Decision-making in a proximate model framework: How behaviour flexibility is generated by arousal and attention

```
Call debug_interpolate_plot_save
(&grid_xx=MOTIVATION}<)))°>_COMPET_THRESHO
LD_<°)))>{CURVE_ABSCISSA,grid_yy=MOTIVATIO
NCOMPET_THRESHOLD_CURVE_ORDINATE, &
ipol_value=this%go_)<°)))>{s_arousal, <°)))>{
algstr  =  "DDPINTERPOL",  &output_file=
"plot_debug_arousal_gos_threshold_" // &
TOSTR (Time_S }<)))°> tep_Model_Current) //
&  TAG_MMDD  ()  //  "_"//  RAND_STRING
(LABEL_LENGTH, & LABEL_CST, LABEL_CEN) &
```

Master of Science in Biology: Theoretical Ecology

Lars Kristian Landsrød

Department of Biology

University of Bergen

June 2017

(30 Credits)

Takk til

En stor takk til Sergey Budaev som har utvist enorm tålmodighet i å lære meg å sette opp og bruke AHA modellen, i tillegg til å bidra med artikler, faglig diskusjon, IT-hjelp og gode tilbakemeldinger på oppgaven.

Jeg vil også rette en stor takk til Sigrunn Eliassen for utallige gjennomlesinger og gode tilbakemeldinger under skriveprosessen. Dine kritiske spørsmål og veiledning har vært til stor hjelp og har hevet kvaliteten på oppgaven mange hakk. Jeg føler jeg har lært utrolig mye av den tette oppfølgingen jeg har fått.

Til slutt vil jeg takke min fantastiske kone Betina for en uvurderlig støtte og tålmodighet disse månedene. Og en takk til Ludvik som alltid muntrer opp en sliten far på slutten av dagen! Jeg er veldig glad i dere begge to.

Lars Kristian Landsrød

Bergen, juni 2017

Sammendrag

Dyr må ta avgjørelser basert på begrenset informasjon og under tidspress. All tid som brukes på å utforske muligheter og innhente informasjon, er tid tapt som kunne vært brukt til å tilegne seg resurser. En realistisk modellering av dyr og deres adferd må ta høyde for at dyrene ikke treffer optimale avgjørelser, men er påvirket av lokale faktorer som lysforhold og konkurranse samt dyrenes egen tilstand, som sult og frykt.

En annen ting som har betydning for hvilke valg dyr tar i ulike situasjoner er deres personlighet. Relativt stabile trekk har blitt observert i en rekke arter. Et trekk som ser ut til å spille en viktig rolle i dyrenes håndtering av stressende situasjoner er *adferds fleksibilitet,* eller til hvilken grad de responderer på enderinger i miljøet.

I denne oppgaven har jeg tatt for meg en datamodell for beslutningstagning i fisk og studert hvordan vi kan skape variasjon i trekket adferds fleksibilitet hos individene i modellen. Adferds fleksibilitet ble målt etter hvor tilbøyelige fiskene var til å endre sin interne tilstand, eller «global organismic state» (GOS). Undersøkelsen ble gjort ved å justere på to parametere. Den første av disse parameterne kontrollerer hvor raskt motivasjonen synker etter å ha blitt aktivert (f.eks. av å oppdage et rovdyr). Den andre kontrollerer hvor mye informasjon som filtreres bort når fiskene er svært motiverte. Disse kalles henholdsvis «arousal dissipation factor» (ADF) og «attention modulation factor» (AMF).

Resultatene viser at både ADF og AMF er med å påvirke adferds fleksibiliteten hos fiskene. ADF påvirker hvor ofte fiskene revurderer sin nåværende tilstand, i lys av den tilgjengelige informasjonen. Fisker som revurderte oftere, var også mer tilbøyelige til å endre sin GOS. Selv om ADF i seg selv var tilstrekkelig for å skape variasjon i fleksibilitet, var filtrering av informasjon (AMF) avgjørende for at fiskene skulle vise spesielt rigid adferd, dvs. sjeldent endre sin indre tilstand.

Abstract

Animals must make decisions based on limited information and during a limited amount of time. The time spent exploring possibilities and sampling environmental information, means less time spent at actually gathering and securing recourses. A realistic modelling of animals and their behaviour must include organisms that do not make optimal decisions. Instead, they are constrained by local factors like illumination and conspecifics, as well as the animal's own state like hungry or afraid.

The animal's personality must also be taken into account when discussing decision-making. Relative stabile traits have been observed in many species, and allow us to predict to a certain extent, their behaviour in the future. In the context of coping with stressful situations, *behaviour flexibility* (or to what degree the animals react to environmental information) seems to be an important trait.

In this thesis, I have explored a computer model for decision-making in fish and studied how behaviour flexibility can be generated in the agents. Behaviour flexibility was measured as the fish' propensity to change their internal state, called the *global organismic state* (GOS). The study was done by adjusting two parameters. The first of these control the rate at which motivation declines, after first being elevated (e.g. by seeing a predator). The second parameter controls the filtration of irrelevant information, when the agent is highly motivated. These are called *arousal dissipation factor* (ADF) and *attention modulation factor* (AMF), respectively.

The results show that *both* factors affects behaviour flexibility in the fish. ADF influences how often the fish re-evaluate their current state, in light of the available information. Fish that re-evaluates more often were more likely to change their GOS. Even though the ADF was sufficient to generate variation in flexibility, information filtering (AMF) was required to generate particularly rigid behaviour, i.e. rarely changing their internal state.

Contents

1. Introduction

1.1. Animal Personality

Studies of animal behaviour have revealed that, for many species (e.g. birds, rodents, fish (Wolf, van Doorn, & Weissing, 2008)) there are inter-individual differences in behaviour that are relatively stable over time and across situations (Coppens, de Boer, & Koolhaas, 2010). The behavioural differences are often interpreted in terms of internal physiological or psychological mechanisms (e.g. fearfulness and aggression). These differences are thought to represent an analogue of human personality (Culum Brown, Krause, & Laland, 2011), as such we sometimes refer to them as *animal personality*. The term personality can be applied to a wide range of species (theoretically even to bacteria), if we define it as *"a broad domain of behavioural individuality involving the widest range of consistent and enduring behavioural traits"* (Culum Brown et al., 2011). Depending on the context and traits that are being investigated, the term can also appear as the *shyness-boldness continuum* (Wilson, Clark, Coleman, & Dearstyne, 1994), *behaviour syndromes* (Sih, Bell, & Johnson, 2004) or *coping styles* (Coppens et al., 2010).

The observed behaviour correlation in animals likely reflects specific genetic and physiological mechanisms that put a constraint on behaviour variability (Culum Brown et al., 2011). This would indicate that personality also has a heritable component (Bell, 2005) that is subject to natural selection. For instance, tropical poeciliid fish, *Brachyraphis episcopi,* drawn from a high-predation population exhibited bolder behaviour , than fish from a population where predation pressures were low, even when they were raised in laboratory environments (C. Brown, Burgess, & Braithwaite, 2007). Findings like these encourage motivates the search for genetic and physiological explanations of personality differences.

Research into individual differences in fish have often focused on their response to stress (Culum Brown et al., 2011). For instance, bold individuals (e.g. early explorers of novel objects) also exhibit reduced stress responses, which may express a link between personality traits and hormones such as cortisol (Culum Brown et al., 2011). This correlation has also been observed in other species of fish (e.g. Koolhaas et al., 1999; Overli, Winberg, & Pottinger, 2005; Schjolden, Stoskhus, & Winberg, 2005). The authors also point out that there is a relationship between stress responses and coping styles in carp (Huntingford et al., 2010).

Given that much of the behavioural differences found in fish have been observed in a stressful environment, coping styles is a well-suited framework for investigating these differences further. Indeed, a coping style is defined as a *"suite of behavioural and physiological responses of an individual that characterises its reactions to a range of stressful situations"* (Dingemanse, Kazem, Reale, & Wright, 2010).

1.2. Coping Styles, Behaviour Flexibility and Decision-making

The variation in coping styles falls on a continuum between what is often termed *proactive* coping and *reactive* coping (Coppens et al., 2010). Proactive individuals are characterized by high scores[1] on *routine formation* (Benus, Dendaas, Koolhaas, & Vanoortmerssen, 1990) and a low scores on behaviour flexibility (Bohus et al., 1987). As such, proactive individuals are less responsive to environmental stimuli. Instead, they rely on previously formed habits or routines. The reverse is true of reactive individuals, who are considered more flexible, relying more on environmental information in their decision-making. Other points of difference are attack latency, active avoidance, defensive burying, nest building, cue dependency and conditioned immobility (Koolhaas et al., 1999).

The different coping styles have an evolutionary history that has helped the individuals to form adaptive response patterns to the stressors in their natural habitat (Koolhaas et al., 1999). Therefore, the mechanisms for such coping styles should have a physiological basis that is, at least in part, heritable (Castanheira et al., 2017).

Behaviour flexibility is defined in Coppens et al. (2010) as *"the ability of an individual to directly respond and adjust its behaviour to environmental stimuli"*. The authors propose that behaviour flexibility is an important underlying factor that might explain the consistency of coping styles across situations. Thus, understanding individual differences of the proximate mechanisms that control this trait, could give us greater insight into how animal behaviour and coping styles emerges. The meaning of the term behaviour flexibility can vary slightly from paper to paper. For instance, in Wolf et al. (2008) behaviour flexibility points to an organism that displays different behaviour when confronted with the same environment repeatedly. The propensity to adjust its behaviour to the prevailing conditions is then referred to as

---

[1] Referring to the behavioural assessment made by human observers

responsiveness (Wolf et al., 2008). Here, I will use the term behaviour flexibility as defined by Coppens et al. (2010).

Whenever an animal is presented with new information, it can either respond to it or not. Striking the right balance between reacting to, and ignoring information is important. Redirecting attention to new stimuli may disrupt important task and reduce efficiency, while ignoring too much information can result in the animal not detecting important cues, such as the presence of a predator. The act of either reacting to or ignoring information can be viewed as a form of decision-making, albeit unconscious in many instances. Next, we turn to a brief review of some of the biological underpinnings that are involved in this kind of decision-making.

1.3. Survival Circuits, Arousal and Attention

Joseph LeDoux proposed to use the term *survival circuits* to describe the mechanism that allow organisms to make near optimal decisions, i.e. detect and respond to threats and opportunities in the environment, and thereby facilitating survival. *"Survival circuits help organisms survive and thrive by organising brain functions. When activated, specific kinds of responses rise in priority, other activities are inhibited, the brain and body are aroused, attention is focused on relevant environmental and internal stimuli, motivation systems are engaged, learning occurs, and memories are formed"* (LeDoux, 2012).

According to LeDoux (2012), the ultimate origins of survival circuits, can be found in early forms of life. Even single-celled bacteria are known to take in information (e.g. accept chemicals that have nutritional value) and outputs a motor response (e.g. retract from harmful chemicals). In more complex, multi-cellular organisms such survival capacities have increased in complexity and sophistication. Survival circuits involve defence, maintenance of energy and nutritional balance, thermoregulation and reproduction, as a minimum. (LeDoux, 2012).

A key feature of survival circuits is that they constantly interact in a shifting environment so as to best meet the challenges and opportunities face by the organism (LeDoux, 2012). Some circuits monopolize the brains resources, while others are actively suppressed. As LeDoux puts it, over time the need for energy rises in priority and will outweigh the threat of a predator if the need for food is ignored long enough. Survival circuit activation can be reduced in two

ways: actively suppressed by a different circuit, or the goal of the survival circuit is reached and the innate responses terminate the activation (e.g. food is eaten or safety is reached).

Survival circuit activation leads to a rise in arousal from the central nervous system, as well as the motivational systems which lead to goal directed behaviours. All this activation in the body leads to a state *"in which brain resources are coordinated and monopolized for the purpose of enhancing the organism's ability to cope with the challenges and/or benefit from opportunities"*. This is referred to as a Global Organismic State (GOS) (LeDoux, 2012). The GOS makes the organism more attentive and sensitive to stimuli that are relevant for the situation, as well as activating relevant memories and previously learned instrumental responses.

The GOS does not cause arousal activation directly, but rather coordinates the arousal towards a unified goal. As such, the GOS can be viewed as the *qualitative* component of motivations (the direction). The *quantitative* component of motivation (the magnitude) is a matter of arousal.

Weil, Zhang, Hornung, Blizard, and Pfaff (2010) have proposed that the vertebrate nervous system has a function that initiates behavioural activation of large numbers of responses. This function has been termed generalized arousal. *"General arousal is higher in an animal or human being who is: (S) more alert to sensory stimuli of all sorts, and (M) more motorically active, and (E) more reactive emotionally"* (Pfaff, 2006). As such, it seems fair to suggest that general arousal is higher in highly motivated individuals. Neurons involved in generalized arousal mechanisms receive sensory inputs from both the external environment, as well as the internal condition of the organism, and rapidly activates arousal states within the animal. This activation, in turn, facilitates more specific, motivated behavioural responses (Weil et al., 2010).

Pfaff (2006) propose that brain arousal is promoted by unexpected events. For instance, the nucleus accumbens in humans (which is involved in reward appraisal) shows a greater activation when the rewarding stimuli where unpredicted (Pagnoni, Zink, Montague, & Berns, 2002). Similarly, in a monotonous environment animals lose arousal and become less alert (Pfaff, 2006). As such, Pfaff seems to propose a kind of feedback loop in which relatively unexpected events leads to an increase the general arousal, which in turn increases sensory alertness to all stimuli (see definition of generalized arousal). This highlights the dynamic interaction between arousal and attention in how information is perceived and responded to.

According to the *feature integration theory* of attention (Treisman & Gelade, 1980), our attention system can process general features quickly and automatically, but we need deliberate focus in order to identify objects. The fact that we cannot process objects automatically and instantaneous, has important and fitness-related consequences, as almost all of the important (visual) cues are objects, e.g. predators, prey or pieces of food, conspecifics etc. Moving our focal point around our field of view takes time. Time is a limited resource, especially in a life-or-death situation. Thus, it is important to be able to focus on the right kind of information, while ignoring other kinds (Lavie, 2005). This process is referred to as *selective attention* (Moran & Desimone, 1985).

1.4. Behaviour control in a proximate architecture framework

Early attempts to model animal behaviour assume that animals arrive at optimal solutions (e.g., Fisher, 1930; Lotka, 1925) to environmental challenges without considering potential constraints (Andersen, Jorgensen, Eliassen, & Giske, 2016). Andersen et al. (2016) point out that animal's decision-making is *"limited by imperfect information, imperfect ability to analyse it, imperfect ability to foresee consequences of the alternative behavioural options, as well as time constraints"*. Thus, they argue that the decision-making process, instead of being optimal, follows a flexible architecture which they sum up in the following way:

Sensory information is monitored, both externally and internally. The information is used to rank order different tasks (e.g. feeding, escaping). The strongest task is set as the dominant global organismic state, causing the individual to restrict its attention to mostly task- relevant information. This results in a more effective decision-making, because the organism is now using a narrower subset of relevant information.

The term *Proximate architecture for decision-making* was introduced by Eliassen et al (2016) to emphasise that the architecture involves connections between several kinds of information, which results in behavioural and physiological responses. Behaviour models based on an architectural approach places much weight on the *"chain of events from immediate perceptions to instrumental behaviour"* (Eliassen, Andersen, Jorgensen, & Giske, 2016). The proximal architecture framework is based on qualitative architecture of danger avoidance in LeDoux (2012, 2015) as well as Giske et al. (2013); (2014), but with some additions that will be explained below. As in Giske et al. (2014), the agent's perception, neuronal responses and developmental modulation are influenced by the genome. The sum of these factors determine

5

the global organismic state, resulting in attention restriction (Mendl, 1999) to other stimuli and limits the range of behavioural responses.

The competition for dominance over the organism's phenotype follows a *"winner-takes-all"* format (Eliassen et al., 2016) without a compromise in attention between states or within the same state. According to Andersen et al. (2016) attention restriction may be more gradual. They point to studies involving graded attention in three-spine sticklebacks. Sticklebacks showed a change in prey-density preference when they moved from a starved state towards a satiated state (Heller & Milinski, 1979). The starved sticklebacks preferred dense swarms of *Daphnia,* but changed to lower densities as their motivation for feeding decreased (they became less hungry). This was attributed to the attention cost imposed by feeding on a dense population. A strong demand on attention leaves little to be spared for other tasks like predator detection (Andersen et al., 2016). Similarly, when a shadow in the shape of a predator where introduced at the beginning of the experiment, the sticklebacks preferred low prey-density, even when hungry (Heller & Milinski, 1979). In the current AHA model, agents do not alter their foraging strategies in face of potential predators, meaning that there is no state for a "cautious" forager. Instead, the agents can mitigate a high-risk situation by quickly switching state, e.g. from foraging to escaping. The agents switch states more readily when they perceive the environment as "risky" compared to a "safe" environment.

1.5. Aim

The animal's responsiveness to new environmental information (behaviour flexibility) has implications for their decision-making process. The *Adaptive Heuristic and Architecture* (AHA) model that is presented in this thesis, use a proximate architecture framework to study animal decision-making. This means that the agents in the model make decisions based on the limited information that is available, rather than acting as omniscient beings as has been the case in many previous behaviour models. For the simulation experiments, the aim was to use the model study how interactions between *arousal* and *attention* can lead to individual differences in behaviour flexibility.

In this context, behaviour flexibility was measured as the propensity to change ones internal state in response to new environmental information. Arousal is here considered a qualitative measure of motivation in which high arousal corresponds to a strong motivation. In the model, arousal spontaneously dissipates with a rate that is dependent on a parameter called the *arousal*

dissipation factor (ADF). Furthermore, attention allows the agents to place more weight on certain kinds of stimuli than others. This kind of selective attention is controlled by the parameter *attention modulation factor* (AMF).

The goal for this thesis is to investigate how 1) the arousal dissipation factor and 2) the attention modulation factor interact to produce individual differences behaviour flexibility, in the AHA model.

2. Material and Methods

2.1. Introducing the new AHA Model

The simulations for this thesis were done using a new version of the Adapted Heuristics and Architecture (AHA) model presented in Giske et al. (2013). The model has since then been modified, but not published in any academic paper. First, I present a general overview of the *new* AHA model. Next, the model in Giske et al. (2013) is presented and some relevant differences and similarities are pointed out.

The AHA model is a multilayer modelling framework focusing on animal decision making and adaptive behaviour. The organisms in the model have a genome, rudimentary physiology, a hormonal system, a cognitive architecture and behavioural repertoire. The organisms navigate in a stochastically spatial explicit environment that includes a host of variables such as light conditions, predators, food and conspecifics to name a few.

Central to this thesis is the cognitive architecture, which is based on a set of motivational systems. These systems act as a common currency for *decision-making*. The organisms make decisions based on a predictive assessment of external and internal stimuli, which interact with the agent's motivational state at any given moment. The assessment is subjective to each agent, and the organism selects behaviours from the available repertoire in an effort to reduce its motivational arousal. Thus, decision-making is based on predicting one's own internal state as a consequence of the choice that is made.

