives an additional way to describe a mechanism that can account for the three basic functions of the central executive in the working memory model of Baddeley and Hitch (1974). Additionally, we demonstrated, for the first time to our knowledge, how phase code processes can interact with general learning principles. Still, the current study leaves a lot of unanswered questions that open opportunities for future research.

References

Aston-Jones, G., & Cohen, J. D. (2005). AN INTEGRATIVE THEORY OF LOCUS COERULEUS-NOREPINEPHRINE FUNCTION: Adaptive Gain and Optimal Performance. *Annual Review of Neuroscience*, *28*(1), 403–450. http://doi.org/10.1146/annurev.neuro.28.061604.135709

Axmacher, N., Mormann, F., Fernández, G., Elger, C. E., & Fell, J. (2006). Memory formation by neuronal synchronization. *Brain Research Reviews*, *52*(1), 170–182. http://doi.org/10.1016/j.brainresrev.2006.01.007

Baddely, A. D., & Hitch, G. (1974). Working memory. In *The psychology of learning and motivation: advances in research and theory* (volume 8, pp. 47–89). New York: Academic press.

Bastos, A. M., Vezoli, J., & Fries, P. (2014). Communication through coherence with inter-areal delays. *Current Opinion in Neurobiology*, *31*(31), 173–180. http://doi.org/10.1016/j.conb.2014.11.001

Borisyuk, R. M., & Hoppensteadt, F. C. (1998). Memorizing and recalling spatial-temporal patterns in an oscillator model of the hippocampus. *Bio Systems*, *48*(1–3), 3–10. http://doi.org/10.1016/S0303-2647(98)00044-6

Braver, T. S., Cohen, J. D., Nystrom, L. E., Jonides, J., Smith, E. E., & Noll, D. C. (1997). A parametric study of prefrontal cortex involvement in human working memory.

NeuroImage, *5*(1), 49–62. http://doi.org/10.1006/nimg.1996.0247

Canolty, R. T., Edwards, E., Dalal, S. S., Soltani, M., Nagarajan, S. S., Berger, M. S., … Knight, R. T. (2009). High Gamma Power is Phase-Locked to Theta Oscillations in Human Neocortex. *Science (New York, N.Y.)*, *313*(5793), 1626–1628. http://doi.org/10.1126/science.1128115.High

Cavanagh, J. F., Cohen, M. X., & Allen, J. J. B. (2009). Prelude to and Resolution of an Error : EEG Phase Synchrony Reveals Cognitive Control Dynamics during Action Monitoring. *Journal of Neuroscience*, *29*(1), 98–105. http://doi.org/10.1523/JNEUROSCI.4137-08.2009

Cavanagh, J. F., & Frank, M. J. (2014). Frontal theta as a mechanism for cognitive control. *Trends in Cognitive Sciences*, *18*(8), 414–421. http://doi.org/10.1016/j.tics.2014.04.012

Clark, L., Cools, R., & Robbins, T. W. (2004). The neuropsychology of ventral prefrontal cortex: Decision-making and reversal learning. *Brain and Cognition*, *55*(1), 41–53. http://doi.org/10.1016/S0278-2626(03)00284-7

Cohen, J. D., Botvinick, M., & Carter, C. S. (2000). Anterior cingulate and prefrontal cortex: who's in control? *Nature Neuroscience*, *3*(5), 421–423. http://doi.org/10.1038/74783

Cohen, J. D., Dunbar, K., & McClelland, J. L. (1990). On the control of automatic processes: a parallel distributed processing account of the Stroop effect. *Psychological Review*, *97*(3), 332–61. http://doi.org/10.1037/0033-295X.97.3.332

Engel, A. K., Fries, P., & Singer, W. (2001). DYNAMIC PREDICTIONS : OSCILLATIONS AND SYNCHRONY IN TOP – DOWN PROCESSING. *Nature Neuroscience*, *2*(October), 704–717. http://doi.org/doi:10.1038/35094565

Fell, J., & Axmacher, N. (2011). The role of phase synchronization in memory processes. *Nature Reviews. Neuroscience*, *12*(2), 105–118. http://doi.org/10.1038/nrn2979

Fell, J., Fernández, G., Klaver, P., Elger, C. E., & Fries, P. (2003). Is synchronized neuronal gamma activity relevant for selective attention? *Brain Research Reviews*, *42*(3), 265–272. http://doi.org/10.1016/S0165-0173(03)00178-4

Frank, M. J. (2006). Hold your horses : A dynamic computational role for the subthalamic

nucleus in decision making. *Neural Networks*, *19*, 1120–1136. http://doi.org/10.1016/j.neunet.2006.03.006

Fries, P. (2005). A mechanism for cognitive dynamics: neuronal communication through neuronal coherence. *Trends in Cognitive Sciences*, *9*(10), 474–480. http://doi.org/10.1016/j.tics.2005.08.011

