

CHALMERS



Modeling Hydra Behavior Using Methods Founded in Behavior-Based Robotics

MALIN AKTIUS

Department of Applied Mechanics
Division of Vehicle Safety
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2007

Master's Thesis 2007:17

MASTER'S THESIS 2007:17

**MODELING HYDRA BEHAVIOR USING
METHODS FOUNDED IN BEHAVIOR-BASED
ROBOTICS**

MALIN AKTIUS

Department of Applied Mechanics
Division of Vehicle Safety
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2007

Modeling Hydra Behavior Using Methods Founded in Behavior-Based Robotics
MALIN AKTIUS
Master's Thesis in Engineering Mechatronics

© MALIN AKTIUS, 2007

Master's Thesis 2007:17
ISSN 1652-8557
Department of Applied Mechanics
Division of Vehicle Safety
Chalmers University of Technology
SE-41296 Göteborg
Sweden
Tel. +46-(0)31 772 1000

Chalmers Reproservice
Göteborg, Sweden 2007

Till Ann och Lennart

Modeling Hydra Behavior Using Methods Founded in Behavior-Based Robotics

Master's Thesis in Engineering Mechatronics

MALIN AKTIUS

Department of Applied Mechanics

Division of Vehicle Safety

Chalmers University of Technology

Abstract

In this thesis work methods from the field of behavior-based robotics are used to model the behavior of a biological organism. The cnidarian *Hydra* was selected as the model organism for this project, and the colony-style architecture was chosen as the method to be used for organizing the constituent behaviors of *Hydra* into an overall behavior. The individual behaviors were generated and organized in a simulated animal. Validation of the individual behaviors were carried out with respect to quantitative data available from previously conducted experiments on the animal. In the case of the overall behavior, which emerged from the organization of the individual behaviors using the colony-style architecture, only a qualitative comparison was feasible. Tests conducted on the modeled *Hydra* suggests that the colony-style architecture can be used to model behavior properties like latency, activation threshold, habituation, and duration of the individual behaviors of *Hydra*. Simulations of the overall behavior of the animal show that the spatial movement patterns of *Hydra* in the absence of external stimuli, as well as the effect of starvation on its contraction frequency, agrees with results from experiments conducted on the physical animal.

Keywords: behavior-based modeling, *Hydra*, colony-style architecture, evolutionary algorithms, recurrent neural networks.

Acknowledgments

First, I would like to thank my supervisor and examiner, Dr. Mats Nordahl. You taking on this thesis project enabled me to absorb in the fascinating field of behavior-based robotics during the final semester of my MSc studies. I am thankful for your good advice.

I am also deeply grateful to Carl-Henrik Fant, Johanna Hector, Mikael Carlsson, and Markus Kvist. Without your support during the past semester, I would not have been able to complete this thesis. Thank you.

Finally, thank you Fredrik for standing beside me during this time.

Malin Aktius
Göteborg, 2007

Table of Contents

1	Introduction	1
1.1	Objectives	3
1.2	Related work	3
1.3	Outline	4
2	Notation	5
3	Animal behavior	7
3.1	Sensory system and motivational state	8
3.2	Patterns of movement	9
3.2.1	Reflexes	9
3.2.2	Fixed action patterns	10
3.2.3	Orientation of movement	10
4	Behavior-based robotics	13
4.1	Generation of robot behaviors	13
4.2	Organization of robot behaviors	14
5	Hydra	15
5.1	<i>Hydra</i> as a model organism	15
5.1.1	Physiology	16
5.1.2	Sensory cells	17
5.1.3	Behavior repertoire	17
5.1.4	Behavior coordination	18

6	Models and methods	21
6.1	The simulated system	21
6.1.1	Movement patterns	22
6.1.2	Exteroceptive sensory system	23
6.1.3	Proprioceptive sensory system	23
6.1.4	Motivational state	24
6.1.5	Environment	24
6.1.6	Model parameters	25
6.2	Behavioral organizer	25
6.2.1	Applicability clause	26
6.2.2	Transfer function	26
6.3	Constituent behaviors	26
7	Results and discussion	29
7.1	Generation of behaviors	29
7.1.1	B1: Spontaneous actions	30
7.1.2	B2: Respond to mechanical stimulus	31
7.1.3	B3: Respond to light stimulus	37
7.1.4	B4: Feeding	38
7.2	Generation of behavioral organizer	40
7.2.1	Experiment A: Spontaneous locomotion patterns	40
7.2.2	Experiment B: Effect of starvation on contraction frequency	40
8	Summary and conclusions	45
	Bibliography	47
A	Evolutionary algorithms	53
A.1	Encoding scheme	53
A.2	Fitness function	54
A.3	Selection	54
A.4	Crossover	55
A.5	Mutations	55
A.6	Elitism	55

B	Recurrent neural networks	57
B.1	Network equations	57
B.2	Learning algorithms	59

Introduction

During the past decades **autonomous robots** have undergone rapid development, and it is expected that such robots will play an increasingly important role in our society in the future [4, 5, 57]. Lawn mower robots, vacuum cleaning robots, and various entertainment robots reached the market several years ago, while robots designed for domestic assistance, service, and surveillance are under development. Examples of various autonomous robots, and their application areas, can be found in [4, 5, 58]. Autonomous robots are expected to operate in **unstructured environments**, i.e. environments that change in an unpredictable way. For such robots traditional control strategies defined *a priori* are often insufficient, since the robot must be able to perceive and adapt to changes in its environment in order to function successfully.