The virtual population of organisms can evolve as only individuals that survive and reproduce will leave descendants in the next generation, with new variants introduced by potential mutation of the parents' genome. As such, this modelling machinery can aid our understanding of the evolution of decision-making mechanisms, personality, emotion and behavioural plasticity within a realistic and fully controllable ecological and evolutionary framework. The ultimate aim of the AHA model is to create a complex and extensible virtual (*in silico*) digital lab for studying adaptation and evolution. For more details see: (Weblink, 2017a)
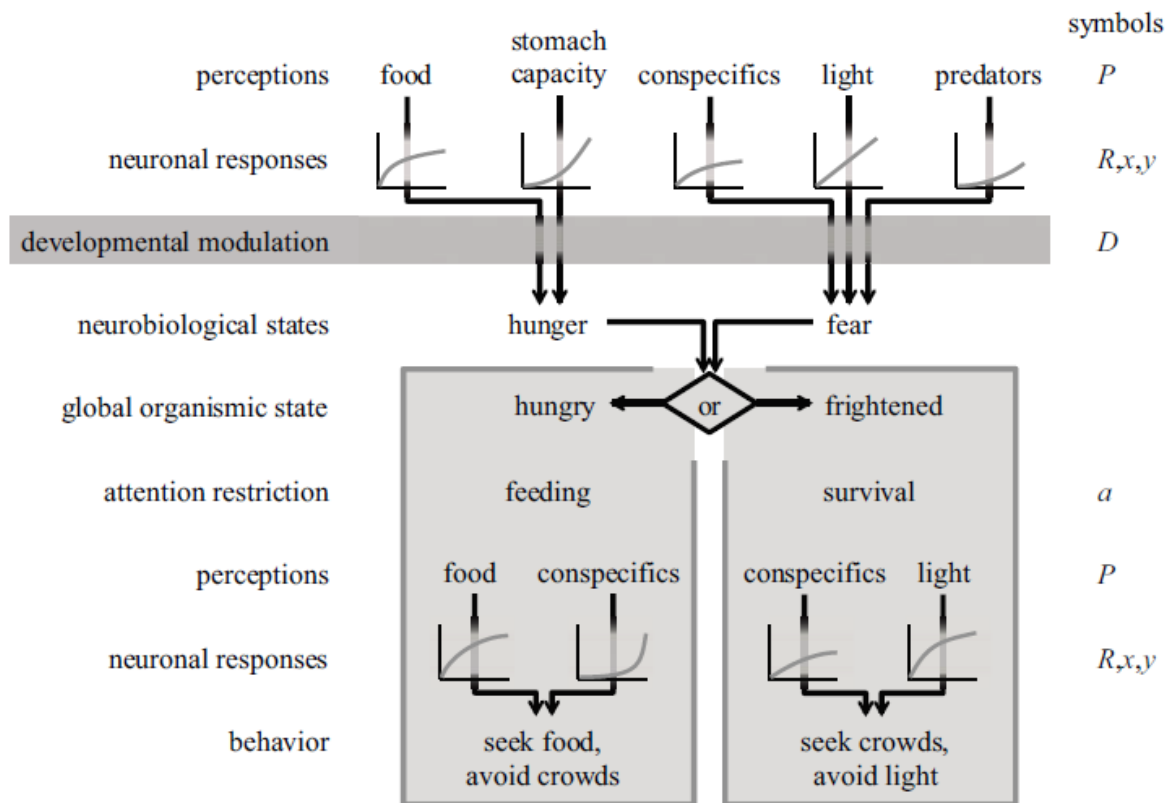
Figure 1: A generalized example of the proximate architecture for decision-making. Stimuli feeds into all neurobiological states simultaneously. Although only two states are presented here, the current AHA model includes four states: Hunger, Escape, Freeze and Reproduction. Stimuli are converted to neuronal responses, R, and modulated by modulation factors, D. Each stimuli can feed into several states. The most dominant GOS "wins" the competition for control of the agent's physiological and behavioural priorities. There is no graded response within a state or between states (i.e. a winner-takes-all fashion, without compromise with other states). From Giske et al. (2013)

Perceptions, *P*

The basis for the organism's decision-making is the perception of internal and external stimuli (Fig.1). Stimuli signals can vary in strength, rather than being binary (e.g. a predator in close proximity evoke a stronger stimulus signal than a distant predator). The fish takes in three types of perception: [1] external spatial perception objects (depending on the visual range), [2] internal perception objects (depending on the body) and [3] light and depth perception (which get its environmental factor directly) (Budaev, 2017). The external spatial perception objects such as food items, conspecifics and predators are truly "local" and proximate, meaning that the perception of such items depend on individual visual range (Budaev, 2017). Thus, the agents not only have local perception limited by their perceptual capacities and the local environment, (e.g. the visual range is reduced at low illumination), but their knowledge (memory) is limited

by the local contexts and environments they experienced before. In this regard, the AHA model is different from traditional simple models, which usually assumes that individuals are omniscient.

Neuronal responses, *R*

The stimuli signals are fed into *neuronal response functions* and converted to neuronal responses, *R*. The response depend on the strength of the stimulus, *S,* and two heritable genes, *x* and *y*.

$$R \ = \ \frac{(S/y)^x}{1+ (S/y)^x}$$
[1]

In the new model, the genes can take integer values from 1 to 10000 which constitutes as different alleles[2]. Genes are arranged into arrays that represent chromosomes. These chromosomes are further split into pairs that represents homologue chromosomes. As such, the genome is diploid (polyploid genomes can also be modelled). There is a fixed correspondence between a gene and each specific trait that are set by the Boolean genotype to phenotype matrices[3]. The response functions allow for graded responses to weak signals and saturation of strong signals (Giske et al., 2013). In the new model, [1] can be extended to capture multigene inheritance with arbitrary number of additive genetic components[4]

Developmental Modulation, *D*

Modulation genes also regulate the strength of the neuronal response. The modulatory system can up- or down regulate the signal strength throughout the agent's life, making it more likely or less likely that a specific GOS will prevail (Giske et al., 2013). For example, organisms might prioritize feeding and growth early in life until they reach a certain body mass, then they might favour reproduction.

---

[2] Link in references: (Weblink, 2017b)

[3] Link in references: (Weblink, 2017c)

[4] Link in references: (Weblink, 2017d)

Neurobiological states

After a signal has passed the attention modulation, converted to the neuronal response function [1] and modulated by the relevant modulation systems, the signal is passed through to a *neurobiological state* (NBS). In Giske et al. (2013) (Fig.1), two such states are presented: hunger and fear. The strongest of these neurobiological states dictates which *Global Organismic State* (GOS) to adopt – hungry or frightened (Fig.1).

Here, the new model deviates from Giske et al. (2013) as the "fear" neurobiological state is represented as *two* separate neurobiological states in the new model: active avoidance (fleeing behaviour) and passive avoidance (freezing behaviour). Additionally, the new model also includes a neurobiological state - Reproduction. Consequently, there are also *four* Global Organismic States, rather than two. The names of the GOS' are the same as the name of the corresponding neurobiological states: active avoidance, passive avoidance, hunger and reproduction. This is because the new model has been reformulated in terms of proximate behaviours rather than high level "decisions". The agents perform specific behavioural responses (e.g. walking, eating etc.) like real organisms.

In the new model, the strength of the neurobiological states is expressed in terms of arousal, *a*. Each of the four neurobiological states is associated with its own distinct arousal level that is recalculated for every time step. The arousal level of the various neurobiological states does not have a "memory" or "inertia" in the sense that the arousal level of a given step influences the arousal level in the next. Instead, arousal level is based on the retrieved and processed information at that point. Collectively, the arousal level of the neurobiological states is referred to as the *NBS arousal*, a_{NBS}. Any single stimulus can simultaneously generate more than one neuronal response for each GOS component.

Changing Global Organismic States

In Giske et al. (2013) the organism adopts the *Global Organismic State, GOS* that receive the strongest signal from the *neurobiological state*. The current GOS always corresponds to the strongest neurobiological state at any given time. Consequently, the organisms are prone to shifting back and forth between two states, when confronted with equally salient or distracting stimuli. This does not constitute a realistic, nor adaptive solution to conflicting needs. Rather, organisms should (and do) try to complete one task before taking on a new in turn. This problem

is addressed in the new model with the introduction of a new function: the dynamic threshold (see details below).

In the new AHA model, the current GOS is associated with its own arousal level, called the *GOS arousal, a_{GOS}*. To be clear, GOS arousal is *not* the same as the NBS arousal. While the NBS arousal fluctuates independently from one time step to the next, the GOS arousal can sometimes depend on the previous time step. As such, the GOS arousal sometimes exhibits an "inertia-like" property that influences how fast the arousal level declines. This is explained in more detail later. The GOS arousal can be viewed as a *quantitative* expression of the current motivation while the specific GOS is a *qualitative* expression of motivation. For instance, a hungry individual can be either peckish (low GOS arousal) or starving (high GOS arousal) within the same specific GOS (hunger). In both cases, the organism is motivated for finding food, but the strength of the motivation is different.

For every time step, the *GOS arousal* is compared to each of the *NBS arousals*. To avoid constant shifting of GOS the model requires that that the difference between the NBS arousal and the GOS arousal exceeds a threshold value. If not, the old GOS is retained and the minor fluctuations are ignored. The threshold is a function that is inversely related to the absolute value of GOS arousal (Fig.2), which makes it *dynamic* rather than static. It is therefore called the *Dynamic Threshold* (DT). The individual will switch GOS if:

$$a_{NBS} - a_{GOS} > t(a_{GOS}) \qquad\qquad [2]$$

Here a_{NBS} is the strongest of the four NBS', a_{GOS} is the current GOS arousal and *$t(a_{GOS})$* is the dynamic threshold as a function of the GOS arousal (Fig.2). If the dynamic threshold is reached, there are two possible outcomes: a *switch* or a *re-election*. A switch occurs when the strongest NBS is different from the current GOS. As such, a switch entails a shift in the motivational goal. A re-election occurs when the strongest NBS is of the same type as the current GOS. In this case, the agent's goal is unchanged. In both cases, the GOS arousal increases to the level of the NBS arousal.
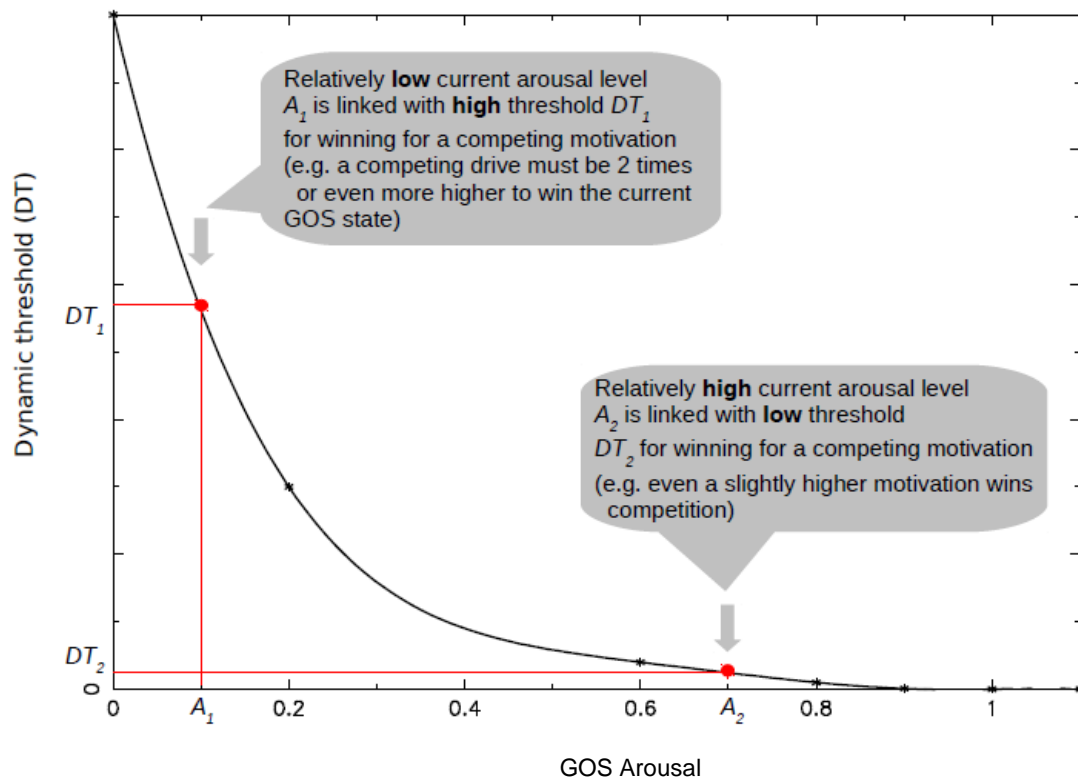
Figure 2: The Dynamic Threshold, DT, is a function of GOS arousal. The dynamic threshold dictates the level of arousal that is required for switching GOS (scale is 0:1). The function $t(a_{GOS})$ gives a higher threshold (DT_1) when GOS arousal is low (A_1), and a lower threshold (DT_2) when GOS arousal is high (A_2). (Budaev, S. unpublished)

Switching from one GOS to a different kind, entails a qualitative shift in motivation that moves the organism towards other goals. The new goal is likely to activate a new set of behavioural responses. A re-election does not cause a change in the quality of motivation, but it does alter the *strength* of the motivation. Since the organism can re-elect the same kind of GOS successively, there is theoretically no upper limit to how long the organism can remain motivated for the same goal.

On the other hand, a strong motivation must never be able to "seize control" of the organism, even if the current level of arousal is high. The prevailing motivational state spontaneously dissipate over time, which in natural neurobiological systems corresponds to exhaustion of neurotransmitters, cost of long-term neuronal excitation and stimulus habituation. The conditions that leads to high motivation and specific GOS (e.g. hunger based on starvation) can however, last for a long time and recur multiple times.

I will refer to switches and re-elections collectively as *re-evaluations.* The term re-evaluations are meant to emphasise that deciding to remain focused on a certain goal is an *active* decision,

13

rather than being an expression of apathy or faulty sensory systems. The process of periodically re-evaluating priorities without *necessarily* changing those priorities, can be considered analogous to the concept *vigilance* in e.g. J. S. Brown (1999) in the sense that the organism is periodically sampling information from the environment and that such sampling can represent a trade-off between feeding efficiency and predator detection. Here, he also points out that hungry salmon where more willing to take risks than well-fed salmon. This phenomena is replicated in the model, in that some fish are willing to forgo detection of threat signals in order to reduce distraction from feeding. This is explained further in the result section.

So far, we have considered what happens in a time step when the DT is reached - the GOS is either switched or re-elected. There is also a third option, that the dynamic threshold is not reached. In this case, the organism repeats the same GOS in the next time step, we call a *GOS repetition, r_{GOS}*. In a GOS repetition, the specific GOS is retained (similar to a re-election), but the GOS arousal does not increase, rather it decreases (see next section). The GOS repetition is 1 after a new GOS has been adopted and as long as the switching criteria is not reached [2], increases by one each time step.

Reducing the GOS arousal

In the new model, the strength of an organism's motivation is fluctuating every time step, reflecting the constantly changing stochastic stimulus environment. Processes involved in increasing GOS arousal (i.e. the motivation strength) has already been presented (Fig.1). Here we discuss the two ways GOS arousal decrease.

The first and most intuitive way is by reaching the goal set forth by the global organismic state, e.g. obtaining food. The agents in the model are programmed to choose the option that provides the greatest reduction in arousal. This means that when confronted with two food items, the fish chooses the item that is thought to result in the greatest reduction in hunger-related arousal. This approach is inspired by Drive-reduction theory, in which an excitatory state such as hunger is seen as a "homeostatic disturbance" that must be balanced (Seward, 1956).

The second way of reducing arousal involves a spontaneous reduction in the GOS arousal. As explained earlier, reducing the agents' GOS arousal is necessary to allow for re-evaluations and changes in GOS when arousal is initially high. In the AHA model, GOS arousal is reduced by the *Arousal Dissipation Factor*, (ADF) (See appendix A2). The ADF controls the rate of arousal dissipation according to the following equation:

$$a_{GOS}(n + 1) = d_a \times a_{GOS}(n) \tag{3}$$

Here $a_{GOS}(n)$ is GOS arousals as functions of time steps n, the ADF has symbol d_a and is always a real number from 0 to 1, ensuring that the arousal a_{GOS} is reduced over time

The organism's individual genome, physiology and current state interact to produce arousal fluctuations. So far, we have looked at processes for increasing and decreasing arousal (i.e. neurobiological states and arousal dissipation factor), in response to situational cues. Next, we turn to one of the psychological consequences that are linked to a high arousal level – attention restriction.

Attention restriction

Attention restriction is also included in the model in Giske et al. (2013). The organism's GOS determine the motivational goals as well as the behavioural response, thus, for any given GOS, stimuli can regarded as relevant or irrelevant depending on the context (e.g. food items can be ignored when the motivation for escaping predators is strong). A realistic view of decision-making recognizes that sampling time and processing speed are important constraints that prevent organisms from making optimal decisions (Andersen et al., 2016).

In the new model, attention restriction is implemented by multiplying all stimuli irrelevant for the current GOS with the *Attention Modulation Factor,* (AMF) (Fig.3). The term "irrelevant" does not point to the survival value of the stimuli, rather it is in reference to the current motivational state (GOS). As such, ignoring "irrelevant" information can be a mistake in terms of survival. The AMF uses the symbol m_a and is normally a function of the GOS arousal. The signal strength of irrelevant stimulus, s_i, is affected by the AMF according to:

$$s_r = s_i \times m_a(a_{GOS}) \tag{4}$$

The restricted stimulus, s_r is fed into the neuronal response function (Fig.1). Since $m_a(a_{GOS})$ is a number between 0 and 1, the numerical value of s_i is either reduced or unchanged, depending on a_{GOS}. During high arousal, the strength of irrelevant stimuli is almost completely diminished (Fig.3). This makes switching unlikely to happen during highly aroused situations.
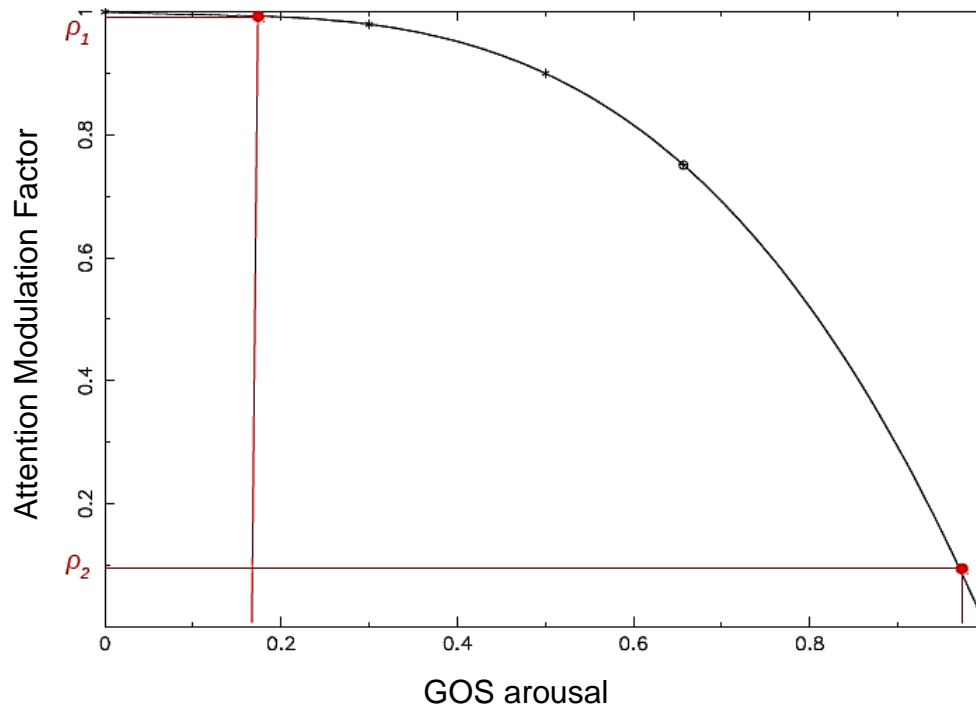
.