Fries, P. (2009). Neuronal gamma-band synchronization as a fundamental process in cortical computation. *Annual Review of Neuroscience*, *32*, 209–24. http://doi.org/10.1146/annurev.neuro.051508.135603

Fries, P. (2015). Rhythms for Cognition: Communication through Coherence. *Neuron*, *88*(1), 220–235. http://doi.org/10.1016/j.neuron.2015.09.034

Gray, C. M., & Singer, W. (1989). Stimulus-specific neuronal oscillations in orientation columns of cat visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, *86*(5), 1698–1702. http://doi.org/10.1073/pnas.86.5.1698

Gregoriou, G. G., Gotts, S. J., Zhou, H., & Desimone, R. (2009). High-Frequency, Long-Range Coupling Between Prefrontal and Visual Cortex During Attention. *Science*, *324*(5931), 1207–1210. http://doi.org/10.1126/science.1171402

Hammond, C., Bergman, H., & Brown, P. (2007). Pathological synchronization in Parkinson's disease: networks, models and treatments. *Trends in Neurosciences*, *30*(7), 357–364. http://doi.org/10.1016/j.tins.2007.05.004

Hebb, D. . (1949). The Organization of Behavior. A neuropsychological theory. *The Organization of Behavior*, *911*(1), 335. http://doi.org/10.2307/1418888

Holroyd, C. B., & Coles, M. G. (2002). The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. *Psychol Rev*, *109*(4), 679–709. http://doi.org/10.1037//0033-295X.109.4.679

Houk, J. C., Adams, J. L., & Barto, A. G. (1995). A model of how the basal ganglia generate and use neural signals that predict reinforcement. *Models of Information Processing in the Basal Ganglia*, *13*(July 1995), 249–270. http://doi.org/Cited By (since 1996) 262\rExport Date 9 May 2012

Howard, M. W., Rizzuto, D. S., Caplan, J. B., Madsen, J. R., Lisman, J., Aschenbrenner-Scheibe, R., … Kahana, M. J. (2003). Gamma Oscillations Correlate with Working

Memory Load in Humans. *Cerebral Cortex*, *13*(12), 1369–1374. http://doi.org/10.1007/BF00200803

Izquierdo, A., & Jentsch, J. D. (2012). Reversal learning as a measure of impulsive and compulsive behavior in addictions. *Psychopharmacology*, *219*(2), 607–620. http://doi.org/10.1007/s00213-011-2579-7

Jensen, O., Bonnefond, M., & VanRullen, R. (2012). An oscillatory mechanism for prioritizing salient unattended stimuli. *Trends in Cognitive Sciences*, *16*(4), 200–205. http://doi.org/10.1016/j.tics.2012.03.002

Jensen, O., & Colgin, L. L. (2007). Cross-frequency coupling between neuronal oscillations. *Trends in Cognitive Sciences*, *11*(7), 267–269. http://doi.org/10.1016/j.tics.2007.05.003

Jutras, M. J., & Buffalo, E. A. (2010). Synchronous neural activity and memory formation. *Current Opinion in Neurobiology*, *20*(2), 150–155. http://doi.org/10.1016/j.conb.2010.02.006

Koechlin, E., Ody, C., & Kouneiher, F. (2003). The architecture of cognitive control in the human prefrontal cortex. *Science (New York, NY)*, *302*(5648), 1181–1185. http://doi.org/10.1126/science.1088545

Kondo, H., Osaka, N., & Osaka, M. (2004). Cooperation of the anterior cingulate cortex and dorsolateral prefrontal cortex for attention shifting. *NeuroImage*, *23*(2), 670–679. http://doi.org/10.1016/j.neuroimage.2004.06.014

Li, Z., & Hopfield, J. J. (1989). Modeling the olfactory bulb and its neural oscillatory processings. *Biological Cybernetics*, *61*, 379–392.

Lisman, J. E., & Jensen, O. (2013). The Theta-Gamma Neural Code. *Neuron*, *77*(6), 1002–1016. http://doi.org/10.1016/j.neuron.2013.03.007

Mac Donald, A. W., Cohen, J. D., Stenger, A. V., & Carter, C. S. (2000). Dissociating the Role of the Dorsolateral Prefrontal and Anterior Cingulate Cortex in Cognitive Control. *Science*, *288*(June), 1835–1838. http://doi.org/10.1126/science.288.5472.1835

Miller, E. K. (2013). The “working” of working memory. *Dialogues in Clinical Neuroscience*, *15*(4), 411–418.