There are many different approaches to the problem of generating brains¹ for autonomous robots. In classical **artificial intelligence** (AI) the flow of information is arranged in a sequential way: the robot uses information acquired through its sensors to build a world model, based on which it reasons about the effect of various actions. As a last step, the robot then decides upon an action, which is executed. This sense-plan-act approach of classical AI requires implementation of high-level reasoning, something that turned out to be difficult, as pointed out e.g. in [9, 67]. Thus, while classical AI has been successfully applied in various fields, e.g. pattern recognition, its goal of generating intelligent artificial agents has not (yet) been reached [67].

An alternative to the classical AI approach to machine intelligence, **behavior-based robotics** (BBR) was introduced in the mid 1980s, by R. Brooks and others. See e.g. [8] for one of the pioneering works in BBR. In contrast with classical AI, behavior-based robotic brains are designed using a parallel, decentralized architecture. In BBR the aim is to organize simple, often **reactive behaviors**² in such a way to make the robot function adequately in an unstructured environment. (Introductions to BBR and AI can be found

¹Sometimes also referred to as control systems.

²Direct coupling between sensors and actuators.

e.g. in [1, 37], and in [55], respectively.) With this aim, BBR has taken on a broader definition of intelligence than classical AI. This, however, is in analogy with nature, as argued e.g. in [40]. With reactive behaviors as building blocks of the robotic brain, behavior-based artificial systems are often modeled using biology as inspiration [3]. Thus, rather than facing the problem of modeling high-level reasoning, one of the main challenges of BBR is to design a mechanism that will allow the robot to activate the most appropriate behavior at any given time. This is referred to as the **action selection problem**³ [34], and many methods have been suggested in order to resolve it, as is discussed in Section 4.2. Throughout this thesis, the terms **action selection**, **behavior selection**, **behavior organization**, and **behavior coordination** will be used interchangeably.

Turning to nature, **ethology**⁴ studies show that even rather simple animals are capable of quite complex overall behavior, provided that they operate in their natural environment. Examples can be found e.g. in [27, 28], where the behavior of several unicellular and lower multicellular organisms have been studied, e.g. the bacterium *Paramecium*, the flatworm *Dugesia*, and the cnidarian *Hydra*. Furthermore, in [13] learning and memory in the much studied nematode worm *C. Elegans* is described. Albeit referred to by man as simple biological organisms, the behavior of *Paramecium*, *Dugesia*, *Hydra*, and *C. Elegans* is intelligent in the manner that the organisms survive and reproduce in their natural, yet unstructured, habitats. How can these simple creatures be capable of such behavior? The answer lies, of course, in the process of evolution. In simple animals, behavior and behavior selection occurs not as a result of reasoning or trial-and-error behavior in the individual animal, but rather as a result of trial-and-error on an evolutionary scale. During the life of its species, these simple organisms have developed skills that make them capable of functioning successfully, i.e. survive long enough to reproduce, in their natural environment. Several examples of evolutionary development and adaption in nature can be found in the literature, e.g. in [15, 16, 17]. Using a minimal amount of memory and “reasoning” in their adaptive behavior, simple biological organisms make good models for artificial behavior-based systems.

Though biology often serves as a source of inspiration in the design of artificial behavior-based systems, the degree of biological inspiration is, however, quite diverse: ranging from classical control systems using only vague arguments founded in biology, to actual replication of biological behavior [3]. Examples of **biorobotic**⁵ models include ant navigation, bird flocking, cockroach escape, and owl sound location, as discussed in [70]. While models of individual biological behaviors have been frequently used in BBR, there are few examples of behavior-based models of an entire biological organism, taking also the organization of its constituent behaviors (i.e. the way in which the behaviors of the animal’s behavior repertoire should be combined to form an overall behavior) into account.

³Sometimes also called the behavior coordination problem.

⁴The study of animal behavior under natural conditions.

⁵The intersection of biology and robotics.

Such modeling may eventually give important clues on how robust, adaptive behavior can be accomplished also in artificial systems REFS!!. This thesis focuses on behavior-based modeling of a simple biological organism, taking into account the constituent behaviors of its behavior repertoire, as well as the behavior organization system.

1.1 Objectives

The objectives of this project are:

1. A literature survey of candidate organisms, followed by selection of one appropriate biological organism to be modeled in a behavior-based manner.
2. Based on literature studies on existing methods in BBR, to select one appropriate method for behavior-based modeling of the selected organism.
3. To model, using the selected method, the constituent behaviors as well as the behavioral organization system for the selected organism.

1.2 Related work

As previously mentioned, and as discussed e.g. in [70], several researchers have proposed models of behavioral subsystems in biological organisms.

In [61, 62], Staddon proposes a model for decision upon straight-line swimming, or tumbling behavior in bacteria. The model is used to represent the fact that bacteria move up (or down) a gradient of e.g. the concentration of a chemical substance. (Bacteria use an indirect form of gradient-following, called **kinesis**, which will be discussed in Section 3.2.3.) Using a differentiator model in discrete time, with a threshold added to its output, Staddon arrives at a model for straight-line swimming vs. tumbling behavior of bacteria.