Figure 3: The Standard attention restriction curve in the new AHA model. The attention modulation factor (AMF) is a function of GOS arousal. In a relaxed context (ρ_1) the GOS arousal is low and attention restriction is minimal. During times of high motivation (ρ_2), irrelevant perceptions are restricted by as much as 90 % in this example. (Budaev, S. unpublished)

2.2.Exploring interaction between arousal dissipation and selective attention

<u>The subroutine</u>

Exploring the interactions between arousal dissipation and selective attention was done using a separate subroutine[5] that used a subset of the AHA modelling framework. For every time step, the agents perceive new stimuli through the process described earlier (Fig.1), the agents also moved in space in a random direction and of a random length (varying around a Gaussian curve). This approach provided a simplistic standardized experimental situation, without depending on consistent or long-term changes in motivation or overall condition of the agents, e.g. success in finding food and linked level of hunger, behaviour and growth with further complex cascade effects.

<u>The environment</u>

When we consider an agent's responsiveness to stimuli, it is instructive to know how the virtual environment is changing. The environment in the AHA model changes each time step as the agents move through it (spatially and temporally). New food items, conspecifics[6] and occasional predator can appear and illumination level changes as the agents move up or down in the water column. In the simulations considered here, the environment remained constant as all the food items, conspecifics and predators were uniformly distributed and the simulations were too short for any environmental fluctuations (e.g. diurnal change in the illumination) to occur.

<u>Data collection</u>

In this thesis, I chose to measure *behaviour flexibility* by using the frequency of GOS switching as a proxy. Behaviour flexibility in this context points to the animal's ability to respond and adjust to environmental signals. Although I am not recording any change in behaviour directly, I expect the internal GOS "competition" to reflect the agent's responsiveness to environmental signals. See discussion for more.

---

[5] The subroutine was written and added by Sergey Budaev. See appendix A1 for the complete code.

[6] The maximum population size is 10 000 in the AHA model

A switch is defined here as a GOS shift that also entails a shift in the motivational goal, and is presented as the probability of switching GOS in any given time step. An organism that shows high behaviour flexibility is more likely to switch GOS than an organism with low behaviour flexibility. The probability of switching GOS was collected in both Part I and Part II. In addition, I also recorded the probability of *re-evaluating* the GOS in a given time step. A re-evaluation occurs when the arousal level of a neurobiological state crosses the dynamic threshold (Fig.2), but because it points towards the same motivational goal as the current GOS, there is no change in the organism's motivation.

In Part I, I also collected data for *longest GOS chain*. This is defined here as the longest duration an organism retains the same GOS. As such, re-electing the same GOS over and over does not "break the chain". This type of data was added to gain some additional insight into the relationship between the arousal dissipation factor and behaviour flexibility.

Part I: Exploring the effect of the Arousal Dissipation Factor

In the new AHA model, there are two ways of implementing the arousal dissipation factor (ADF). In Part I, both of these were tested to see if the measurements of behaviour flexibility were very different for the two types.

First, I let the ADF be a *constant* between 0 and 1 (d_a = constant), which means that the agent's GOS arousal drops off at a constant rate over time. A lower ADF translates to a faster reduction in GOS arousal. To explore the range of behaviour flexibility in this condition, I performed 10 simulations using 10 different ADF constants. The number of GOS switches, number of re-evaluations and the longest GOS chain was recorded. For the data for the longest GOS chain only, this process was repeated, and the mean results from the two sets of simulations were used. This was done to reduce the role of random "noise" in the data set. The GOS chain data were analysed using statistical analysis[7] of the standard deviation of residuals and Akaike Information Criterion (see appendix A4 for R script).

---

[7] The analysis was done by Sergey Budaev.

Second, I let the ADF be a *function* of GOS repetitions (Fig.4). In this case, GOS arousal is reduced at an *accelerating* rate with increasing number of GOS repetitions (Table 2). All agents in a single simulation had identical ADF functions controlling their rate of dissipation. See Figure 4 for a graphical representation of the three functions. The three functions represent a *slow*, *intermediate* and *fast* rate of dissipation. I performed one simulation for each of the three ADF functions (in later versions of the AHA model, the ADF will be implemented as individual genes inherited from parents). For each of these simulations, I recorded the probability of switching GOS and the probability of re-evaluating.
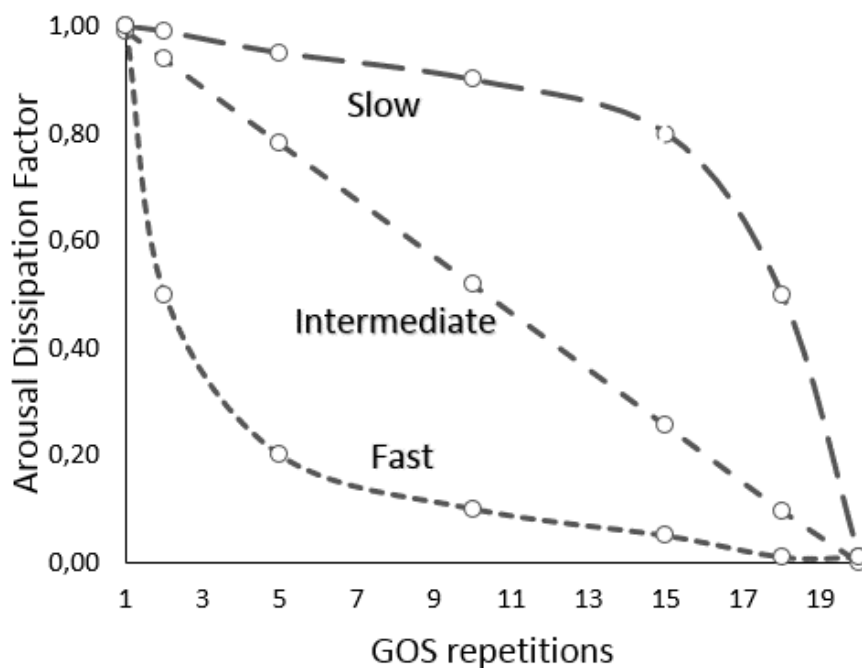
Figure 4: Arousal Dissipation Factor as a function of GOS repetitions. Arousal dissipation factor (ADF) controls the rate of dissipation. The functions are labeled according to their rate of dissipation: slow, intermediate and fast. A high ADF corresponds to a slow rate of dissipation. GOS repetitions equals the number of time steps that an individual has spent in the same global organismic state (Table 2). Instead of inputting a formula, the AHA model uses specified y- values for a predetermined set of x-values (shown here as empty circles), to represent the functions.

Part II: Exploring the effect of the Attention Modulation Factor

The attention modulation factor (AMF) controls the agent's attention restriction. The effect of the AMF was explored by simulating different combinations of the ADF and the AMF. The combinations are presented as *experimental conditions* in Table 1, and will be referred to throughout this section.

The ADF constants *0.20, 0.85* and *0.95* were chosen as representative values for *fast*, *intermediate* and *slow* (Table 1). The three candidates provide the base point for further testing of the "constant-option" of the ADF and makes comparison with the "function-option" easier.

The function presented in Figure 3 represents the standard AMF, here referred to as the *standard attention restriction*. In Part I, all simulations included this AMF. In Part II, the effect of different AMFs was explored by changing the model in two ways. First, I looked at the effect of *no attention restriction* ($m_a = 1$), where all sensory information is recorded equally independent of the current global organismic state (GOS). Second, I used a *linear attention restriction*, where the AMF was a negative linear function ($m_a = 1 - a_{GOS}$) (Fig.5). For the different combinations of ADF and AMF in Table 1, I recorded the probability of switching GOS and probability of re-election for each of the three ADF candidates: slow, intermediate and fast. Each simulation included 10 individuals, selected at random.
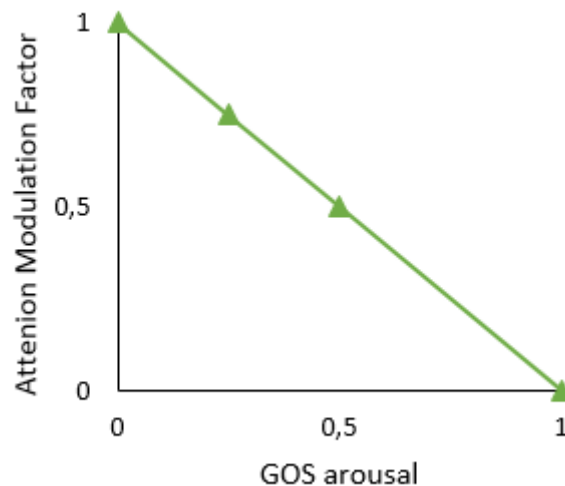
Figure 5: Linear attention restriction as a function of GOS arousal. Attention modulation factor controls the restriction of attention. GOS arousal is a quantitative measure of motivation (Table 2). A high GOS arousal corresponds to a strong motivation.

Table 1: Overview of all experimental conditions with different representation of the attention restriction factor (ADF) and attention modulation factor (AMF). Condition 1 and 2 explores the arousal dissipation factor for the standard AMF (Part I), while condition 3, 4, 5 and 6 explores other attention modulation factors (Part II see text for details). For every condition, I recorded the probability of GOS switching and probability of GOS re-electing. Additionally, I recorded the longest GOS chain in condition 1 (see text).

			AMF		
			Standard attention restriction	*No attention restriction*	*Linear attention restriction*
ADF	Constant	Slow	Condition 1	Condition 3	Condition 5
		Int.			
		Fast			
	Function	Slow	Condition 2	Condition 4	Condition 6
		Int.			
		Fast			

Table 2: List of central concepts

Name	Description
Agent	The individuals in the simulation are referred to as agents. The agents in the new AHA model are not based on any particular species of fish.
AHA! Model (The)	The name of the model. The abbreviation stands for Adaptive Heuristics and Architecture. The model is Individual Based.
Arousal Dissipation Factor (ADF)	A variable that controls the rate of GOS arousal dissipation. The ADF can be expressed as a constant or as a function of time. Both alternatives are explored here.
Attention Modulation Factor (AMF)	The numerical weight (within the range 0:1) given to the perception values to control their overall impact on the motivation and GOS. During times of high GOS arousal, attention is restricted to primarily situation-relevant stimuli. The latter are given the weights equal to 1.0 whereas irrelevant stimuli, i.e. those that contribute to different GOS',

are given smaller weights (e.g. 0.5). The restriction is controlled by the AMF.

Behavioural responses	The agents can choose from a repertoire of six different behaviours to operate in its environment: eating food, accelerate, decelerate, freeze, move up in depth and move down in depth.
Conspecifics	Other stochastic agents of the same species.
Dynamic Threshold, DT	A variable that is inversely dependent on GOS arousal. The dynamic threshold prevents the organism from erratic back-and-forth switching between goals. Only relatively major changes in the stimuli can bring about a GOS switch.
Global Organismic State (GOS)	The agent is always in one of four global organismic states that guides motivation and behaviour. The dominant GOS will dictate which behaviour is executed and which information is worth paying attention to.
GOS arousal	The quantitative expression of motivation in the model. Technically, the agents seek to reduce their arousal level and make decisions based on the perceived reduction of arousal that will follow as a consequence of that decision.
GOS repetition	If the dominant GOS is not switched or re-elected, this parameter increases by 1. The arousal dissipation function depends on GOS step repetition.
GOS Switch	A shift from one GOS to a different GOS. This is accompanied by an increase in GOS arousal and the resetting of the GOS repetition counter.

Neurobiological state (NBS)	These are survival circuits that fluctuates in arousal based on incoming stimuli. If the arousal level of a NBS becomes sufficiently high, it will reorganize the organisms global organismic state to address the survival function related to the NBS.
Re-election	An instance in which the GOS arousal level increases, and the GOS repetition counter is reset, but the GOS remains the same. As such, a re-election acts as a GOS switch back to itself.
Re-evaluation	The sum of GOS switches and re-evaluations. A re-evaluation is always accompanied by an elevation in GOS arousal.
Time step	In the simulation the lifetime of the organisms is divided into discrete time steps. All experiments this thesis spanned only 100 steps (corresponding to 1 h 12 min in the simulation)

2.3.Programs and language

The model runs on Fortran 2003 programming language. The code for the subroutine that I worked with is available in the appendix A1 (the complete code contains about 60 000 lines and hence is not included). The following software was used in various parts of this project:

- *Gfortran* – A free GNU compiler for the Fortran language.
- *Tortoise SVN 1.7.10.23359 (64 bit)* – This gives access to the subversion-based repository of the code. Code can be regularly updated whenever changes are made in the model. It is also possible to commit your own changes to the main model if you want to keep them.
- *ConEmu 161206.x64* – A terminal. The model is ran from the terminal using command line interface.
- *Far Manager 3 x64* – This plugin to ConEmu makes the interface easier to handle.
- *Code::Blocks version: svn build rev 10905* – This Integrated Development Environment (IDE) was used to make changes in the code in the experimental phase of the project.
- *Evince 2.32.0* – This is a graphic program used to view PostScript graphs and histograms that are generated by the model.
- *R for Windows 3.3.2* - Were used for statistical analysis.

3. Results

3.1. Part I: The role of the arousal dissipation factor

In *Part I,* the goal was to explore the effects of two different implementations of the arousal dissipation factor (ADF) on behaviour flexibility, recorded as the probability of switching global organismic state (GOS). As illustrated in Figure 6, the probability of switching GOS decreases with higher ADF constants. Each black dot or box in Figure 6 represents the average of 10 fish, each of which has their own randomly generated "genome". Since there is no environmental gradient in these simulations and no systematic differences in the fish' genome, the observed changes in GOS switching probability can be attributed to the agent's ADF value.
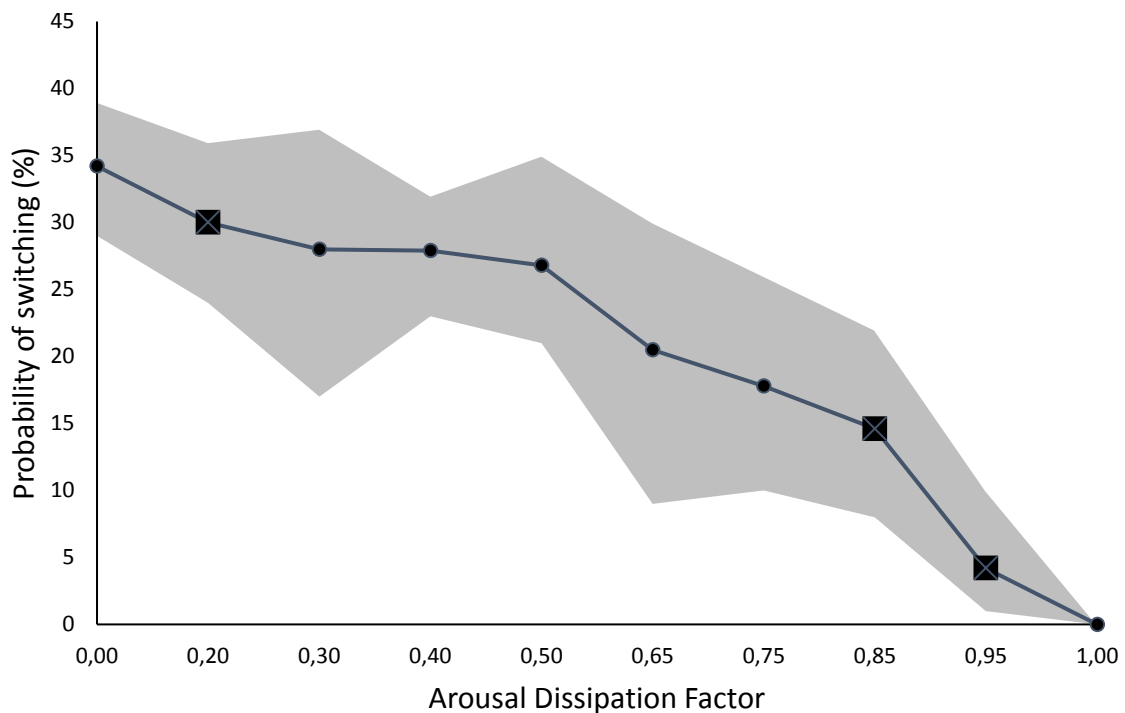
Figure 6: The effect of constant arousal dissipation factor (ADF) on the probability of switching global organismic state (GOS). The probability of switching GOS is a measure of behaviour flexibility. A high ADF corresponds to a slow rate of arousal dissipation. The black boxes highlight three ADF constants (0.20, 0.85 and 0.95) which represent a slow, intermediate and fast rate of dissipation respectively. The grey shading illustrates the total variation for the 10 simulations, while the black line represents the average probability of switching.

The probability of GOS switching also changes for the different ADF functional relationship (Fig.4) as seen in Table 3. Agents with a *slow* dissipation rate (Fig.4) were also the least likely to switch GOS during the simulations. The ADF function for *intermediate* dissipation rate seems to produce organisms that are somewhat more similar to the *slow* organisms, than the *fast* ones (Table 3).

Table 3: The effect of arousal dissipation factor (ADF) on probability of switching GOS. The rate of dissipation corresponds to the three functions in Figure 4. Probability of switching GOS is a measure of behaviour flexibility, with high switch-probability corresponds to a flexible behaviour.