Miller, E. K., & Cohen, J. D. (2001). An Integrative Theory of Prefrontal Cortex Function. *Annual Review of Neuroscience*, *24*, 167–202. http://doi.org/10.1146/annurev.neuro.24.1.167

Nigbur, R., Cohen, M. X., Ridderinkhof, K. R., & Stürmer, B. (2011). Theta Dynamics Reveal Domain-specific Control over Stimulus and Response Conflict. *Journal of Cognitive Neuroscience*, *24*(5), 1264–1274. http://doi.org/10.1162/jocn_a_00128

O'Reilly, R. C., & Norman, K. A. (2002). Hippocampal and neocortical contributions to memory: Advances in the complementary learning systems framework. *Trends in Cognitive Sciences*, *6*(12), 505–510. http://doi.org/10.1016/S1364-6613(02)02005-3

Reilly, R. C. O., & Frank, M. J. (2006). Making Working Memory Work : A Computational Model of Learning in the Prefrontal Cortex and Basal Ganglia. *Neural Computation*, *18*, 283–328. http://doi.org/http://dx.doi.org/10.1162/089976606775093909

Righetti, L., Buchli, J., & Ijspeert, A. J. (2006). Dynamic Hebbian learning in adaptive frequency oscillators. *Physica D: Nonlinear Phenomena*, *216*(2), 269–281. http://doi.org/10.1016/j.physd.2006.02.009

Rodriguez, E., George, N., Lachaux, J.-P., Martinerie, J., Renault, B., & Varela, F. J. (1999). Perception ' s shadow : long- distance synchronization of human brain activity. *Nature*, *397*(February), 430–433. http://doi.org/10.1038/17120

Rosenblum, M. G., Pikovsky, A., & Kurths, J. (1997). From Phase to Lag Synchronization in Coupled Chaotic Oscillators. *Physical Review Letters*, *78*(22), 4193–4196. http://doi.org/10.1103/PhysRevLett.78.4193

Scarpetta, S., Li, Z., & Hertz, J. (2002). Hebbian Imprinting and Retrieval in Oscillatory Neural Networks. *Neural Computation*, *14*, 2371–2396. http://doi.org/https://doi.org/10.1162/08997660260293265

Silvetti, M., Seurinck, R., & Verguts, T. (2011). Value and Prediction Error in Medial Frontal Cortex: Integrating the Single-Unit and Systems Levels of Analysis. *Frontiers in Human Neuroscience*, *5*(August), 75. http://doi.org/10.3389/fnhum.2011.00075

Springer, M., & Paulsson, J. (2006). Harmonies from noise. *Nature*, *439*(January), 27–29. http://doi.org/doi:10.1038/439027a

Strogatz, S. (2003). *Sync: The emerging science of spontaneous order*. New York: Hyperion.

Sutton, R., & Barto, A. G. (1998). *Reinforcement learning: an introduction* (28th ed.).

Trujillo, L. T., & Allen, J. J. B. (2007). Theta EEG dynamics of the error-related negativity. *Clinical Neurophysiology*, *118*(November 2006), 645–668. http://doi.org/10.1016/j.clinph.2006.11.009

Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: the leaky, competing accumulator model. *Psychological Review*. http://doi.org/10.1037/0033-295X.108.3.550

Verguts, T. (2017). Binding by random bursts: A computational model of cognitive control. *Journal of Cognitive Neuroscience*, *29*(6), 1103–118. http://doi.org/10.1162/jocn_a_01117

Verguts, T., & Notebaert, W. (2008). Hebbian Learning of Cognitive Control : Dealing With Specific and Nonspecific Adaptation. *Psychological Review*, *115*(2), 518–525. http://doi.org/10.1037/0033-295X.115.2.518

Voloh, B., Valiante, T. a, Everling, S., & Womelsdorf, T. (2015). Theta-gamma coordination between anterior cingulate and prefrontal cortex indexes correct attention shifts. *Proc. Natl. Acad. Sci. U. S. A.*, *112*(27), 8457–62. http://doi.org/10.1073/pnas.1500438112

Widrow, B., & Hoff, M. (1960). Adaptive switching circuits. *IRE WESCON Convention Record*, *4*(1), 96–104.

Womelsdorf, T., & Fries, P. (2007). The role of neuronal synchronization in selective attention. *Current Opinion in Neurobiology*, *17*, 154–160. http://doi.org/10.1016/j.conb.2007.02.002

Womelsdorf, T., Fries, P., Mitra, P. P., & Desimone, R. (2006). Gamma-band synchronization in visual cortex predicts speed of change detection. *Nature*, *439*(7077), 733–736. http://doi.org/10.1038/nature04258

Womelsdorf, T., Johnston, K., Vinck, M., & Everling, S. (2010). Theta-activity in anterior cingulate cortex predicts task rules and their adjustments following errors. *Proceedings of the National Academy of Sciences of the United States of America*,

107(11), 5248–5253. http://doi.org/10.1073/pnas.0906194107

Womelsdorf, T., Schoffelen, J.-M., Oostenveld, R., Singer, W., Desimone, R., Engel, A. K., & Fries, P. (2007). Modulation of neuronal interactions through neuronal synchronization. *Science (New York, N.Y.)*, *316*(5831), 1609–12. http://doi.org/10.1126/science.1139597

Zhou, T., Chen, L., & Aihara, K. (2005). Molecular Communication through Stochastic Synchronization Induced by Extracellular Fluctuations. *Physical Review Letters*, *178103*(October), 2–5. http://doi.org/10.1103/PhysRevLett.95.178103