In the literature mentioned above, Staddon also proposes a model for avoidance behavior in the single-celled *Stentor*, based on Jennings' detailed studies [27, 28] of the organism. Jennings identified four distinct avoidance behaviors carried out by *Stentor* on the encountering of a noxious substance. In the model proposed by Staddon, the response of each avoidance behavior is represented by means of a leaky integrator, and the responses are then connected in a winner-take-all circuit [62].

In [2] parts of the visuo-motor behavior (such as prey-acquisition, mating, predator-avoidance, obstacle-avoidance, and the "chantlitalia" behavior⁶) of the praying mantis⁷ is modeled using the framework of schema theory, and implemented into a hexapod robot.

⁶The search for a hospitable environment.

⁷A large, carnivorous insect.

1.3 Outline

Following the introduction in this chapter, this thesis is arranged in the following manner:

Chapter 2: In this chapter, an overview of recurring abbreviations and notations used throughout this thesis is presented.

Chapter 3-4: These chapters introduce relevant theory for this project, aiming at putting the problem into context. Chapter 3 provides a background to animal behavior, whereas Chapter 4 introduces the framework of behavior-based robotics.

Chapter 5: This chapter presents the model organism of choice for this thesis, and concludes with a description of that organism from a behavioral point of view.

Chapter 6: Here, the models and methods used in this project are described.

Chapter 7: This chapter presents and discusses the resulting behavioral model in terms of the tests conducted, and the results obtained from these experiments.

Chapter 8: This final chapter presents the conclusions drawn, and recommendations for future work.

Appendix A-B: Appendices A and B present a brief introduction to evolutionary algorithms and recurrent neural networks, respectively.

Chapter 2

Notation

This chapter provides a list of recurring abbreviations, as well as notations used throughout this thesis. In general, technical terms are typeset in **bold** on their first occurrence. Names of biological organisms are typeset in *italics*.

Abbreviations

AC	Applicability clause
AI	Artificial intelligence
ANN	Artificial neural network
BBR	Behavior-based robotics
CNS	Central nervous system
CP	Contraction pulse
CSA	Colony-style architecture
EA	Evolutionary algorithm
ER	Evolutionary robotics
ES	Evolution strategies
ESS	Evolutionary stable strategy
FAP	Fixed action pattern
FFNN	Feed forward neural network
FSM	Finite state machine
GA	Genetic algorithm
GP	Genetic programming
GSH	Glutathione
ISI	Inter-stimulus interval
LP	Locomotion pulse
RMSE	Root mean squared error
RNN	Recurrent neural network
TF	Transfer function

Upper-case letters

B	Behavior
$E(\lambda)$	Exponential distribution with expected value $1/\lambda$
M	Motivational state variable
R	Stochastic variable representing rate
T	Threshold
$U(a, b)$	Uniform distribution on $[a, b]$

Lower-case letters

b	Bias term
m	Mutation operator
p	Probability
w	Weight parameter

Greek letters

μ	Expected value
τ	Time constant

Subscripts

beh	Behavior
c	Chemical
con	Contraction
ext	Extension
f	Foot, Feeding
h	Hunger
l	Light
loc	Locomotion
m	Mechanical

Chapter 3

Animal behavior

The aim of this work is to model, using the framework of BBR, the overall behavior of a simple biological organism. The following two chapters introduce the relevant theory for that approach. In this chapter, some remarks concerning modeling animal behavior will first be given, aimed at putting the problem into context. The chapter then continues with a brief introduction to animal behavior. In the next chapter, an introduction to the relevant theory from BBR for this thesis project will be given.

First, it should be noted that modeling behavior in biological organisms (in any but the simplest cases) concerns modeling a system which is far from completely understood. When modeling biological behavior, not only physical laws apply, but also physiological conditions, internal drives, and environmental context [72]. Also complicating the issue is the interaction between success and survival at the individual and species levels. Hence, inevitably a number of questions arise regarding the purpose and aim of the modeling task. In e.g. [18, 35, 65], behavioral modeling is discussed in more detail. See also e.g. [21] for a more general introduction to the modeling of biological systems. In addition, B. Webb has addressed the question of modeling biological behavior by defining seven dimensions on which models of biological behavior can differ [70]. While a thorough discussion on aspects of biological behavioral models is beyond the scope of this text, it should be mentioned that a behavioral model of a biological organism may aim at describing the organism on one or more out of several levels. In [1], it is argued that the field of BBR can gain from the study of at least three such levels, namely **neuroscience**¹, **psychology**², and ethology (see Chapter 1), as described below.

Neuroscience: This level concerns the neural processes underlying behavior, e.g. neural network properties such as synaptic transmission. See e.g. [30, 44] for the modeling of neural processes.

¹The study of the nervous system.

²The study of mind and behavior.

Psychology: Here, behavior is modeled from a functional aspect, using mental functions such as memory, emotions, and cognitive processes as the bases of behavior. Often, knowledge of the human mind is used also for animal behavior. Different schools of psychology include **behaviorism**, **ecological psychology**, and **cognitive psychology**. See e.g. [1] for a general description of psychological models used in BBR. Introductions to behaviorism, ecological psychology, and cognitive psychology can be found in [69], [19], and [20, 26], respectively.