Rate of dissipation (ADF function)	Probability of switching GOS (%)
Slow	3.3
Intermediate	8.8
Fast	30.0

The data from the *longest GOS chain* (Fig.7) show that a two-linear-component breakdown model provided a better fit than a single line model based on the Akaike Information Criterion[8]. This means that the data from the longest GOS chain support the notion of *two* types of behaviour: flexible and rigid. The *breakpoint* marks the best fit for a threshold between flexible and rigid behaviour. As seen in Figure 7, an arousal dissipation factor of 0.65 provides the lowest deviation of the residuals. As such, this data set (longest GOS chain) suggests that individuals with an ADF of 0.65 has intermediate behaviour flexibility, which is somewhat lower than the intermediate ADF value indicated in Figure 6 (ADF = 0.85).
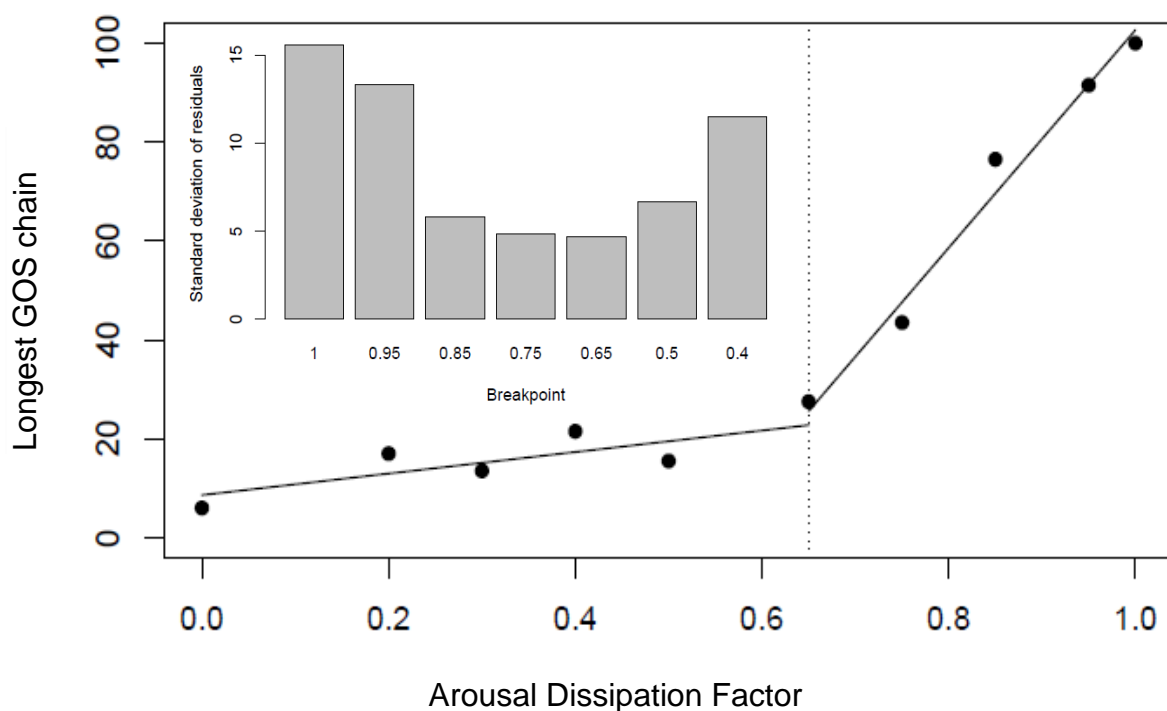
Figure 7: Breakpoint analysis of the longest GOS chain. The longest GOS chain is a measure of behaviour flexibility. A long GOS chain corresponds to a rigid behaviour (Table 2). The arousal dissipation factor controls the rate of dissipation. A high ADF corresponds to a slow arousal dissipation. An analysis of the standard deviation of residuals is superimposed in the top left corner (histogram). See appendix A4 for R-script.

---

[8] AIC = 64.2 and 87.9 for the best breakdown model and a single line model

The left regression curve (the interval between $0.0 - 0.65$) represents organisms with a flexible behaviour style. There is relatively little variation among these groups in the longest GOS chain as indicated by the almost horizontal regression curve, but we see a drop in the probability of switching GOS for higher ADF values (Fig.6). For further analysis, I selected one candidate (ADF = 0.20) from this interval. The right regression curve shows a much stronger relationship between flexibility and ADF values, and it therefore makes sense to look at two candidates (ADF = 0.85 and 0.95) from this interval.

So far, all simulations have included the *standard attention modulation curve* (Fig.3). By utilizing different constants *or* different functions for the arousal dissipation factor, we get organisms with a flexible or rigid behaviour style.

3.2. Part II: Exploring the effect of the attention modulation factor

In *Part II* of the simulation experiments, the effect of the attention modulation factor (AMF)[9] was explored. As shown in Table 3, I tested both the *constant* and *function* version of the ADF with different AMF values. Because the *slow, intermediate* and *fast* candidates from each version (constants and functions) gave similar results, I present only the results from the *constant* version here, but data for both versions are provided in appendix A5.

First, I consider the probability of GOS switching (Fig.8). The organisms with a *standard attention restriction* (red colour, Fig.8) is the same as the one in Figure 6, and represents a baseline which allows us to compare how *no attention restriction* (blue line, Fig. 8) and *linear attention restriction* (green line, Fig.8) affects the agents in the AHA model.

---

[9] For more explanation of the alternative attention modulation factors that where explored, see section 2.2
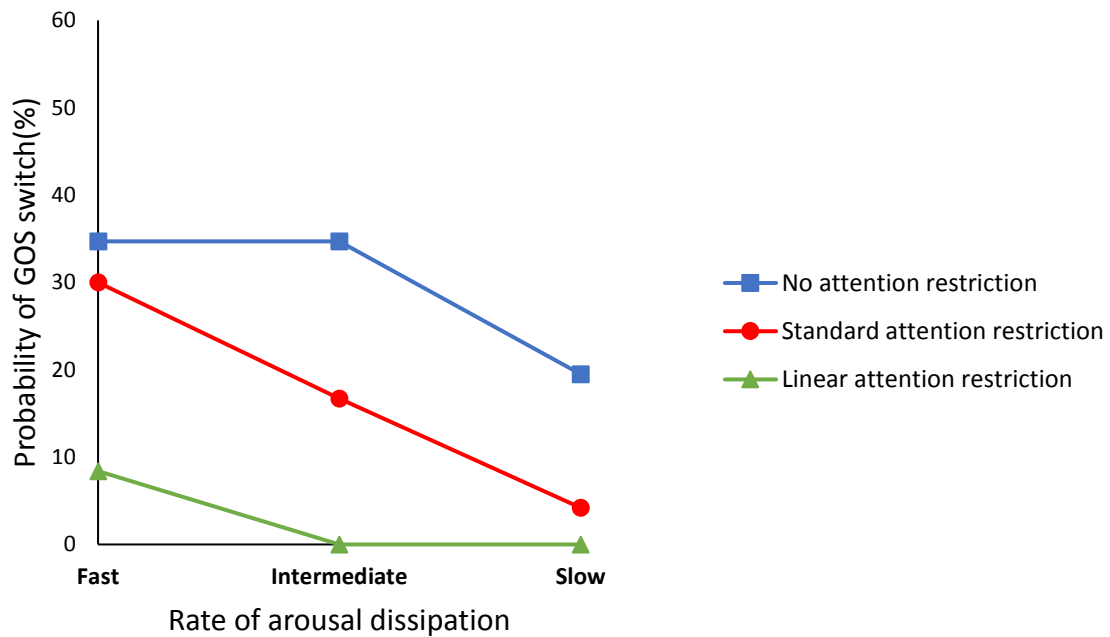
Figure 8: The effect of the attention modulation factor (AMF) on the probability of switching GOS. The probability of switching is a measure of behaviour flexibility. A high switch-probability corresponds to a flexible behaviour. The rate of arousal dissipation is controlled by the arousal dissipation factor (ADF). A high ADF corresponds to a slow rate of dissipation. As seen in the legend, the colour (blue, red and green), represents three different attention modulation factors (AMF). The AMF controls the attention restriction.

When we account for the organism's rate of arousal dissipation, organisms with *no attention restriction* where more likely to switch GOS than other organisms. For these organisms, *all* stimuli are evaluated independent of what is currently the GOS. This means that the neurobiological states (NBS) of an energy-deprived fish with no attention restriction (blue line, Fig.8) will not place any extra emphasis on food-related stimuli. Because they show no preference for GOS relevant stimuli, the NBS for hunger is no more likely to get "elected" than any other. The linear attention restriction curve (Fig.5) and the standard attention restriction curve (Fig.3), differ in how much they suppress signals at moderate (i.e. around 0.5) levels of GOS arousal. This distinction turns out to have important implications for the organism's propensity to switch GOS. The green curve (Fig.8) illustrates the effect of a linear attention restriction. Compared to organisms with a standard attention restriction (red line) they exhibit a reduced propensity to switch from their current global organismic state (Fig.8).

The attention modulation factor does not seem to play a role in the regulation of re-evaluations, as indicated by the overlap of AMFs (Fig.9). This can be explained by the fact that attention

restriction works by suppressing information that is *not* relevant to the current GOS, while leaving *relevant* information about the current GOS as salient as before [4]. The GOS arousal will hence dissipate at a rate corresponding to the ADF and lead to a re-evaluation as the dynamic threshold is reached. Organisms that lose *GOS arousal* at a faster rate (low ADF values), will reach the dynamic threshold sooner which leads to more frequent The agent's propensity to re-evaluate its global organismic state (GOS) does, however, seem to correlate with *the rate of arousal dissipation* (Fig.9). Organisms with a *faster* dissipation rates were more prone to re-evaluate their current global organismic state.
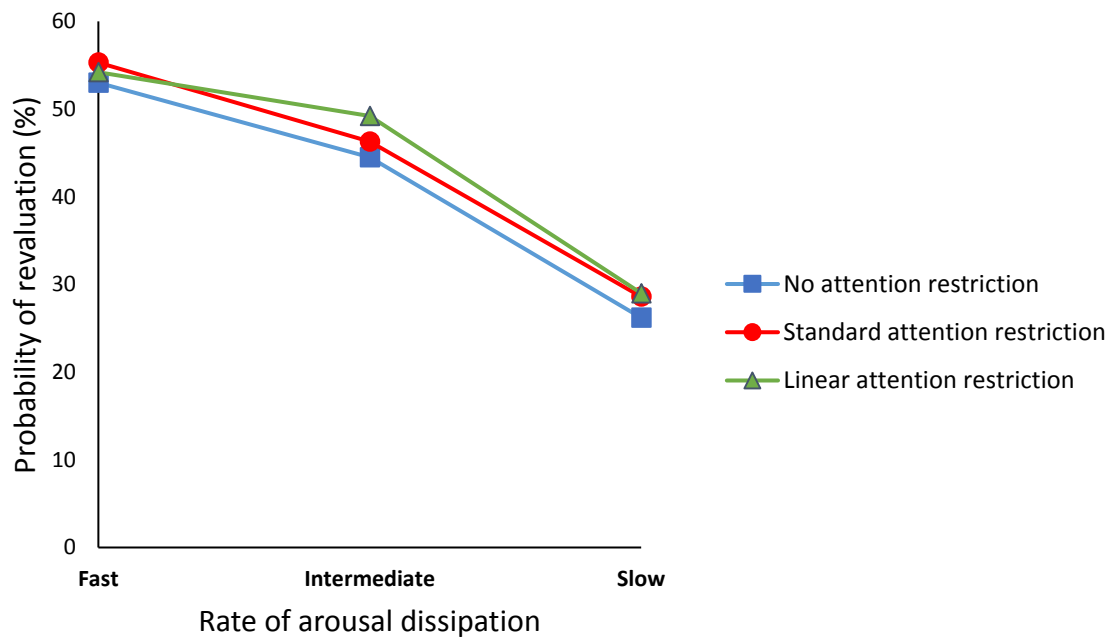
Figure 9: The probability of re-evaluating as a function of the rate of arousal dissipation. The probability of switching is a measure of behaviour flexibility. A high switch-probability corresponds to a flexible behaviour. The rate of arousal dissipation is controlled by the arousal dissipation factor (ADF). A high ADF corresponds to a slow rate of dissipation . As seen in the legend, the colour (blue, red and green), represents three different attention modulation factors (AMF). The AMF controls the attention restriction.

When the filtering of stimuli is quite strong at moderate levels of arousal (linear AMF; green line Fig. 8,9), while the neurobiological state (NBS) related to the global organismic state receives all relevant information, the re-evaluation will be based on a skewed perception of the environment. Since attention restriction remains strong at the time of this re-evaluation, information related to the current GOS is perceived much more salient, with the result that the agent remain in the same GOS after re-evaluation (Fig 8 & 9; green line)

Although not manipulated directly in this thesis, it is possible to draw some inferences about the role of the *dynamic threshold* (DT) for GOS switching (Fig.2).The most flexible individuals, with the highest probability of switching (Fig.9), re-evaluated the situation about half of the time, while deciding to switch GOS based on that re-evaluation only one-third of the time (Fig.8). As the agents cannot switch GOS more often than they re-evaluate, it looks like the maximum probability of switching were just above 55 %, in these simulations. This "upper limit" is likely caused by the size of the DT.

The ADF influences the *rate* at which the individuals make decisions about their GOS (Fig.9), independent of the AMF. Still, the AMF was shown to influence the number of GOS switches. This indicates that the AMF controls how many of the re-evaluations that lead to a GOS switch. By dividing the number of GOS switches by the number of re-elections, we get a "switch ratio", as illustrated in Figure 10. With a standard attention restriction, organisms with a slow rate of dissipation are less likely to switch GOS when re-evaluating, compared to organism with a fast rate of dissipation. Thus, the low probability of GOS switching seen by the slow organisms in Figure 8 is partly caused by fewer re-elections than the fast organisms (Fig.9), and partly caused by a relative difference in the outcome of a re-election (Fig.10)
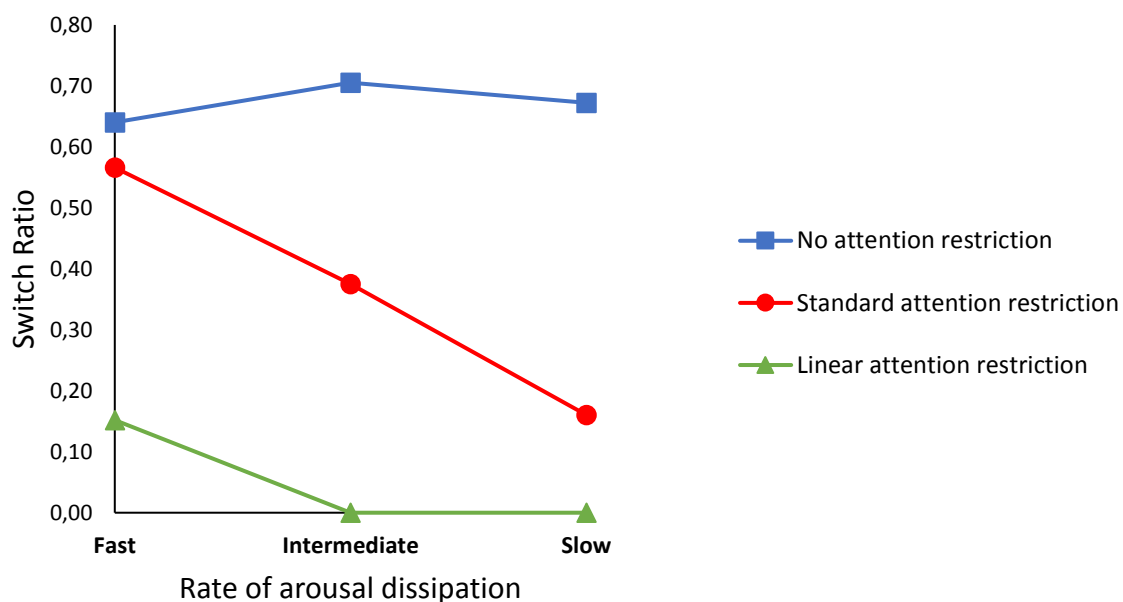
Figure 10: Switch ratios. The switch ratio is a measure of the relative behaviour flexibility when we account for the number of re-evaluations. A high switch ratio corresponds to many GOS switches per re-evaluation (in other words a high relative flexibility). The rate of arousal dissipation is controlled by the arousal dissipation factor (ADF). A high ADF corresponds to a slow rate of dissipation. As seen in the legend, the colour (blue, red and green), represents three different attention modulation factors (AMF). The AMF controls the attention restriction.

As expected, organisms with a linear attention restriction where even less likely to switch GOS (Fig.10). Organisms with *no attention restriction* seem to choose switching 7 out of every 10 re-elections. The probability of switching is the same for each re-evaluation and independent of the ADF.

4. Discussion

The arousal dissipation factor (ADF) and the attention modulation factor (AMF) both contribute to generating individual differences in behaviour flexibility in the organisms. The ADF controls the rate at which the arousal level decreases over time. This was found to affect how often the fish re-evaluated their current GOS. More frequent re-evaluations lead to a higher behaviour flexibility. The AMF controls attention restriction and was found to be important for generating rigid behaviour. Although the AMF did not influence the rate of re-evaluations, it did influence the outcome of those re-evaluations by making agents more likely to remain focused on their current goal.

The propensity to switch GOS as a measure of behavioural flexibility

As pointed out in Culum Brown et al. (2011), the descriptive label attached to a measure of personality (here: propensity to switch GOS) must correspond to a theoretical concept. In the thesis, behaviour flexibility was defined as the individual's ability to directly respond and adjust its behaviour to environmental stimuli (Coppens et al., 2010). Here, I argue that studying fluctuations in internal states (GOS) is a reasonable way of measuring behaviour flexibility.

LeDoux (2012) describes the organism's global organismic state as resulting from survival circuit activation. These survival functions, according to LeDoux, have been achieved through behaviour interaction with the environment. Indeed, LeDoux points out the highly conserved sensory-motor systems that makes out the foundation of survival circuits, while also adding that these systems do not exist in isolation. Instead, they have evolved to negotiate interactions with the environment for the purpose of sustaining life. Thus, by focusing on global organismic states, we *are* addressing the organism's response to its environment.

One could also argue that focusing on internal states is a better measure of behaviour flexibility, than observing external behaviours. Experiments with real fish do not grant us direct insight to the subjective motivational states. These can only be inferred from the observed behaviour (Culum Brown et al., 2011). This approach is a tedious uncertainty- and error-prone process. Consider for example immobility, which can be an expression of the freezing response associated with danger, or a hunting strategy. Thus, the very same behaviour response could be the result of different stimuli (LeDoux, 2012). It is also the case that specific survival circuits have different behavioural responses in different animals. For instance, while humans will

escape a threatening situation on two legs, a dog will do it on four legs. Thus, by focusing on the function of a circuit, rather than the observable behavioural response, we can set species-independent criteria for brain systems that are involved in detecting and responding to challenges and opportunities posed by the environment (LeDoux, 2012).

There are several studies that refer to internal processes as behaviour flexibility. For instance, Ruiz-Gomez, Huntingford, Overli, Thornqvist, and Hoglund (2011) measured behaviour flexibility as post-stress plasma cortisol level in rainbow trout. Here, confinement was used as a standardized stressor. In this way, they could select for high-responsive and low-responsive teleosts, showing that responsiveness to stress is heritable. In this thesis, the organisms did not respond exclusively to stressors, but all environmental cues (e.g. food, conspecifics and predators).

As the AHA model continues to develop, the validity of "propensity to switch GOS" as a measure behaviour flexibility could also be determined by converging observations. For instance, behavioural traits such as aggressiveness, routine formation and attack latency are known to correlate with behaviour flexibility (Koolhaas et al., 1999). These correlated behaviours characterises differences in *copings styles*. As stated earlier, behaviour flexibility may be an underlying factor for the consistency of coping styles (Coppens et al., 2010). As such, the presented results indicate that coping styles might emerge after several generations, if differences in behaviour flexibility proves to have more than one fitness optimum (see discussed below).

Finally, the simulations presented here involved a standardized environment that is stochastic, but stationary and homogenous. Any patterns in GOS switching should therefore be caused by properties of the agent. The most salient features of behaviour flexibility were reproduced based on a very simple mechanistic model: a combination of internal "drive" (GOS) competition, arousal and attention restriction. These mechanisms have direct counterparts in neurobiology.

Fitness related aspects of behavioural flexibility

In the finalized AHA model, the ADF values will be linked to the genome and hence be subjected to a selection process as the organisms live and reproduce. The current version of the AHA Model did not allow genetic adaptation of the agents. As such, inferences about the organism's *fitness* could not be made based on my results. Instead, they illustrate how the joint

effect of main mechanisms, competition between motivations and attention restriction, mediated by the overall arousal, can produce adaptive goal-directed behaviour. In the finished model, individual differences in behaviour flexibility will likely have fitness-related implications for the agents. Here, I give a brief outline of some of the fitness- related costs and benefits to behaviour flexibility based on prior research.

Organisms face a fundamental challenge in deciding between maintaining a current goal in the face of distraction, or to switch between goals in response to changes (Dreisbach & Goschke, 2004). Flexibility and rigidity are both potentially adaptive and the fitness consequences of each approach is often context dependent (Mittelbach, Ballew, & Kjelvik, 2014).

During times of urgent need, which corresponds to high level of arousal, it's reasonable to suspect that rigidity is more beneficial. For example, a starved fish runs the risk of dying from starvation, thus placing the fish in a highly motivated (aroused) state. The only way to reduce arousal is to fixate on foraging, even at the risk of detection from predators or reduced reproduction. When an animal is close to dying from starvation, paying attention and responding to various stimuli that is not directly associated with reducing hunger must be regarded as a distraction. Dreisbach and Goschke (2004) found that there is a positive relationship between flexibility and distractibility. However, Lavie (2005) argues that distractions can be prevented when the task-relevant stimuli carries a high perceptual load. The rationale for this seems to be that when the perception processing capacity is full, distracting stimuli is not registered by the sensory system.

Similarly, if an agent has encountered many predators recently and continues to see them, it would be adaptive to fix on *predator avoidance* and disregard all other stimuli that are not directly linked to reducing the fear-state. Switching to a foraging state (i.e. being distracted) at this point can be a fatal mistake (Lima & Dill, 1990).

However, such rigid commitment to a task comes with its own set of costs (Dreisbach & Goschke, 2004). Predation risk is not constant and so an adaptive response to fluctuations in risk requires that the animal sample information on the presence or absence of predators (Sih, 1992). For instance, Sih (1992) showed that one hour exposure of a predator was enough to keep juvenile notonectids (*Notonecta hoffmanni)* in refuge during the entire duration of the experiment (24 h) in some cases. The results also showed that well fed notonectids were more reluctant to come out of hiding, suggesting that their current state was involved in their decision.

This study demonstrates the importance of information sampling and risk assessment. According to Dall et al. (2005) such information use is key to adaptive behaviour of animals.

Risk assessment is also discussed in Lima and Dill (1990). Here, they argue that a "false positive" is better than a "false negative" when it comes to predator risk assessment. When in doubt, it is better to assume the presence of a predator and forgo feeding for a day, than to assume the opposite and get eaten. They also point out that vigilance is an important aspect of feeding efficiency because it detracts from energy intake. As pointed out earlier, the rate of re-evaluations in our model can be viewed as a form of vigilance measurement, because organisms who frequently re-evaluate their GOS will be more likely notice the approach of a predator, but also more likely to get distracted.

Wolf et al. (2008) also show that investing sampling time to respond more adaptively to environmental cues can produce additional payoffs, when we consider the costs (of time and energy) of such information sampling. Furthermore, their simple "responsiveness-model" illustrated that the benefits of responsiveness were frequency- dependent, which may explain why both flexible and rigid individuals are represented in many populations. Frequency-dependent selection has also been found in previous versions of the AHA model (Giske et al., 2013). Wolf et al. (2008) also showed that positive-feedback mechanisms could further reduce the cost of behaviour flexibility. One such feedback mechanism could reduce cost of sampling due to a training effect. If this is the case it might pay to be *consistent,* rather than alternate between a flexible and rigid approach to environmental cues. The relationship between flexibility and plasticity is discussed below.

Responsiveness to environmental change has also been studied in teleost fish. Ruiz-Gomez et al. (2008) found that highly responsive rainbow trout where quicker to find and consume relocated food, than unresponsive trout. An interesting find was that the unresponsive individuals swam over the (clearly visible) relocated food to get to the previously rewarding arm of the test-maze. The fact that the trout seemed to ignore visible food on its way to a previously rewarding location, highlights that the trout was motivated by food incentives, but failed to notice food when it appeared unexpectedly. This ties in well with the presented results, as they show that organisms almost completely disregard environmental cues, when attention is strongly restricted. Empirical findings as in Ruiz-Gomez et al. (2008) implies that too much attention restriction can have negative, fitness-related consequences for the animal because

important information is ignored, when they should be attended to. They also support the role of selective attention (AMF) as a mechanism that affects behaviour flexibility.

<u>Behaviour plasticity</u>

The scientific literature seems to suggest that flexibility and rigidity both have its merits (Ruiz-Gomez et al., 2011). This begs the question "why choose?" If flexible and rigid behaviour represents different optima in different environments, then animals should be plastic enough to exhibit the optimum in both environments (Sih et al., 2004). Again, it might not be that simple as there are important constraints to such behavioural plasticity.

One such constraint is that the underlying *components* of behaviour are limited in their flexibility (Duckworth, 2010). For example, even though the adult human brain can respond to experience by increased connectivity of the neurons, such growth and rewiring are slow on a scale of weeks to months (Duckworth, 2010). In fish, as well as many other animals, responsiveness or behaviour flexibility seems to correlate with hypothalamus-pituitary-adrenal (HPA-axis) reactivity (Ruiz-Gomez et al., 2011). This time lag for the physical components to change represents an absolute constrain to behavioural plasticity (Jacobs & Wingfield, 2000). Another consideration is that even if phenotypes *are* plastic, but with time lags, the unpredictability of the environment makes it likely that the organism will spend some time in the wrong phenotype (Sih et al., 2004).

However, we should note that personality traits are not incompatible with behaviour plasticity. While "stability" usually refers to the absolute measure of behaviour, "consistency" refers to the predictability of behaviour, when a measurement is repeated at a later time (Culum Brown et al., 2011). As such, animals can become more inclined to change its GOS (i.e. be more flexible) while its relative flexibility within the population stays the same. The adaptive value of certain behaviours can vary throughout the animal's lifetime (Adriaenssens & Johnsson, 2016) and so a slow and perhaps energy costly "reconfiguration" of neural circuits that control those behaviours might still be adaptive. This is implemented in the AHA model by the use of developmental modulation (Fig.1).

Additionally, there are examples of animals showing temporarily increased risk-taking behaviour during mating seasons (Lastein, Hoglund, Mayer, Overli, & Doving, 2008). The increased risk of predation is compensated for by increased reproductive success. In the terminology of the AHA model, this suggests that animals can temporarily favour a specific

GOS, leading to a "GOS-specific-rigidity". As of now, this is not implemented in the AHA model.

Finally, we must also consider the unpredictable nature of the animal's environment. When information about the world is uncertain and "noisy", it may be adaptive to ignore environmental cues and behave consistently (Culum Brown et al., 2011). In this case, "adaptive coin-flipping" could be a useful strategy (Kaplan & Cooper, 1984). The adaptive coin-flipping principle states that, under certain circumstances, there may be a selective advantage to allow for some level of random phenotypic expression (Kaplan & Cooper, 1988) . Furthermore, if the animal is in a state of high need, it does not pay much to collect additional information, even though such information would be valuable. Because uncertainty is an inherent fundamental state of the environment, animals can use a range of strategies, which would translate into a range of motivation competition threshold or attention restriction parameters.

<u>Further perspectives</u>

It's an interesting theoretical point that whereas previous models have focused on global *objective* characteristics of animals (e.g. body size, age, sex etc.) there is a clear need to focus on *internal* subjective states and *experiences* like motivation, global organismic state, subjective estimations of probability etc. which actually mediate various aspects of decision-making and can significantly affect fitness.

The potential of a complex "artificial life" model system such as the AHA model, is that it allows us insight into these *internal* subjective states. It also provides us with the opportunity to trace how selection affects specific cognitive mechanisms in a specific group of animals and how variability within a population is shaped. Furthermore, the natural selection "in silico" can be traced back to the genes, specific allele frequencies and mutations in real life. This kind of understanding is not possible with optimisation models, but with a finalized AHA model, it will be.

5. References

Adriaenssens, B., & Johnsson, J. I. (2016). Shy trout grow faster: do personality traits predict fitness of brown trout in the wild? (vol 22, pg 135, 2016). *Behavioral Ecology, 27*(5), E8-E8. doi:10.1093/beheco/arw111

Andersen, B. S., Jorgensen, C., Eliassen, S., & Giske, J. (2016). The proximate architecture for decision-making in fish. *Fish and Fisheries, 17*(3), 680-695. doi:10.1111/faf.12139

Bell, A. M. (2005). Behavioural differences between individuals and two populations of stickleback (Gasterosteus aculeatus). *Journal of Evolutionary Biology, 18*(2), 464-473. doi:10.1111/j.1420-9101.2004.00817.x

Benus, R. F., Dendaas, S., Koolhaas, J. M., & Vanoortmerssen, G. A. (1990). Routine Formation and Flexibility in Social and Nonsocial Behavior of Aggressive and Nonaggressive Male-Mice. *Behaviour, 112*, 176-193. doi:Doi 10.1163/156853990x00185

Bohus, B., Benus, R. F., Fokkema, D. S., Koolhaas, J. M., Nyakas, C., van Oortmerssen, G. A., . . . Steffens, A. B. (1987). Neuroendocrine states and behavioral and physiological stress responses. *Progress in Brain Research <D>, 72*, 57-70.

Brown, C., Burgess, F., & Braithwaite, V. A. (2007). Heritable and experiential effects on boldness in a tropical poeciliid. *Behavioral Ecology and Sociobiology, 62*(2), 237-243. doi:10.1007/s00265-007-0458-3

Brown, C., Krause, J., & Laland, K. N. (2011). *Fish cognition and behavior* (2nd ed.). Chichester, West Sussex, UK ; Ames, Iowa: Wiley-Blackwell.

Brown, J. S. (1999). Vigilance, patch use and habitat selection: Foraging under predation risk. *Evolutionary Ecology Research, 1*(1), 49-71.

Budaev, S. (2017). *BASE_UTILS.pdf*. Fortran Manual. Department of Biology. University of Bergen.

Castanheira, M. F., Conceicao, L. E. C., Millot, S., Rey, S., Begout, M. L., Damsgard, B., . . . Martins, C. I. M. (2017). Coping styles in farmed fish: consequences for aquaculture. *Reviews in Aquaculture, 9*(1), 23-41. doi:10.1111/raq.12100

Coppens, C. M., de Boer, S. F., & Koolhaas, J. M. (2010). Coping styles and behavioural flexibility: towards underlying mechanisms. *Philosophical Transactions of the Royal Society B-Biological Sciences, 365*(1560), 4021-4028. doi:10.1098/rstb.2010.0217

Dall, S. R. X. (2005). Information and its use by animals in evolutionary ecology. *Trends in Ecology & Evolution, 20*(4), 187-193. doi:10.1016/j.tree.2005.01.010

Dingemanse, N. J., Kazem, A. J. N., Reale, D., & Wright, J. (2010). Behavioural reaction norms: animal personality meets individual plasticity. *Trends in Ecology & Evolution, 25*(2), 81-89. doi:10.1016/j.tree.2009.07.013

Dreisbach, G., & Goschke, T. (2004). How positive affect modulates cognitive control: Reduced perseveration at the cost of increased distractibility. *Journal of Experimental Psychology-Learning Memory and Cognition, 30*(2), 343-353. doi:10.1037/0278-7393.30.2.343

Duckworth, R. A. (2010). Evolution of Personality: Developmental Constraints on Behavioral Flexibility. *Auk, 127*(4), 752-758. doi:10.1525/auk.2010.127.4.752

Eliassen, S., Andersen, B. S., Jorgensen, C., & Giske, J. (2016). From sensing to emergent adaptations: Modelling the proximate architecture for decision-making. *Ecological Modelling, 326*, 90-100. doi:10.1016/j.ecolmodel.2015.09.001

Fisher, R. A. (1930). *The genetical theory of natural selection*. Oxford,: The Clarendon press.

Giske, J., Eliassen, S., Fiksen, O., Jakobsen, P. J., Aksnes, D. L., Jorgensen, C., & Mangel, M. (2013). Effects of the Emotion System on Adaptive Behavior. *American Naturalist, 182*(6), 689-703. doi:10.1086/673533

Giske, J., Eliassen, S., Fiksen, O., Jakobsen, P. J., Aksnes, D. L., Mangel, M., & Jorgensen, C. (2014). The emotion system promotes diversity and evolvability. *Proceedings of the Royal Society B-Biological Sciences, 281*(1791). doi:ARTN 20141096

10.1098/rspb.2014.1096

Heller, R., & Milinski, M. (1979). Optimal Foraging of Sticklebacks on Swarming Prey. *Animal Behaviour, 27*(Nov), 1127-1141. doi:Doi 10.1016/0003-3472(79)90061-7

Huntingford, F. A., Andrew, G., Mackenzie, S., Morera, D., Coyle, S. M., Pilarczyk, M., & Kadri, S. (2010). Coping strategies in a strongly schooling fish, the common carp Cyprinus carpio. *Journal of Fish Biology, 76*(7), 1576-1591. doi:10.1111/j.1095-8649.2010.02582.x

Jacobs, J. D., & Wingfield, J. C. (2000). Endocrine control of life-cycle stages: A constraint on response to the environment? *Condor, 102*(1), 35-51. doi:Doi 10.1650/0010-5422(2000)102[0035:Ecolcs]2.0.Co;2

Kaplan, R. H., & Cooper, W. S. (1984). The Evolution of Developmental Plasticity in Reproductive Characteristics - an Application of the Adaptive Coin-Flipping Principle. *American Naturalist, 123*(3), 393-410. doi:Doi 10.1086/284211

Kaplan, R. H., & Cooper, W. S. (1988). On the Evolution of Coin-Flipping Plasticity - a Response to Mcginley, Temme, and Geber. *American Naturalist, 132*(5), 753-755. doi:Doi 10.1086/284887

Koolhaas, J. M., Korte, S. M., De Boer, S. F., Van Der Vegt, B. J., Van Reenen, C. G., Hopster, H., . . . Blokhuis, H. J. (1999). Coping styles in animals: current status in behavior and stress-physiology. *Neuroscience and Biobehavioral Reviews, 23*(7), 925-935. doi:Doi 10.1016/S0149-7634(99)00026-3

Lastein, S., Hoglund, E., Mayer, I., Overli, O., & Doving, K. B. (2008). Female Crucian Carp, Carassius carassius, Lose Predator Avoidance Behavior when Getting Ready to Mate. *Journal of Chemical Ecology, 34*(11), 1487-1491. doi:10.1007/s10886-008-9553-9

Lavie, N. (2005). Distracted and confused?: Selective attention under load. *Trends in Cognitive Sciences, 9*(2), 75-82. doi:10.1016/j.tics.2004.12.004

LeDoux, J. E. (2012). Rethinking the Emotional Brain. *Neuron, 73*(4), 653-676. doi:10.1016/j.neuron.2012.02.004

LeDoux, J. E. (2015). *Anxious : using the brain to understand and treat fear and anxiety*. New York, New York: Viking.

Lima, S. L., & Dill, L. M. (1990). Behavioral Decisions Made under the Risk of Predation - a Review and Prospectus. *Canadian Journal of Zoology-Revue Canadienne De Zoologie, 68*(4), 619-640. doi:DOI 10.1139/z90-092

Lotka, A. J. (1925). *Elements of physical biology*. Baltimore,: Williams & Wilkins company.

Mendl, M. (1999). Performing under pressure: stress and cognitive function. *Applied Animal Behaviour Science, 65*(3), 221-244. doi:Doi 10.1016/S0168-1591(99)00088-X

Mittelbach, G. G., Ballew, N. G., & Kjelvik, M. K. (2014). Fish behavioral types and their ecological consequences. *Canadian Journal of Fisheries and Aquatic Sciences, 71*(6), 927-944. doi:10.1139/cjfas-2013-0558

Moran, J., & Desimone, R. (1985). Selective Attention Gates Visual Processing in the Extrastriate Cortex. *Science, 229*(4715), 782-784. doi:DOI 10.1126/science.4023713

Overli, O., Winberg, S., & Pottinger, T. G. (2005). Behavioral and Neuroendocrine Correlates of Selection for Stress Responsiveness in Rainbow Trout--a Review. *Integr Comp Biol, 45*(3), 463-474. doi:10.1093/icb/45.3.463

Pagnoni, G., Zink, C. F., Montague, P. R., & Berns, G. S. (2002). Activity in human ventral striatum locked to errors of reward prediction. *Nature Neuroscience, 5*(2), 97-98. doi:10.1038/nn802

Pfaff, D. W. (2006). *Brain arousal and information theory : neural and genetic mechanisms*. Cambridge, Mass.: Harvard University Press.