Ethology: Like psychology, ethology uses a functional approach to model behavior. However, ethology focuses on the biological aspect, i.e. the anatomical and physiological properties of the animal, rather than the psychological. Also, it stresses the importance of natural environmental context, i.e. that the behavioral system not only includes the animal, but also the environment in which it lives. See e.g. [36, 39, 60].

In addition to the disciplines mentioned above, an evolutionary approach is also commonly used in biological modeling. This approach includes fields like **sociobiology** and **evolutionary psychology**, where evolutionary advantages of social behaviors and mind functions, respectively, are considered. In evolutionary behavioral models, only those behaviors that are beneficial for a population, in the sense that no alternative behavior is more advantageous, survive on an evolutionary time scale. This argument was formalized by Maynard Smith [38], and is termed **evolutionary stable strategy** (ESS).³ For more information on evolutionary biological modeling, see e.g. [38].

Different disciplines, as the ones just mentioned, of behavioral study contribute to behavioral modeling in different ways. This work, aiming at a behavior-based model of a simple biological organism, will solely concern the ethology level. In Section 3.2 below, the functional breakdowns of animal behavior, according to ethology, will be discussed.

3.1 Sensory system and motivational state

The sensory system of an organism can be divided into **exteroceptive** sensors, and **proprioceptive** sensors [4]. Exteroceptive sensors are used to gather information from the external environment in which an organism is situated, while proprioceptive sensors monitor the internal state of the organism.

As stated in e.g. [40], it is the **motivational state** of an animal that gives rise to any behavior. The motivational state consists of the combination of **perceptual state**⁴, and **physiological state**⁵, as well as the current behavior of the animal. Thus, the motivational state of an animal depend on external as well as internal factors, as shown in Fig. 3.1 below.

³ESS is based on game theory and adds one condition to the Nash equilibrium, see [38].

⁴The environment, as perceived by the exteroceptive sensory system.

⁵The physiology of the animal, as monitored by the proprioceptive sensory system.

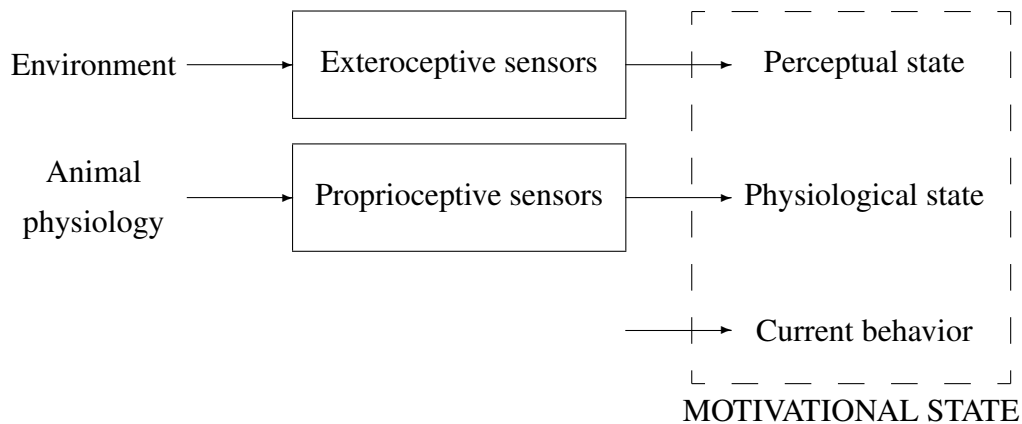


Figure 3.1: Motivational state of an organism.

3.2 Patterns of movement

Behavior definition in animals is strongly connected to the movements of the animal. Animal behavior is commonly divided into three comprehensive classes [1, 60], namely **reflexes**, **orientation of movement**, and **fixed action patterns**, which will now be described briefly. A more thorough description can be found e.g. in [18, 36, 39, 60, 62].

3.2.1 Reflexes

A reflex is an automatic response to an external stimulus. While reflexes in humans are regarded as simple, involuntary actions, they constitute the entire behavior repertoire of simple biological organisms. The intensity and duration of a reflexive response generally depend on stimulus configuration. Reflexes share a number of properties, as discussed in e.g. [36, 60]. Below is a list of some of these properties, as used in this thesis project.

Threshold: Normally, the stimulus for any reflex must rise above a certain minimum level, known as the reflex threshold, in order to evoke a response.

Latency: The time between the onset of a stimulus and the occurrence of a response.

Refractory period: Following a reflexive response, the threshold for the specific reflex may be elevated briefly, thus generating a period of time during which it is unlikely that a stimulus may evoke any response.

Temporal summation and sensitization: Temporal summation refers to the fact that two or more sub-threshold stimuli may evoke a response, if they occur within a short

period of time. Related to temporal summation is sensitization, where a reflexive response may be more easily evoked (in practice having a lower threshold) following repeated stimulus presentation.

Spatial summation: Two or more stimuli applied within a short distance from each other may evoke a response where one stimulus would fail to do so. Thus, a reflexive response has a receptive field.

Momentum (after-discharge): The response of some reflexes will outlast the stimulus evoking the response. The duration as well as the magnitude of the after-discharge generally increase with stimulus intensity.