Ruiz-Gomez, M. D., Huntingford, F. A., Overli, O., Thornqvist, P. O., & Hoglund, E. (2011). Response to environmental change in rainbow trout selected for divergent stress coping styles. *Physiology & Behavior, 102*(3-4), 317-322. doi:10.1016/j.physbeh.2010.11.023

Ruiz-Gomez, M. D., Kittilsen, S., Hoglund, E., Huntingford, F. A., Sorensen, C., Pottinger, T. G., . . . Overli, O. (2008). Behavioral plasticity in rainbow trout (Oncorhynchus mykiss) with divergent coping styles: When doves become hawks. *Hormones and Behavior, 54*(4), 534-538. doi:10.1016/j.yhbeh.2008.05.005

Schjolden, J., Stoskhus, A., & Winberg, S. (2005). Does individual variation in stress responses and agonistic behavior reflect divergent stress coping strategies in juvenile rainbow trout? *Physiol Biochem Zool, 78*(5), 715-723. doi:10.1086/432153

Seward, J. P. (1956). Drive, Incentive, and Reinforcement. *Psychological Review, 63*(3), 195-203. doi:DOI 10.1037/h0048229

Sih, A. (1992). Prey Uncertainty and the Balancing of Antipredator and Feeding Needs. *American Naturalist, 139*(5), 1052-1069. doi:Doi 10.1086/285372

Sih, A., Bell, A., & Johnson, J. C. (2004). Behavioral syndromes: an ecological and evolutionary overview. *Trends in Ecology & Evolution, 19*(7), 372-378. doi:10.1016/j.tree.2004.04.009

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cogn Psychol, 12*(1), 97-136.

Weblink. (2017a). The AHA! Project Development Pages.   Retrieved from http://158.37.63.57

Weblink. (2017b). HEDG2_04.F90/Alleles.   Retrieved from http://158.37.63.57/doxydoc/HEDG2__04_8f90_source.html#l00957

Weblink. (2017c). HEDG2_04.F90/genotype.   Retrieved from http://158.37.63.57/doxydoc/HEDG2__04_8f90_source.html#l01676

Weblink. (2017d). HEDG2_04.f.90/gamma2gene.   Retrieved from http://158.37.63.57/doxydoc/interfacecommondata_1_1gamma2gene.html

Weil, Z. M., Zhang, Q. Y., Hornung, A., Blizard, D., & Pfaff, D. W. (2010). Impact of generalized brain arousal on sexual behavior. *Proceedings of the National Academy of Sciences of the United States of America, 107*(5), 2265-2270. doi:10.1073/pnas.0914014107

Wilson, D. S., Clark, A. B., Coleman, K., & Dearstyne, T. (1994). Shyness and Boldness in Humans and Other Animals. *Trends in Ecology & Evolution, 9*(11), 442-446. doi:10.1016/0169-5347(94)90134-1

Wolf, M., van Doorn, G. S., & Weissing, F. J. (2008). Evolutionary emergence of responsive and unresponsive personalities. *Proceedings of the National Academy of Sciences of the United States of America, 105*(41), 15825-15830. doi:10.1073/pnas.0805473105

6. Appendices

A1. The subroutine that was used in the simulations

```fortran
subroutine LARS_TEST_LAB()

    integer :: ind, count_rand_walks, j, i
    integer :: food_item_selected

    real :: step_rwalk, cost_step

    integer, dimension(proto_parents%population_size) ::
random_sample_individuals

    !> Lars' variables are prefixed with lars_
    !> OUTPUT: Declaring record which has the data values appended for each
individual
    character(len=2000) :: lars_file_record_append_data_gos_label
    character(len=2000) :: lars_file_record_append_data_gos_arousal
    character(len=2000) :: lars_file_record_append_data_gos_repeated

    !! OUTPUT: Declaring file names as character string variables
    character(len=:), allocatable :: lars_output_filename_data_gos_label
    character(len=:), allocatable :: lars_output_filename_data_gos_arousal
    character(len=:), allocatable ::
lars_output_filename_data_gos_repeated

    !> OUTPUT: Declaring file units as integer numbers. We need file units
for
    !! behind the scene work, even though they are not directly used here.
    !! All the CSV routines can refer to the file by its name.
    integer lars_output_fileunit_data_gos_label
    integer lars_output_fileunit_data_gos_arousal
    integer lars_output_fileunit_data_gos_repeated

    !> This variable keeps a short description component for the csv output
    !! file names:
    character(len=*), parameter :: lars_ADF_File_descript = "pattern_1"

    !> Make an array of random integers that we will use for sampling random
    !! fish from the whole population
    random_sample_individuals =
PERMUTE_RANDOM(proto_parents%population_size)


!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    !> OUTPUT: Opening the output file for **gos label**.
    ! 1. we first set file name:
    lars_output_filename_data_gos_label = "0000_lars_gos_label_ADF_" //
&
                         lars_ADF_File_descript // csv
    ! 2. second, set internal file unit (we do not use the unit afterwards
but it is
    !!   used by fortran internally)
    lars_output_fileunit_data_gos_label = GET_FREE_FUNIT() ! get file unit
automatically
    ! 3. and physically open the output file for writing:
```

```fortran
    call CSV_OPEN_WRITE ( lars_output_filename_data_gos_label,
&
                          lars_output_fileunit_data_gos_label )
    ! 4. producing a whole record with column labels using our function
    !    'do_row_header': VAR_001, VAR_002.... VAR_100
    lars_file_record_append_data_gos_label = do_row_header(100)
    ! 5. write this first record that contains column labels
    call CSV_RECORD_WRITE( record=lars_file_record_append_data_gos_label,
&

csv_file_name=lars_output_filename_data_gos_label )

    !> OUTPUT: Opening the output file for **gos arousal**.
    lars_output_filename_data_gos_arousal = "0000_lars_gos_arousal_ADF_"
//    &
                          lars_ADF_File_descript // csv
    lars_output_fileunit_data_gos_arousal = GET_FREE_FUNIT() ! get file unit
automatically
    call CSV_OPEN_WRITE ( lars_output_filename_data_gos_arousal,
&
                          lars_output_fileunit_data_gos_arousal )
    !> producing a whole record with column labels
    lars_file_record_append_data_gos_arousal = do_row_header(100)
    call CSV_RECORD_WRITE(
record=lars_file_record_append_data_gos_arousal,   &

csv_file_name=lars_output_filename_data_gos_arousal )

    !> OUTPUT: Opening the output file for **gos repeated counter**.
    lars_output_filename_data_gos_repeated = "0000_lars_gos_repeated_ADF_"
// &
                          lars_ADF_File_descript // csv
    lars_output_fileunit_data_gos_repeated = GET_FREE_FUNIT() ! get file
unit automatically
    call CSV_OPEN_WRITE ( lars_output_filename_data_gos_repeated,
&
                          lars_output_fileunit_data_gos_repeated )
    !> producing a whole record with column labels
    lars_file_record_append_data_gos_repeated = do_row_header(100)
    call CSV_RECORD_WRITE(
record=lars_file_record_append_data_gos_repeated, &

csv_file_name=lars_output_filename_data_gos_repeated )

!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

    ! First loop through a random sample of 10 fish out from the whole
population
    INDS: do j=1, 10

        ! Choose the current individual ID number to work with from the
random sample.
        ind = random_sample_individuals(j)

        ! Exclude dead fish.
        if (proto_parents%individual(ind)%is_dead()) then
```

```fortran
          call LOG_MSG("WARNING: Found dead agent # " // TOSTR(ind) )
          exit INDS
        end if


!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        !> OUTPUT: Make the record an empty string when we start writing
data
        !! for each new individual
        lars_file_record_append_data_gos_label = ""
        lars_file_record_append_data_gos_arousal = ""
        lars_file_record_append_data_gos_repeated = ""

!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

        ! Start random walks of the fish
        WALKS: do i=1, 100

          call LOG_DELIMITER(LOG_LEVEL_CHAPTER)
          call LOG_DBG("Agent walk no=" // TOSTR(i) // " , agent ID " //
&
                     TOSTR(proto_parents%individual(ind)%get_id()) //
&
                     " (# " // TOSTR(ind) // "), name:"
&
                     // proto_parents%individual(ind)%individ_label()
//".")

          ! do random walk
          step_rwalk = dist2step(170.0)
          call LOG_DBG("   Step size for random walk: " // TOSTR(step_rwalk)
// &
                     ", " // TOSTR(step_rwalk /
proto_parents%individual(ind)%get_length()) // &
                     " agent's body sizes." )

          call proto_parents%individual(ind)%rwalk( step_rwalk,0.5, &
habitat_safe)

          call LOG_DBG("   cycle ind:walk "// TOSTR(ind) // ":"// TOSTR(i)
// &
TOSTR(proto_parents%individual(ind)%location(.TRUE.)))
          call LOG_DBG("             way "//
&
TOSTR(proto_parents%individual(ind)%way()))

          cost_step =
proto_parents%individual(ind)%cost_swim_burst(step_rwalk)
          call LOG_DBG("   Cost of random walk step: " // TOSTR(cost_step)
// &
                     " is " // TOSTR(100.0_SRP * cost_step /
proto_parents%individual(ind)%body_mass ) // &
                     "% of agent's body mass." )
```

```fortran
          !> Subtract the cost of swimming here:

proto_parents%individual(ind)%body_mass=proto_parents%individual(ind)%bod
y_mass - &
                                cost_step

          !==================================================
          ! Inner perceptions: stomach, bodymass, energy, age
          call proto_parents%individual(ind)%perceptions_inner()

          !==================================================
          ! Environmental perceptions: light, depth
          call proto_parents%individual(ind)%perceptions_environ()
          call LOG_DBG("Environmental perceptions: light " //
&

TOSTR(proto_parents%individual(ind)%perceive_light%get_current()) // &
             ", depth " //
&

TOSTR(proto_parents%individual(ind)%perceive_depth%get_current()) )

          !==================================================
          ! Spatial perceptions food, conspecifics, predators
          call
proto_parents%individual(ind)%see_food(habitat_safe%food,1)

          call
proto_parents%individual(ind)%see_consp(proto_parents%individual,&
                              proto_parents%individual%get_length(),
&
                              proto_parents%individual%is_alive() )

          call
proto_parents%individual(ind)%see_pred(habitat_safe%predators, &
                  habitat_safe%predators%get_size())

          !==================================================
          call
proto_parents%individual(ind)%motivations_percept_components()
          call proto_parents%individual(ind)%motivations_primary_calc()
          call proto_parents%individual(ind)%modulation()
          call proto_parents%individual(ind)%motivations_to_memory()
          call proto_parents%individual(ind)%gos_find()


!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

!------------------------------------------------------------
          ! OUTPUT: We are to place some code for producing outputs of
motivational
          ! variables below here.
          call CSV_RECORD_APPEND(
lars_file_record_append_data_gos_label,   &
```

```fortran
proto_parents%individual(ind)%gos_label() )

            call CSV_RECORD_APPEND(
lars_file_record_append_data_gos_arousal,    &

proto_parents%individual(ind)%arousal() )

            call CSV_RECORD_APPEND(
lars_file_record_append_data_gos_repeated,    &

proto_parents%individual(ind)%gos_repeated )

!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

            !> Check if the fish has died of starvation
            if (proto_parents%individual(ind)%starved_death()) then
              call proto_parents%individual(ind)%dies()
              call LOG_DELIMITER(LOG_LEVEL_SECTION)
              call LOG_DBG ("INFO: Agent dies due to starvation, ID: " //
&
TOSTR(proto_parents%individual(ind)%get_id()))
              call LOG_DBG ("      Body length: " //
&
                  TOSTR(proto_parents%individual(ind)%body_length) //
&
                  ", body mass: " //
&
                  TOSTR(proto_parents%individual(ind)%body_mass) //
&
                  ", maximum mass: " //
&
TOSTR(proto_parents%individual(ind)%body_mass_maximum) // &
                  ", birth mass : " //
&
                  TOSTR(proto_parents%individual(ind)%body_mass_birth)
)
              call LOG_DBG("      Energy :" //
&
                  TOSTR(proto_parents%individual(ind)%energy_current)
//    &
                  ", energy maximum: " //
&
                  TOSTR(proto_parents%individual(ind)%energy_maximum)
)
              call LOG_DELIMITER(LOG_LEVEL_SECTION)
              exit WALKS
            end if

            call LOG_DBG( "GOS is      :" //
proto_parents%individual(ind)%gos_label() )
            call LOG_DBG( "GOS arousal :" //
TOSTR(proto_parents%individual(ind)%arousal()) )
```

```fortran
            call LOG_DBG("**** can see food:   " //
TOSTR(proto_parents%individual(ind)%perceive_food%get_count()))

            !> Check if there is any food items in proximity (visibility
range)
            if ( proto_parents%individual(ind)%has_food() ) then
                call LOG_DBG("  distance     >" //
&

TOSTR(proto_parents%individual(ind)%perceive_food%foods_distances))
                call LOG_DBG("  dist. (d/l) >" //
&

TOSTR(proto_parents%individual(ind)%perceive_food%foods_distances &
                              /
proto_parents%individual(ind)%get_length()))

            !===============================================
            call LOG_DBG("   +++ Current mass: " //
TOSTR(proto_parents%individual(ind)%mass()) //  &
                       ", length: " //
TOSTR(proto_parents%individual(ind)%length()) //       &
                       ", energy: " //
TOSTR(proto_parents%individual(ind)%get_energy())  )
            !> Select the optimal food item out from its perception:
            food_item_selected =
proto_parents%individual(ind)%food_item_select(rescale_max_motivation=6.0
_SRP)

            !> Try to eat the optimal food item:
            call
proto_parents%individual(ind)%food_item_eat(food_item_selected,
habitat_safe%food)

            call LOG_DBG("**** Tried to eat food item: " //
TOSTR(food_item_selected))
            call LOG_DBG("   +++ Updated mass: " //
TOSTR(proto_parents%individual(ind)%mass()) //  &
                       ", length: " //
TOSTR(proto_parents%individual(ind)%length()) //       &
                       ", energy: " //
TOSTR(proto_parents%individual(ind)%get_energy())  )
            !stop "EATEN"
          else
            !> If no food objects were encountered we still grow with zero
food gain.
            call proto_parents%individual(ind)%mass_grow(0.0_SRP)
            call proto_parents%individual(ind)%len_grow(0.0_SRP)
          end if

          call LOG_DBG("**** can see consp: " //
TOSTR(proto_parents%individual(ind)%perceive_consp%get_count() ) )
          if ( proto_parents%individual(ind)%has_consp() ) then
              call LOG_DBG("  coord(1)     >" //
&
```

```fortran
TOSTR(proto_parents%individual(ind)%perceive_consp%conspecifics_seen(1)%l
ocation(.TRUE.)))
              call LOG_DBG("  iid          >" //
&

TOSTR(proto_parents%individual(ind)%perceive_consp%conspecifics_seen%get_
cid()))
          end if

          call LOG_DBG("**** can see pred:   " //
TOSTR(proto_parents%individual(ind)%perceive_predator%get_count() ) )
          if ( proto_parents%individual(ind)%has_pred() ) then
            call LOG_DBG("  coord(1)    =" //
&

TOSTR(proto_parents%individual(ind)%perceive_predator%predators_seen(1)%l
ocation(.TRUE.)))
              call LOG_DBG("  iid         =" //
&

TOSTR(proto_parents%individual(ind)%perceive_predator%predators_seen(1)%g
et_cid()))
              call LOG_DBG("  dist        =" //
&

TOSTR(proto_parents%individual(ind)%perceive_predator%predators_seen(1)%g
et_dist()))
          end if


        end do WALKS


!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        !> OUTPUT: Physically write the record to the disk
        call CSV_RECORD_WRITE(
record=lars_file_record_append_data_gos_label, &

csv_file_name=lars_output_filename_data_gos_label )

        call CSV_RECORD_WRITE(
record=lars_file_record_append_data_gos_arousal, &

csv_file_name=lars_output_filename_data_gos_arousal )

        call CSV_RECORD_WRITE(
record=lars_file_record_append_data_gos_repeated, &

csv_file_name=lars_output_filename_data_gos_repeated )

!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


        call LOG_DBG("INFO: Subtracting cost of living for agent # " //
&
                TOSTR(ind) // " and add weight and length to the
```

```fortran
history.")

            !> Subtract the cost of living
            call proto_parents%individual(ind)%subtract_living_cost()

            call
add_to_history(proto_parents%individual(ind)%body_length_history, &
                            proto_parents%individual(ind)%body_length)

            call
add_to_history(proto_parents%individual(ind)%body_mass_history, &
                            proto_parents%individual(ind)%body_mass)

            if (proto_parents%individual(ind)%starved_death()) then
                call proto_parents%individual(ind)%dies_debug()
                call LOG_DELIMITER(LOG_LEVEL_SECTION)
                call LOG_DBG ("INFO: Agent dies due to starvation, ID: " //
&

TOSTR(proto_parents%individual(ind)%get_id()))
                call LOG_DBG ("      Body length: " //
&
                        TOSTR(proto_parents%individual(ind)%body_length) //
&
                        ", body mass: " //
&
                        TOSTR(proto_parents%individual(ind)%body_mass) //
&
                        ", maximum mass: " //
&

TOSTR(proto_parents%individual(ind)%body_mass_maximum) // &
                        ", birth mass : " //
&
                        TOSTR(proto_parents%individual(ind)%body_mass_birth)
)
                call LOG_DBG("      Energy :" //
&
                        TOSTR(proto_parents%individual(ind)%energy_current)
//    &
                        ", energy maximum: " //
&
                        TOSTR(proto_parents%individual(ind)%energy_maximum)
)
                call LOG_DELIMITER(LOG_LEVEL_SECTION)
              end if

    end do INDS


!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    !> OUTPUT: Finally, we are closing the output files.
    call CSV_CLOSE( csv_file_name=lars_output_filename_data_gos_label )
    call CSV_CLOSE( csv_file_name=lars_output_filename_data_gos_arousal )
    call CSV_CLOSE( csv_file_name=lars_output_filename_data_gos_repeated )
```

```fortran
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

  contains


!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    ! OUTPUT: Produce a whole record with the names of the columns.
    function do_row_header(n_vars) result (string_record)
      integer, intent(in) :: n_vars
      character(len=2000) :: string_record

      !> Local vars
      integer :: i

      !> producing a whole record with column labels
      string_record = ""
      do i=1, n_vars
        call CSV_RECORD_APPEND( string_record, "VAR_" // TOSTR(i,n_vars) )
      end do

    end function do_row_header

  end subroutine LARS_TEST_LAB
```

A2. The Global Organismic State

```fortran
!> Find and set the global organismic state (GOS) based on the various
!! available motivation  values.
!! @note  GOS generation is a little changed in the new generation model.
!!        1. We try to avoid constant switching of the GOS by requiring that
!!           the difference between motivational components should exceed
!!           some threshold value, if it does not, retain old GOS. So minor
!!           fluctuations in the stimulus field are ignored. Threshold is
!!           a dynamic parameter, so can also be zero.
!!        2. The threshold is inversely related to the absolute value of
the
!!           motivations compared, when the motivations are low, the
!!           threshold is big, when their values are approaching 1, the
!!           threshold approaches zero. So motivations have relatively
little
!!           effects.
subroutine gos_find_global_state(this)
  class(GOS_GLOBAL), intent(inout) :: this

  !> Local variables
  !> Arousal is the maximum level of motivation among all available new
  !! incoming motivations ones. But we still have the older/previous
"current"
  !! arousal value `%gos_arousal` until it is updated from the newly
incoming
  !! perceptions and motivations.
  real(SRP) :: arousal_new

  !> Dynamic threshold of GOS, the threshold a motivation has to exceed to
  !! win the competition with the current motivation.
  real(SRP) :: gos_dthreshold

  !> PROCNAME is the procedure name for logging and debugging (with
MODNAME).
  character(len=*), parameter :: PROCNAME = "(gos_find_global_state)"

  !> Arousal is the maximum level among all available motivations (**final**
  !! motivational components). This is the **new** state depending on all
  !! the currently incoming perceptions.
  arousal_new = this%motivations%max_final()

  !> The GOS competition threshold is a function of the current arousal
  !! level, if it is very low, we need a relatively high competing motivation
  !! to win competition, if it is high (1) then very small difference is
  !! enough. But note that this is the relative differences. So if we have
  !! a low motivation 0.1, we need 0.155 to win (threshold=0.55,
  !! 0.155=0.1+0.1Ã—0.55 ), but if we have high motivation 0.8, almost any
  !! exceeding  motivation (>0.808) will win. So we limit the possible
  !! effects of low motivations. We get the actual value as a nonparametric
  !! function, currently by nonlinear interpolation of the grid values
  !! defined by the `MOTIVATION_COMPET_THRESHOLD_CURVE_` parameter arrays.
  !! @plot `aha_gos_arousal_winthreshold.svg`
  gos_dthreshold = DDPINTERPOL(
```

```fortran
                    MOTIVATION_COMPET_THRESHOLD_CURVE_ABSCISSA, &

                    MOTIVATION_COMPET_THRESHOLD_CURVE_ORDINATE, &
                                        this%gos_arousal )

    !> Save the interpolation plot in the debug mode using external command.
    !! @warning Involves **huge** number of plots, should normally be
    !!          disabled.
    call debug_interpolate_plot_save(                                        &
            grid_xx=MOTIVATION_COMPET_THRESHOLD_CURVE_ABSCISSA,              &
            grid_yy=MOTIVATION_COMPET_THRESHOLD_CURVE_ORDINATE,              &
            ipol_value=this%gos_arousal, algstr="DDPINTERPOL",               &
            output_file="plot_debug_arousal_gos_threshold_" //              &
                        TOSTR(Global_Time_Step_Model_Current) //            &
                        TAG_MMDD() // "_a_"// trim(this%individ_label()) //  &
                        "_" // RAND_STRING(LABEL_LENGTH,
LABEL_CST,LABEL_CEN) &
                        // PS )


    !> Now as we have the dynamic threshold, we can compare the current
    !! motivation level with the current (previous) arousal. If the motivation
    !! exceeds the current arousal by more than the threshold, the GOS
    !! changes to the new motivation. If not, we are still left with the
    !! previous GOS.
    AROUSAL_THRESHOLD: if (arousal_new – this%gos_arousal <                  &
                                    gos_dthreshold * this%gos_arousal)
then
      !> If the maximum current arousal does not exceed the threshold,
      !! we are left with the old GOS. However, we reduce the current arousal
      !! spontaneously using a simple linear or some non-linear dissipation
      !! pattern using the `%gos_repeated` parameter that sets the number of
      !! repeated occurrences of the same (current) GOS.
      !! First, increment GOS repeat counter.
      this%gos_repeated = this%gos_repeated + 1
      !> And spontaneously decrease, **dissipate**, the current arousal
level.
      !! Spontaneous dissipation of arousal is implemented by multiplying the
      !! current level by a factor within the range [0.0..1.0] that can depend
      !! on the number of times this GOS is repeated.
      !! @note Note that the dissipation function is local to this procedure.
      !!       `arousal_decrease_factor_fixed` = fixed value
      !!       `arousal_decrease_factor_nonpar` = nonlinear,
nonparametric,
      !!       based on nonlinear interpolation.
      !! @plot `aha_gos_arousal_dissipation.svg`
      this%gos_arousal = this%gos_arousal *                                  &
```

```fortran
arousal_decrease_factor_nonpar(this%gos_repeated)
    else AROUSAL_THRESHOLD
      !> If the maximum new arousal exceeds the threshold, we get to a
      !! **new GOS**. That is, the **highest** among the **new** competing
      !! motivations defines the new GOS.
      !! @note Use `associate`construct to set alias for long object
hierarchy.
      !! @note Note that `this%gos_repeated` is initialised to 1 at
`gos_reset`.
      associate ( MOT => this%motivations )
        !> Check **hunger**.
        GOS_IS_MAX: if (MOT%is_max_final(MOT%hunger)) then
          !> Reset all motivations to **non-dominant**.
          call this%gos_reset()
          !> Set new GOS for hunger...
          MOT%hunger%dominant_state = .TRUE.
          this%gos_main = MOT%hunger%label
          this%gos_arousal = MOT%hunger%motivation_finl
        !> Check **passive_avoidance**.
        else if (MOT%is_max_final(MOT%avoid_passive)) then GOS_IS_MAX
          !> Reset all motivations to **non-dominant**.
          call this%gos_reset()
          !> Set new GOS for passive_avoidance...
          MOT%avoid_passive%dominant_state = .TRUE.
          this%gos_main = MOT%avoid_passive%label
          this%gos_arousal = MOT%avoid_passive%motivation_finl
        !> Check **active_avoidance**.
        else if (MOT%is_max_final(MOT%avoid_active)) then GOS_IS_MAX
          !> Reset all motivations to **non-dominant**.
          call this%gos_reset()
          !> Set new GOS for active_avoidance...
          MOT%avoid_active%dominant_state = .TRUE.
          this%gos_main = MOT%avoid_active%label
          this%gos_arousal = MOT%avoid_active%motivation_finl
        !> Check **reproduction**.
        else if (MOT%is_max_final(MOT%reproduction)) then GOS_IS_MAX
          !> Reset all motivations to **non-dominant**.
          call this%gos_reset()
          !> Set new GOS for reproduction...
          MOT%reproduction%dominant_state = .TRUE.
          this%gos_main = MOT%reproduction%label
          this%gos_arousal = MOT%reproduction%motivation_finl
        end if GOS_IS_MAX
      end associate

    end if AROUSAL_THRESHOLD

    !> Add the current GOS parameters to the emotional memory stack
    !! @note Note that the memory stack arrays are defined in
    !!       APPRAISAL and cleaned/init in `init_appraisal`
    !! @note We can use the dedicated procedures. Here disabled so far to avoid
    !!       speed overhead.
    !call this%memory_motivations%gos_to_memory(
&
    !                 v_gos_label=this%gos_main,
```

```fortran
    &
    !                    v_gos_arousal= this%gos_arousal,           &
    !                    v_gos_repeated=this%gos_repeated  )
    call add_to_history(this%memory_motivations%gos_main, this%gos_main)
    call add_to_history(this%memory_motivations%gos_arousal,
this%gos_arousal)
    call add_to_history(this%memory_motivations%gos_repeated,
this%gos_repeated)


    !> Finally recalculate the attention weights for all the states'
perception
    !! components. The dominant GOS state will now get its default attention
    !! weights whereas all non-dominant states will get modulated values, i.e.
    !! values recalculated from a non-linear interpolation based **attention
    !! modulation curve**.
    call this%attention_modulate()

    !! @note   Note that type-bound functions can be used (although this makes
    !!         sense only outside of this module to avoid a small function-call
    !!         overhead): `if ( this%motivations%hunger%is_dominant() )
then`. For the
    !!         motivational state label we can use the accessor function
    !!         `%label_is` : `return_gos =
this%motivations%hunger%label_is()` (it is
    !!         **mandatory** outside of this module as label is declared
    !!         `private`).
    if (this%motivations%hunger%dominant_state) then
      return_gos = this%motivations%hunger%label
    else if (this%motivations%avoid_passive%dominant_state) then
      return_gos = this%motivations%avoid_passive%label
    else if (this%motivations%avoid_active%dominant_state) then
      return_gos = this%motivations%avoid_active%label
    else if (this%motivations%reproduction%dominant_state) then
      return_gos = this%motivations%reproduction%label
    end if

  end function gos_global_get_label


  !-----------------------------------------------------------------------
-----
  !> Calculate the overall level of arousal. Arousal is the current level
  !! of the dominant motivation that has brought about the current GOS at the
  !! previous time step.
  elemental function gos_get_arousal_level(this) result (arousal_out)
    class(GOS_GLOBAL), intent(in) :: this

    !> Arousal is the current level of motivation that has brought about GOS.
    real(SRP) :: arousal_out

    !> It is saved in this GOS-object component.
    arousal_out = this%gos_arousal

  end function gos_get_arousal_level
```

A3. The Attention Modulation Factor

```fortran
!> Modulate the attention weights to suppress all perceptions alternative
!! to the current GOS. This is done using the attention modulation
!! interpolation curve.
!! @warning This subroutine is called from within `gos_find` and should not
!!          be called separately.
subroutine gos_attention_modulate_weights(this)
  class(GOS_GLOBAL), intent(inout) :: this

  !> Local variable, the weight given to the attention weight components
  !! of all the non-dominant motivation states. Based on nonlinear
  !! interpolation.
  real(SRP) :: percept_w

  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
  !> **First**, we calculate the attention weight given to all non-dominant
  !! perceptions via nonlinear interpolation.
  percept_w = DDPINTERPOL( ATTENTION_MODULATION_CURVE_ABSCISSA,
&
                           ATTENTION_MODULATION_CURVE_ORDINATE,
&
                           this%gos_arousal )

  !> Save the interpolation plot in the debug mode using external command.
  !! @warning Involves **huge** number of plots, should normally be
  !!          disabled.
  call debug_interpolate_plot_save(
&
          grid_xx=ATTENTION_MODULATION_CURVE_ABSCISSA,
&
          grid_yy=ATTENTION_MODULATION_CURVE_ORDINATE,
&
          ipol_value=this%gos_arousal, algstr="DDPINTERPOL",
&
          output_file="plot_debug_attention_modulation_" //
&
                      TOSTR(Global_Time_Step_Model_Current) //
&
                      TAG_MMDD() // "_a_"// trim(this%individ_label()) //
&
                      "_" // RAND_STRING(LABEL_LENGTH,
LABEL_CST,LABEL_CEN) &
                      // PS )

  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
  !> **Second**, we reset the attention weights for the **dominant GOS
  !! state** to their **default** parameter values whereas for all other
  !! states, to the **recalculated** `percept_w` modulated
  !! value.
  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```fortran
    - - -
        !> The **dominant** state is **hunger**:
        RESET_DOMINANT: if ( this%motivations%hunger%is_dominant() ) then

            !> @note Dominant is **hunger**.
            call this%motivations%hunger%attention_weight%attention_init    &
                    (weight_light    = ATTENTION_WEIGHT_HUNGER_LIGHT,        &
                     weight_depth    = ATTENTION_WEIGHT_HUNGER_DEPTH,        &
                     weight_food_dir = ATTENTION_WEIGHT_HUNGER_FOOD_DIR,     &
                     weight_food_mem = ATTENTION_WEIGHT_HUNGER_FOOD_MEM,     &
                     weight_conspec  = ATTENTION_WEIGHT_HUNGER_CONSPEC,      &
                     weight_predator = ATTENTION_WEIGHT_HUNGER_PREDATOR,     &
                     weight_stomach  = ATTENTION_WEIGHT_HUNGER_STOMACH,      &
                     weight_bodymass = ATTENTION_WEIGHT_HUNGER_BODYMASS,     &
                     weight_energy   = ATTENTION_WEIGHT_HUNGER_ENERGY,       &
                     weight_age      = ATTENTION_WEIGHT_HUNGER_AGE,          &
                     weight_reprfac  = ATTENTION_WEIGHT_HUNGER_REPRFAC )

            call this%motivations%avoid_passive%attention_weight%attention_init  &
                    (weight_light    = ATTENTION_WEIGHT_AVOID_PASS_LIGHT *
percept_w,    &
                     weight_depth    = ATTENTION_WEIGHT_AVOID_PASS_DEPTH *
percept_w,    &
                     weight_food_dir = ATTENTION_WEIGHT_AVOID_PASS_FOOD_DIR *
percept_w,&
                     weight_food_mem = ATTENTION_WEIGHT_AVOID_PASS_FOOD_MEM *
percept_w,&
                     weight_conspec  = ATTENTION_WEIGHT_AVOID_PASS_CONSPEC *
percept_w, &
                     weight_predator = ATTENTION_WEIGHT_AVOID_PASS_PREDATOR *
percept_w,&
                     weight_stomach  = ATTENTION_WEIGHT_AVOID_PASS_STOMACH *
percept_w, &
                     weight_bodymass = ATTENTION_WEIGHT_AVOID_PASS_BODYMASS *
percept_w,&
                     weight_energy   = ATTENTION_WEIGHT_AVOID_PASS_ENERGY *
percept_w,  &
                     weight_age      = ATTENTION_WEIGHT_AVOID_PASS_AGE * percept_w,
&
                     weight_reprfac  = ATTENTION_WEIGHT_AVOID_PASS_REPRFAC *
percept_w )

            call this%motivations%avoid_active%attention_weight%attention_init    &
```

57

```fortran
           (weight_light    = ATTENTION_WEIGHT_AVOID_ACT_LIGHT * percept_w,
&
            weight_depth    = ATTENTION_WEIGHT_AVOID_ACT_DEPTH * percept_w,
&
            weight_food_dir = ATTENTION_WEIGHT_AVOID_ACT_FOOD_DIR *
percept_w, &
            weight_food_mem = ATTENTION_WEIGHT_AVOID_ACT_FOOD_MEM *
percept_w, &
            weight_conspec  = ATTENTION_WEIGHT_AVOID_ACT_CONSPEC *
percept_w,  &
            weight_predator = ATTENTION_WEIGHT_AVOID_ACT_PREDATOR *
percept_w, &
            weight_stomach  = ATTENTION_WEIGHT_AVOID_ACT_STOMACH *
percept_w,  &
            weight_bodymass = ATTENTION_WEIGHT_AVOID_ACT_BODYMASS *
percept_w, &
            weight_energy   = ATTENTION_WEIGHT_AVOID_ACT_ENERGY *
percept_w,   &
            weight_age      = ATTENTION_WEIGHT_AVOID_ACT_AGE * percept_w,
&
            weight_reprfac  = ATTENTION_WEIGHT_AVOID_ACT_REPRFAC *
percept_w )

      call this%motivations%reproduction%attention_weight%attention_init
&
           (weight_light    = ATTENTION_WEIGHT_REPRODUCE_LIGHT * percept_w,
&
            weight_depth    = ATTENTION_WEIGHT_REPRODUCE_DEPTH * percept_w,
&
            weight_food_dir = ATTENTION_WEIGHT_REPRODUCE_FOOD_DIR *
percept_w, &
            weight_food_mem = ATTENTION_WEIGHT_REPRODUCE_FOOD_MEM *
percept_w, &
            weight_conspec  = ATTENTION_WEIGHT_REPRODUCE_CONSPEC *
percept_w,  &
            weight_predator = ATTENTION_WEIGHT_REPRODUCE_PREDATOR *
percept_w, &
            weight_stomach  = ATTENTION_WEIGHT_REPRODUCE_STOMACH *
percept_w,  &
            weight_bodymass = ATTENTION_WEIGHT_REPRODUCE_BODYMASS *
percept_w, &
            weight_energy   = ATTENTION_WEIGHT_REPRODUCE_ENERGY *
percept_w,   &
            weight_age      = ATTENTION_WEIGHT_REPRODUCE_AGE * percept_w,
&
            weight_reprfac  = ATTENTION_WEIGHT_REPRODUCE_REPRFAC *
percept_w )

    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
    !> The **dominant** state is **avoid_passive**:
    else if ( this%motivations%avoid_passive%is_dominant() ) then
RESET_DOMINANT

      call this%motivations%hunger%attention_weight%attention_init
&
```