Habituation: Repeated reflex responses may eventually lead to a decrease in response intensity, and finally to no response at all. Habituation to a weak stimulus generally occurs sooner than to a strong one. Related to habituation is spontaneous recovery, the fact that the effect of habituation ceases with time.

Fatigue: Sense organs tire out, so that the animal is unable to detect stimulus, or muscles become exhausted, which make them unable to perform any movement.

Reflexes may interact in various ways, e.g. by competition, thus inhibiting each other, or by cooperation. The adaptive behavior emerging from sensitization and habituation is regarded as the simplest form of learning in biological organisms. Both processes, however, are reversible and thus the organism will eventually return to its original state if left alone for a sufficient long period of time. In general, this is not true for other types of learning mechanisms.

3.2.2 Fixed action patterns

Fixed action patterns (FAPs) is defined by a stereotyped series of actions following a releasing stimulus. Thus, the response of an FAP typically outlasts the triggering stimulus, but in contrast to reflexive behavior, the duration of the response is not dependent on stimulus characteristics such as duration and intensity. See e.g. [40, 60] for a discussion on, and examples of, FAPs.

3.2.3 Orientation of movement

The orientation of animal movement may be roughly divided into three categories, i.e. **kinesis**, **taxis**, and **navigation**.

Kinesis: Indirect following of a gradient for orientation of movement. As an example, bacteria are, in general not capable of detecting a gradient of the concentration of a

chemical substance [62]. However, by making use of time dimension, they are able to sample a gradient which can be used to reorient movement.

Taxes: Direct gradient following. Gradient ascent (or descent) in response to a perceived directed stimulus.

Navigation: More complex orientation of movement is found e.g. in homing pigeons, which use the sun as a compass to find the direction of their home location. Bees and ants, for example, use landmarks to guide them on their routes. These more complex navigation strategies will not be considered in this project, but the interested reader is referred to e.g. [36, 39, 40, 60].

Chapter 4

Behavior-based robotics

As mentioned in Chapter 1, BBR (in contrast with classical AI) takes a bottom-up approach to robotic behavior, and the complete robotic brain is built from a repertoire of simple behaviors. It should be mentioned that, more recently, **hybrid architectures**¹ have become increasingly common [48], thus filling the gap between BBR and AI somewhat. This chapter provides a brief introduction to the framework of BBR.

The main characteristics of behavior-based systems are **low-level intelligence**, i.e. a tight connection between the robot's sensors and its actuators, and decomposition of the robot's overall behavior into individual behaviors, each of which represents a distinct functional unit [1]. The generation and, in particular, organization of behaviors constitute vast field of studies, and many methods have been suggested by researchers in the field. It should be noted that methods for the generation and organization of behaviors apply not only to architectures in BBR, but also to most hybrid architectures [48]. More information concerning generation and organization of behaviors in BBR can be found e.g. in [1, 47, 67, 71]. For a review of current methods used for behavior coordination, see e.g. [49].

Generation and coordination of robot behaviors can be accomplished by means of hand-coding and traditional engineering techniques, or by means of artificial evolution, in which **evolutionary algorithms** (EAs) [25] are used as an optimization method. The latter case forms the field of **evolutionary robotics** (ER) [42]. A brief introduction to EAs is provided in Appendix A.

4.1 Generation of robot behaviors

As previously mentioned, in BBR relatively simple behaviors constitute the building blocks of the robotic brain. Several architectures for behaviors exist, e.g. **if-then-else-**

¹Combining reactive behaviors from BBR, and deliberative plans from traditional AI.

rules, various modifications of **finite state machines** (FSMs)², and **artificial neural networks** (ANNs). If-then-else-rules provides a quick way of generating simple behaviors by means of hand-coding instructions directly into the robotic brain. As a simple example consider a robot that moves straight ahead while there is no object in front of it, and stops at the encountering of an obstacle. Assuming that conditions for obstacle-detection by the robot’s sensors, as well as motor signals for the “move forward” and “stop” behaviors have been defined, this behavior may be formulated as follows: *if no object detected then move forward else stop*.

Biologically inspired architectures, such as ANNs, often provide a suitable framework for the generation of robot behaviors by means of EAs. Especially in cases where it is difficult to arrive at an explicit model for a specific behavior, such an approach has an obvious advantage over hand-coded behaviors. A brief introduction to ANNs is provided in Appendix B.

4.2 Organization of robot behaviors

In a behavior-based system, the behavioral organizer specifies how the constituent behaviors are arranged and connected. As mentioned in Chapter 1, many methods have already been suggested, and new architectures continue to arise. One way of classifying methods used for behavior organization, as discussed in [49], is to divide them into two main categories: **arbitration methods** and **command fusion methods**, as suggested e.g. by A. Saffiotti [56]. In arbitration methods, only one behavior is active at a time, whereas in command fusion methods, the robot’s action is obtained by combining the results from several behaviors. In both arbitration methods and command fusion methods, several sub-categories exist, as shown in Fig. 4.1 below. In this project, a priority-based arbitration method was used, which will be described in Chapter 6. For descriptions of other methods, see e.g. [1, 49].

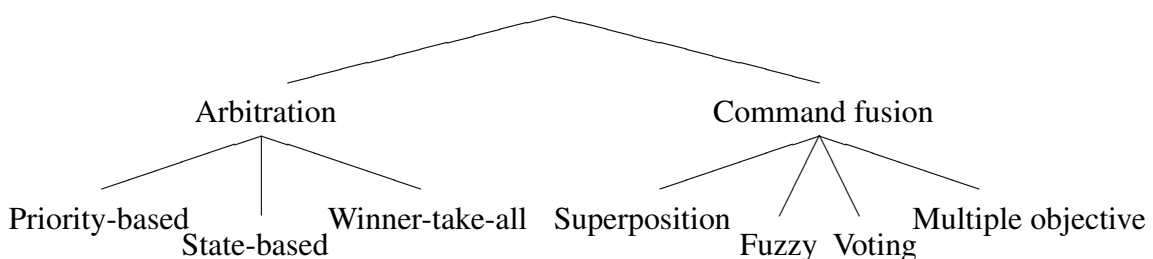


Figure 4.1: Classification of methods for behavior organization.

²Examples include augmented finite state machines, used e.g. in [8], and generalized finite state machines, used e.g. in [68].

Hydra

This chapter presents the biological model organism of choice for this project, *Hydra*. The motivation for selecting *Hydra* as the model organism will first be given, followed by a general description of the organism (in a very condensed form). Lastly, the behavior repertoire of *Hydra*, as described in the literature, is presented.

For this project, several biological organisms were considered as possible model organisms. As mentioned in Section 1.2 Staddon has, based on Jennings' studies [27, 28], modeled behavior selection in bacteria, and in *Stentor* [62]. At least in the case of *Stentor*, Staddon's model could be expanded to include behaviors not only for the attached *Stentor*, but also for the free swimming one. To allow also for modeling of internal processes in the organism, such as e.g. an active feeding behavior, some multicellular organisms were considered as well. The nervous system of *C. Elegans*, consisting of 302 neurons, has been completely mapped out, and the animal has been the subject of numerous studies. Flatworms, such as *Dugesia*, were also considered. Both *C. Elegans* and *Dugesia* are capable of associative learning [6, 13, 28, 50]. While this type of learning in animals is a fascinating topic in its own right, it restricts the possibility of repeating experiments concerning behavior selection. As discussed in Section 3.2.1, habituation is a reversible process, something that is not generally true for associative learning. *Hydra* is an organism more complex than unicellular bacteria and *Stentor*, but less complex than *C. Elegans* and flatworms. It has an active feeding behavior but it is not capable of associative learning. The complexity level of *Hydra*, as well as the number of quantitative and qualitative behavioral studies available, made *Hydra* the model organism of choice for this project.

5.1 *Hydra* as a model organism

Hydra, shown in Fig. 5.1, belongs to the phylum Cnidaria, the first evolved animals (that still exists) to possess nerve cells and sense organs. *Hydra* lives in fresh-water ponds,

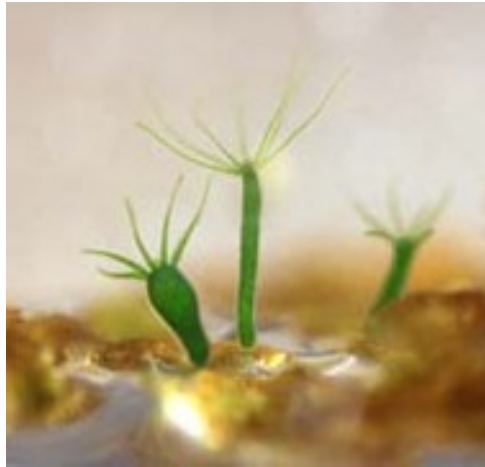


Figure 5.1: The cnidarian *Hydra*. Image courtesy of BioMedia Associates [7].

lakes, and streams. It is most often found attached to some vegetation by the base of its tubular body¹, where it feeds on small aquatic invertebrates, e.g. the multicellular arthropod *Daphnia* or the single-celled protozoan *Euglena* [24]. *Hydra* lives and reproduces sexually and asexually, in the latter case by means of budding. Most cnidarians have a free swimming medusa stage. *Hydra*, however, does not, and hence it remains in its tube-shaped polyp form throughout its life. Because of their radial symmetry, cnidarians are commonly referred to as radiate animals.

5.1.1 Physiology

Hydra is, like other cnidarians, designed around two layers of cell tissue: the outer **ectoderm** layer, and the inner **endoderm** layer. These two cell tissues are separated by a gelatinous section, named the **mesoglea**. The endoderm layer surrounds the animal's gut, which only opening is located at the **hypostome**². *Hydra* does not possess a **central nervous system** (CNS), but rather its nerve cells are organized in a **nerve net** that extends through the body, with a limited number of sensory inputs and motor outputs. Nerve cells are spread throughout the ectoderm as well as the endoderm tissue, and consist of two cell types: **ganglion neurons**³, and **sensory neurons**. However, it is believed that these neurons of *Hydra* actually are multifunctional, thus each possessing the functions of sensory-, motor-, and inter-neurons [63]. *Hydra* has two types of effectors, or motor cells, namely (1) muscular cells of the endoderm and the ectoderm layer, respectively, and (2) nematocysts, which are stinging organells, located on the tentacles of the animal. The muscle

¹Henceforth, the base of *Hydra's* body will be referred to as its foot.