58

```fortran
            (weight_light    = ATTENTION_WEIGHT_HUNGER_LIGHT * percept_w,
&
             weight_depth    = ATTENTION_WEIGHT_HUNGER_DEPTH * percept_w,
&
             weight_food_dir = ATTENTION_WEIGHT_HUNGER_FOOD_DIR * percept_w,
&
             weight_food_mem = ATTENTION_WEIGHT_HUNGER_FOOD_MEM * percept_w,
&
             weight_conspec  = ATTENTION_WEIGHT_HUNGER_CONSPEC * percept_w,
&
             weight_predator = ATTENTION_WEIGHT_HUNGER_PREDATOR * percept_w,
&
             weight_stomach  = ATTENTION_WEIGHT_HUNGER_STOMACH * percept_w,
&
             weight_bodymass = ATTENTION_WEIGHT_HUNGER_BODYMASS * percept_w,
&
             weight_energy   = ATTENTION_WEIGHT_HUNGER_ENERGY * percept_w,
&
             weight_age      = ATTENTION_WEIGHT_HUNGER_AGE * percept_w,
&
             weight_reprfac  = ATTENTION_WEIGHT_HUNGER_REPRFAC * percept_w )
        !> @note Dominant **avoid_passive**.
        call this%motivations%avoid_passive%attention_weight%attention_init
&
             (weight_light    = ATTENTION_WEIGHT_AVOID_PASS_LIGHT,
&
             weight_depth     = ATTENTION_WEIGHT_AVOID_PASS_DEPTH,
&
             weight_food_dir = ATTENTION_WEIGHT_AVOID_PASS_FOOD_DIR,
&
             weight_food_mem = ATTENTION_WEIGHT_AVOID_PASS_FOOD_MEM,
&
             weight_conspec  = ATTENTION_WEIGHT_AVOID_PASS_CONSPEC,
&
             weight_predator = ATTENTION_WEIGHT_AVOID_PASS_PREDATOR,
&
             weight_stomach  = ATTENTION_WEIGHT_AVOID_PASS_STOMACH,
&
             weight_bodymass = ATTENTION_WEIGHT_AVOID_PASS_BODYMASS,
&
             weight_energy   = ATTENTION_WEIGHT_AVOID_PASS_ENERGY,
&
             weight_age      = ATTENTION_WEIGHT_AVOID_PASS_AGE,
&
             weight_reprfac  = ATTENTION_WEIGHT_AVOID_PASS_REPRFAC )

        call this%motivations%avoid_active%attention_weight%attention_init
&
             (weight_light    = ATTENTION_WEIGHT_AVOID_ACT_LIGHT * percept_w,
&
             weight_depth     = ATTENTION_WEIGHT_AVOID_ACT_DEPTH * percept_w,
&
             weight_food_dir = ATTENTION_WEIGHT_AVOID_ACT_FOOD_DIR *
percept_w, &
             weight_food_mem = ATTENTION_WEIGHT_AVOID_ACT_FOOD_MEM *
```

```fortran
percept_w, &
          weight_conspec  = ATTENTION_WEIGHT_AVOID_ACT_CONSPEC *
percept_w,   &
          weight_predator = ATTENTION_WEIGHT_AVOID_ACT_PREDATOR *
percept_w, &
          weight_stomach  = ATTENTION_WEIGHT_AVOID_ACT_STOMACH *
percept_w,   &
          weight_bodymass = ATTENTION_WEIGHT_AVOID_ACT_BODYMASS *
percept_w, &
          weight_energy   = ATTENTION_WEIGHT_AVOID_ACT_ENERGY *
percept_w,     &
          weight_age      = ATTENTION_WEIGHT_AVOID_ACT_AGE * percept_w,
&
          weight_reprfac  = ATTENTION_WEIGHT_AVOID_ACT_REPRFAC *
percept_w )

      call this%motivations%reproduction%attention_weight%attention_init
&
          (weight_light    = ATTENTION_WEIGHT_REPRODUCE_LIGHT * percept_w,
&
          weight_depth    = ATTENTION_WEIGHT_REPRODUCE_DEPTH * percept_w,
&
          weight_food_dir = ATTENTION_WEIGHT_REPRODUCE_FOOD_DIR *
percept_w, &
          weight_food_mem = ATTENTION_WEIGHT_REPRODUCE_FOOD_MEM *
percept_w, &
          weight_conspec  = ATTENTION_WEIGHT_REPRODUCE_CONSPEC *
percept_w,   &
          weight_predator = ATTENTION_WEIGHT_REPRODUCE_PREDATOR *
percept_w, &
          weight_stomach  = ATTENTION_WEIGHT_REPRODUCE_STOMACH *
percept_w,   &
          weight_bodymass = ATTENTION_WEIGHT_REPRODUCE_BODYMASS *
percept_w, &
          weight_energy   = ATTENTION_WEIGHT_REPRODUCE_ENERGY *
percept_w,     &
          weight_age      = ATTENTION_WEIGHT_REPRODUCE_AGE * percept_w,
&
          weight_reprfac  = ATTENTION_WEIGHT_REPRODUCE_REPRFAC *
percept_w )

    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
    !> The **dominant** state is **avoid_active**:
    else if ( this%motivations%avoid_active%is_dominant() ) then
RESET_DOMINANT

      call this%motivations%hunger%attention_weight%attention_init
&
          (weight_light    = ATTENTION_WEIGHT_HUNGER_LIGHT * percept_w,
&
          weight_depth    = ATTENTION_WEIGHT_HUNGER_DEPTH * percept_w,
&
          weight_food_dir = ATTENTION_WEIGHT_HUNGER_FOOD_DIR * percept_w,
&
          weight_food_mem = ATTENTION_WEIGHT_HUNGER_FOOD_MEM * percept_w,
```

```fortran
       &
              weight_conspec  = ATTENTION_WEIGHT_HUNGER_CONSPEC * percept_w,
       &
              weight_predator = ATTENTION_WEIGHT_HUNGER_PREDATOR * percept_w,
       &
              weight_stomach  = ATTENTION_WEIGHT_HUNGER_STOMACH * percept_w,
       &
              weight_bodymass = ATTENTION_WEIGHT_HUNGER_BODYMASS * percept_w,
       &
              weight_energy   = ATTENTION_WEIGHT_HUNGER_ENERGY * percept_w,
       &
              weight_age      = ATTENTION_WEIGHT_HUNGER_AGE * percept_w,
       &
              weight_reprfac  = ATTENTION_WEIGHT_HUNGER_REPRFAC * percept_w
       )

       call this%motivations%avoid_passive%attention_weight%attention_init
       &
              (weight_light    = ATTENTION_WEIGHT_AVOID_PASS_LIGHT *
       percept_w,    &
              weight_depth    = ATTENTION_WEIGHT_AVOID_PASS_DEPTH *
       percept_w,    &
              weight_food_dir = ATTENTION_WEIGHT_AVOID_PASS_FOOD_DIR *
       percept_w,&
              weight_food_mem = ATTENTION_WEIGHT_AVOID_PASS_FOOD_MEM *
       percept_w,&
              weight_conspec  = ATTENTION_WEIGHT_AVOID_PASS_CONSPEC *
       percept_w,  &
              weight_predator = ATTENTION_WEIGHT_AVOID_PASS_PREDATOR *
       percept_w,&
              weight_stomach  = ATTENTION_WEIGHT_AVOID_PASS_STOMACH *
       percept_w,  &
              weight_bodymass = ATTENTION_WEIGHT_AVOID_PASS_BODYMASS *
       percept_w,&
              weight_energy   = ATTENTION_WEIGHT_AVOID_PASS_ENERGY *
       percept_w,   &
              weight_age      = ATTENTION_WEIGHT_AVOID_PASS_AGE * percept_w,
       &
              weight_reprfac  = ATTENTION_WEIGHT_AVOID_PASS_REPRFAC *
       percept_w )

       !> @note Dominant is **avoid_active**.
       call this%motivations%avoid_active%attention_weight%attention_init
       &
              (weight_light    = ATTENTION_WEIGHT_AVOID_ACT_LIGHT,
       &
              weight_depth    = ATTENTION_WEIGHT_AVOID_ACT_DEPTH,
       &
              weight_food_dir = ATTENTION_WEIGHT_AVOID_ACT_FOOD_DIR,
       &
              weight_food_mem = ATTENTION_WEIGHT_AVOID_ACT_FOOD_MEM,
       &
              weight_conspec  = ATTENTION_WEIGHT_AVOID_ACT_CONSPEC,
       &
              weight_predator = ATTENTION_WEIGHT_AVOID_ACT_PREDATOR,
       &
```

```fortran
              weight_stomach  = ATTENTION_WEIGHT_AVOID_ACT_STOMACH,
&
              weight_bodymass = ATTENTION_WEIGHT_AVOID_ACT_BODYMASS,
&
              weight_energy   = ATTENTION_WEIGHT_AVOID_ACT_ENERGY,
&
              weight_age      = ATTENTION_WEIGHT_AVOID_ACT_AGE,
&
              weight_reprfac  = ATTENTION_WEIGHT_AVOID_ACT_REPRFAC )

        call this%motivations%reproduction%attention_weight%attention_init
&
              (weight_light    = ATTENTION_WEIGHT_REPRODUCE_LIGHT * percept_w,
&
              weight_depth     = ATTENTION_WEIGHT_REPRODUCE_DEPTH * percept_w,
&
              weight_food_dir = ATTENTION_WEIGHT_REPRODUCE_FOOD_DIR *
percept_w, &
              weight_food_mem = ATTENTION_WEIGHT_REPRODUCE_FOOD_MEM *
percept_w, &
              weight_conspec  = ATTENTION_WEIGHT_REPRODUCE_CONSPEC *
percept_w,  &
              weight_predator = ATTENTION_WEIGHT_REPRODUCE_PREDATOR *
percept_w, &
              weight_stomach  = ATTENTION_WEIGHT_REPRODUCE_STOMACH *
percept_w,  &
              weight_bodymass = ATTENTION_WEIGHT_REPRODUCE_BODYMASS *
percept_w, &
              weight_energy   = ATTENTION_WEIGHT_REPRODUCE_ENERGY *
percept_w,   &
              weight_age      = ATTENTION_WEIGHT_REPRODUCE_AGE * percept_w,
&
              weight_reprfac  = ATTENTION_WEIGHT_REPRODUCE_REPRFAC *
percept_w )

    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
    !> The **dominant** state is **reproduction**:
    else if ( this%motivations%reproduction%is_dominant() ) then
RESET_DOMINANT

        call this%motivations%hunger%attention_weight%attention_init
&
              (weight_light    = ATTENTION_WEIGHT_HUNGER_LIGHT * percept_w,
&
              weight_depth     = ATTENTION_WEIGHT_HUNGER_DEPTH * percept_w,
&
              weight_food_dir = ATTENTION_WEIGHT_HUNGER_FOOD_DIR * percept_w,
&
              weight_food_mem = ATTENTION_WEIGHT_HUNGER_FOOD_MEM * percept_w,
&
              weight_conspec  = ATTENTION_WEIGHT_HUNGER_CONSPEC * percept_w,
&
              weight_predator = ATTENTION_WEIGHT_HUNGER_PREDATOR * percept_w,
&
              weight_stomach  = ATTENTION_WEIGHT_HUNGER_STOMACH * percept_w,
```

```
&
            weight_bodymass = ATTENTION_WEIGHT_HUNGER_BODYMASS * percept_w,
&
            weight_energy   = ATTENTION_WEIGHT_HUNGER_ENERGY * percept_w,
&
            weight_age      = ATTENTION_WEIGHT_HUNGER_AGE * percept_w,
&
            weight_reprfac  = ATTENTION_WEIGHT_HUNGER_REPRFAC * percept_w )

      call this%motivations%avoid_passive%attention_weight%attention_init
&
            (weight_light   = ATTENTION_WEIGHT_AVOID_PASS_LIGHT *
percept_w,    &
            weight_depth   = ATTENTION_WEIGHT_AVOID_PASS_DEPTH *
percept_w,    &
            weight_food_dir = ATTENTION_WEIGHT_AVOID_PASS_FOOD_DIR *
percept_w,&
            weight_food_mem = ATTENTION_WEIGHT_AVOID_PASS_FOOD_MEM *
percept_w,&
            weight_conspec  = ATTENTION_WEIGHT_AVOID_PASS_CONSPEC *
percept_w, &
            weight_predator = ATTENTION_WEIGHT_AVOID_PASS_PREDATOR *
percept_w,&
            weight_stomach  = ATTENTION_WEIGHT_AVOID_PASS_STOMACH *
percept_w, &
            weight_bodymass = ATTENTION_WEIGHT_AVOID_PASS_BODYMASS *
percept_w,&
            weight_energy   = ATTENTION_WEIGHT_AVOID_PASS_ENERGY *
percept_w,  &
            weight_age      = ATTENTION_WEIGHT_AVOID_PASS_AGE * percept_w,
&
            weight_reprfac  = ATTENTION_WEIGHT_AVOID_PASS_REPRFAC *
percept_w )

      call this%motivations%avoid_active%attention_weight%attention_init
&
            (weight_light   = ATTENTION_WEIGHT_AVOID_ACT_LIGHT * percept_w,
&
            weight_depth    = ATTENTION_WEIGHT_AVOID_ACT_DEPTH * percept_w,
&
            weight_food_dir = ATTENTION_WEIGHT_AVOID_ACT_FOOD_DIR *
percept_w, &
            weight_food_mem = ATTENTION_WEIGHT_AVOID_ACT_FOOD_MEM *
percept_w, &
            weight_conspec  = ATTENTION_WEIGHT_AVOID_ACT_CONSPEC *
percept_w,  &
            weight_predator = ATTENTION_WEIGHT_AVOID_ACT_PREDATOR *
percept_w, &
            weight_stomach  = ATTENTION_WEIGHT_AVOID_ACT_STOMACH *
percept_w,  &
            weight_bodymass = ATTENTION_WEIGHT_AVOID_ACT_BODYMASS *
percept_w, &
            weight_energy   = ATTENTION_WEIGHT_AVOID_ACT_ENERGY *
percept_w,   &
            weight_age      = ATTENTION_WEIGHT_AVOID_ACT_AGE * percept_w,
&
```

```
                weight_reprfac  = ATTENTION_WEIGHT_AVOID_ACT_REPRFAC *
percept_w )

        !> @note Dominant **reproduction**.
        call this%motivations%reproduction%attention_weight%attention_init
&
            (weight_light    = ATTENTION_WEIGHT_REPRODUCE_LIGHT,
&
             weight_depth    = ATTENTION_WEIGHT_REPRODUCE_DEPTH,
&
             weight_food_dir = ATTENTION_WEIGHT_REPRODUCE_FOOD_DIR,
&
             weight_food_mem = ATTENTION_WEIGHT_REPRODUCE_FOOD_MEM,
&
             weight_conspec  = ATTENTION_WEIGHT_REPRODUCE_CONSPEC,
&
             weight_predator = ATTENTION_WEIGHT_REPRODUCE_PREDATOR,
&
             weight_stomach  = ATTENTION_WEIGHT_REPRODUCE_STOMACH,
&
             weight_bodymass = ATTENTION_WEIGHT_REPRODUCE_BODYMASS,
&
             weight_energy   = ATTENTION_WEIGHT_REPRODUCE_ENERGY,
&
             weight_age      = ATTENTION_WEIGHT_REPRODUCE_AGE,
&
             weight_reprfac  = ATTENTION_WEIGHT_REPRODUCE_REPRFAC )

     end if RESET_DOMINANT

  end subroutine gos_attention_modulate_weights
```

A4. R-script for statistical analysis

```
# Breakpoint linear regression, unconstrained, single breakpoint,
#   In this model x is ADF, y is AVERAGE GOS streak (average)
#
# Based on the method from:
#
https://www.r-bloggers.com/r-for-ecologists-putting-together-a-piecew
ise-regression/
#--------------------------------------------------------------------
-----------
# # SVN version info:
# $Id: script.breakpoint.R 3086 2017-03-20 19:02:56Z sbu062 $
#--------------------------------------------------------------------
-----------


####################################################################
###########
# Function to perform a breakdown linear model and determine a breakdown
point.
# the optimal breakdown is determined using the standard parametric sigma
# (standard deviation of the residuals) or AIC.
# NOTE: In the function ADF is the independent variable (x) and
#       AVERAGE is the dependent variable (y))
breakdown.linear.model <- function(ADF, AVERAGE,
                                   search_min=0.4, search_max=0.99,
                                   min_sigma=TRUE,
                                   xlabel= "Predictor",
                                   ylabel= "Response")
{

  # Make a variable to keep range of breakpoints
  breaks <- ADF[which(ADF >= search_min & ADF <= search_max)]


  #--------------------------------------------------------------------
---------
  # Iteratively search breakpoints for the model minimize residual MSE
or AIC
  mse <- numeric(length(breaks))  # Vector to keep residual MSE
  aics <- numeric(length(breaks)) # Vector to keep AIC values

  for(i in 1:length(breaks)){
   model.piecewise.part <- lm(AVERAGE ~ ADF*(ADF < breaks[i])
                                              +
ADF*(ADF>=breaks[i]))
    # Calculate residual standard deviation (sigma)
    mse[i] <- summary(model.piecewise.part)[6] # obtained from summary
    #mse[i] <- sigma(model.piecewise.part)     # or 'sigma' function
    # Calculate AIC, Akaike Information Criterion value
```

```
    aics[i] <- AIC(model.piecewise.part)
    }

  # Print actual breakpoint vector to search the optimum within.
  print("The range of breakpoints to optimise:")
  print(breaks)

  # MSEs AICs are keept in these vectors
  mse <- as.numeric(mse) # require it to make mse a vector
  print("Output all values of 'sigma' and AIC:")
  print(mse)    # print sigmas
  print(aics)   # print AIC

  print("Minimum AIC for the broken model:")
  print(min(aics))

  # The best model and respectively the optimal breakpoint is that which
  # minimises the standard deviation of the residuals (MSE) or AIC.
  min_mse <- breaks[which(mse==min(mse))]
  min_aics <- breaks[which(aics==min(aics))]
  print ("ADF Breakpoint based on sigma and AIC:")
  print(min_mse)    # print these values
  print(min_aics)

  # The breakpoint can be based either on MSE or AIC
  if ( min_sigma ) {
    point <- min_mse
    print("Optimisation is based on 'sigma'.")
    }
  else {
    point <- min_aics
    print("Optimisation is based on AIC.")
    }
  print("The actual breakpoint value is:")
  print(point)

  # Run the final model
  model.piecewise <- lm(AVERAGE ~ ADF*(ADF < point) + ADF*(ADF > point))
  print("Final fitted model parameters:")
  print( summary(model.piecewise) )


#----------------------------------------------------------------------
---------
  # Plotting the two-part linear regression
  # 1. basic scatterplot
  plot(ADF,AVERAGE, ylim = c(0,30),  pch=16, xlab=xlabel, ylab=ylabel)
  # 2. first part of the linear curve with parameter estimates from model
  # summary
  curve((model.piecewise$coefficients[1] +
model.piecewise$coefficients[3]) +
        (model.piecewise$coefficients[2] +
model.piecewise$coefficients[5]) * x,
```

```
        add=T, from=0, to=point)
  # 3. second part of the linear curvem after the breakpoint...
  curve((model.piecewise$coefficients[1] +
model.piecewise$coefficients[4]) +
        model.piecewise$coefficients[2] * x,
        add=T, from=point, to=max(ADF))
  # 4. vertical breakpoint line
  abline(v=point, lty=3)


#-----------------------------------------------------------------------
---------
  # Also plot the breakpoint minimum as bars of MSE or AIC
  print(mse)
  barplot(mse, names.arg = breaks,
            ylab="Standard deviation of residuals", xlab="Breakpoint")
  print(aics)
  barplot(aics, names.arg = breaks, ylab="AIC", xlab="Breakpoint")


}
######################################################################
##########

# Data analysis using this function

# Data are obtained from the CSV data file:
streaks <- read.csv("streaks4_switch.csv")

# Data is saved as 'streaks', attach first
attach(streaks)

# Do the data analysis: breakdown model
breakdown.linear.model(ADF, SWITCHES, 0.4, 1.0, FALSE, "ADF", "Number of
switches")

# Do additional data analysis: single line model
model.nobroken <- lm(SWITCHES ~ ADF)
summary(model.nobroken)
plot (ADF, SWITCHES, ylim=c(0,30), pch=16, ylab="Number of switches")
abline( summary(model.nobroken)$coefficients[1],
        summary(model.nobroken)$coefficients[2] )
print("AIC for the Single-line model:")
print(AIC(model.nobroken))

# Detach the working data frame
detach(streaks)
```

A5. Complete results

Probability of switching (%)			Attention Modulation Factor		
			Standard attention restriction	No attention restriction	Linear attention restriction
Arousal Dissipation Factor	Constant	0,95	4,2	19,5	0
		0,85	16,7	34,7	0
		0,20	30	34,7	8,4
	Function	Slow	3,3	15,4	0
		Intermediate	8,8	25,3	0
		Fast	30	38,1	0,6

Probability of re-evaluating (%)			Attention Modulation Factor		
			Standard attention restriction	No attention restriction	Linear attention restriction
Arousal Dissipation Factor	Constant	0,95	26,2	29	28,6
		0,85	44,5	49,2	46,3
		0,20	53	54,2	55,3
	Function	Slow	20	22,7	21,2
		Intermediate	37	38,5	36,7
		Fast	52,5	54,9	56,1

Switch ratio (switches / re-evaluations)			Attention Modulation Factor		
			Standard attention restriction	No attention restriction	Linear attention restriction
Arousal Dissipation Factor	Constant	0,95	0,16	0,67	0
		0,85	0,38	0,71	0
		0,20	0,57	0,64	0,15
	Function	Slow	0,17	0,68	0
		Intermediate	0,24	0,66	0
		Fast	0,57	0,69	0,1