²Area at the top of the animal's body, and the base of its tentacles.

³Interconnected neurons that either process sensory information or control motor outputs.

cells are responsible for the body elongation and shortening, as well as locomotion of the animal, whereas the nematocysts are used for paralyzing prey during capture.

5.1.2 Sensory cells

Exteroceptive sensors of *Hydra* consist of: (1) **Chemoreceptors**, capable of detecting e.g. the peptide glutathione (GSH), which is involved in activating the feeding behavior of *Hydra*. (2) **Mechanoreceptors**, for detection of physical contact. (3) **Photoreceptors**, for detection of light conditions. Whereas chemoreceptors and mechanoreceptors have been identified [29, 51], there is only indirect evidence for the existence of photoreceptors [63]. Proprioceptive sensors monitor (1) the nutritional state, and (2) the adapted light condition (background illumination).

5.1.3 Behavior repertoire

The distinct movement patterns of *Hydra*, resulting from alternating activity of its motor cells, consist of (1) contraction and expansion of body and tentacles, (2) digestion, and (3) locomotion [63]. Locomotion is accomplished either by gliding (by means of cilia on the foot), or by somersaulting. It should be noted that most of the behavior in *Hydra*, characteristic for lower animals, is not specific but general. Thus, the animal reacts in a way that is usually beneficial, rather than to the specific situation [28].

Given the theory of animal behavior and behavior-based modeling presented in Chapters 3 and 4, the overall behavior of *Hydra* will now be described as the coordination of distinct behaviors.

Spontaneous actions

Without changes in *Hydra*'s external environment, it shows spontaneous, periodic contractions or locomotion. There is an adaption to background illumination in the sense that the contraction frequency varies with the ambient light conditions. The frequency also depends on the nutritional state of the animal: contraction frequency decreases with starvation, while locomotion is more common in starved *Hydra*. After one week of starvation, practically any overt behavior ceases, and the animal eventually perishes [46].

Response to mechanical stimulus

Response to mechanical stimulus, such as shaking or physical contact, occurs by means of contraction or locomotion. The way in which *Hydra* responds depends on stimulus history as well as on nutritional state of animal: starved animals are more likely to respond by locomotion, whereas a contraction response is more common in well fed animals. The response shows habituation to repeated stimulus [53].

Response to light stimulus

On exposure to strong light there is an immediate inhibition of any ongoing contraction. Following a latency, a response consisting of either contraction or locomotion is evoked. As in the case of mechanical stimulus, locomotion is more common in starved animals. The latency is inversely related to light intensity [59]. There is no habituation [53].

Feeding

The feeding behavior of *Hydra* consists of a sequence of actions, namely prey capture, mouth opening, ingestion, digestion, and regurgitation⁴. The behavior is evoked by mechanical stimulation of *Hydra's* tentacles or by the presence of GSH, which is released by prey stung by nematocysts. The behavioral threshold is regulated by *Hydra's* nutritional state, with starved animals exhibiting a lower threshold than recently fed ones. Also, there is a refractory period following a feeding response, with the capacity of responding being gradually regained [32, 33].

5.1.4 Behavior coordination

As described in the literature, feeding inhibits responses to light and mechanical stimuli. In addition, response to light inhibits response to mechanical stimuli, thus forming a priority-based relation of *Hydra's* behaviors [32, 54].

The behaviors and sub-behaviors of *Hydra*, as modeled in this project, are summarized in Tables 5.1 and 5.2 below.

Table 5.1: Behaviors shown by *Hydra*. Priority 1 denotes the highest priority.

Label	Priority	Description
B1	4	Spontaneous actions
B2	3	Response to mechanical stimulus
B3	2	Response to light stimulus
B4	1	Feeding

Table 5.2: Sub-behaviors of *Hydra*

Label	Description
B11, B21, B31	Contraction/Extension
B12, B22, B32	Locomotion

⁴Through regurgitation, stomach contents is transferred back into the mouth, and in this way *Hydra* gets rid of undigested remains of the prey.

Models and methods

In this thesis the overall behavior of *Hydra* is modeled by defining individual behaviors, as well as a behavioral organizer for coordination of those behaviors. The model of *Hydra* aims at describing the behavior of the animal as functions of its motivational state. For the initial purpose of validating the behavioral model, a simulated system was developed. Also, once a behavioral model has been validated, such a system can be used to predict the behavior of the animal in various environmental set-ups. In this project, however, the simulated system was only used for validation experiments, as discussed in Section 7.2. This chapter starts with a description of the simulated system, and concludes by presenting the methods used for generating and organizing the behaviors of *Hydra*.

6.1 The simulated system

Hydra was modeled ignoring the dynamical properties of its body. It should be noted that, as pointed out e.g. in [10, 12], the body of an animal is indeed connected to its behavior. Thus, this simplification was made not because the properties of *Hydra*'s body are unimportant for its behavior, but rather in an attempt to isolate, as much as possible, and focus on the behavioral traits in this first work towards a behavioral model of the animal. Also, the time aspect of this project was a factor in making this simplification. As will be discussed in Chapter 8, an expansion of the model to include a more realistic morphology of *Hydra* might very well be a task for future work.

In this simplified model of *Hydra*, the animal's body is represented in 2D, by a circle with radius r and maximum extension length l_{max} . The movement of the animal is controlled simply by setting the corresponding states directly, which is described in Section 6.1.1. The physical state variables describing the body of the simulated *Hydra* is given in Table 6.1. In the 2D model of *Hydra*, the position of the animal's hypostome, rather than of its foot, was taken as the interface to (the simulated) environment. While this approach

might seem far-fetched at first, the reason for it is twofold: (1) This allows for simulation of the change in position of the body and tentacles caused by contraction and expansion, while the animal remains attached by its foot. (2) It has been argued that the hypostome and the tentacles are the most sensitive regions of *Hydra* [28, 66].

Table 6.1: Physical state variables for the simulated *Hydra* in a 2D environment. For the extension rate, r_{ext} , 0 denotes maximal contraction, and 1 maximal extension. S denotes the size of the arena.

Variable	Range	Description
x_f	$[0, S]$	Position of foot
y_f	$[0, S]$	Position of foot
θ_{ext}	$[0, 2\pi]$	Direction of body extension
θ_{loc}	$[0, 2\pi]$	Direction of animal locomotion
r_{ext}	$[0, 1]$	Extension rate

The interface area between *Hydra* and its environment, in the simulated system, is taken as a circle with radius, r , and center in

$$\begin{aligned} x &= x_f + r_{ext} \cdot l_{max} \cdot \cos(\theta_{ext}), \\ y &= y_f + r_{ext} \cdot l_{max} \cdot \sin(\theta_{ext}). \end{aligned} \quad (6.1)$$

Settings of the model parameters are specified in Section 6.1.6.

6.1.1 Movement patterns

As described in Section 5.1.3, behaviors B1-B3 all generate movement of *Hydra* by (1) contraction, and expansion of its body, or (2) locomotion. These movements are evoked by bioelectric contraction bursts and locomotion bursts, respectively, in pacemakers of the animal, as described e.g. in [46] and in [64]. While, as discussed in Chapter 3, no attempt is made in this project to model *Hydra* on a neural level, the concept of **contraction pulses** (CPs) and **locomotion pulses** (LPs) is adopted. Thus, the behavioral output will, at times, consist of a CP or an LP. The effect of these behavioral outputs on the state variables of the simulated *Hydra* is described below.

Hydra movement during contraction pulses

A contraction pulse has an effect on the simulated *Hydra* according to

$$r_{con}(t + dt) = ar_{con}(t) + bc_p(t), \quad (6.2)$$

where $r_{ext} = 1 - r_{con}$, and $c_p(t)$ is the CP signal (1 during a CP, 0 otherwise) at time t . An example is shown in Fig. 6.1. At the end of a CP, the extension heading, θ_{ext} , of the animal is set to a new, random value within its allowed range.

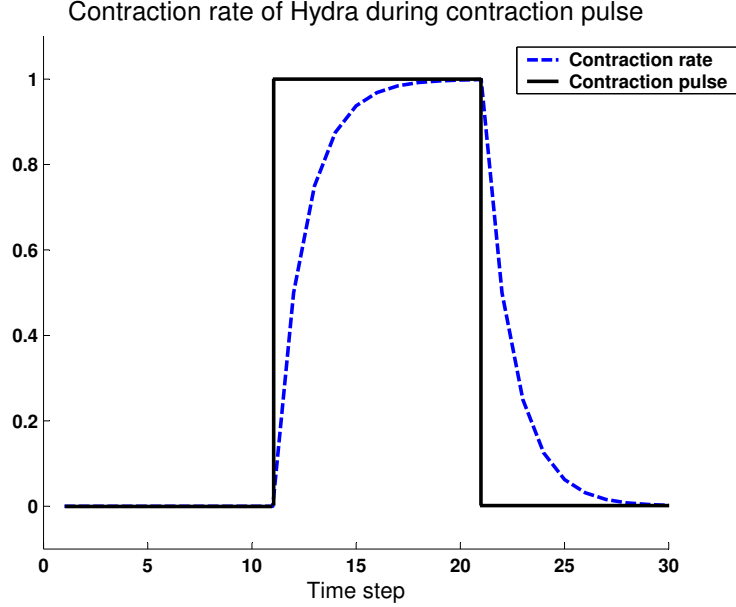


Figure 6.1: Contraction rate, r_{con} , of *Hydra* during a contraction pulse, using Eq. 6.2.

***Hydra* movement during locomotion pulses**

On initiation of an LP the locomotion direction, θ_{loc} , is set to a random value within its allowed range. During each time step of an LP, the position of *Hydra* is updated in the following way:

$$\begin{aligned} x_f(t + dt) &= x_f(t) + dl \cdot \cos(\theta_{loc}), \\ y_f(t + dt) &= y_f(t) + dl \cdot \sin(\theta_{loc}), \end{aligned} \quad (6.3)$$

where $dl = c_{LP} \cdot l_{max} \cdot \frac{dt}{T_{LP}}$. Here, T_{LP} is the duration of an LP and c_{LP} is a parameter determining the length of a locomotion step, as specified in Section 6.1.6.

6.1.2 Exteroceptive sensory system

Hydra reacts to chemical, light, and mechanical stimuli, as discussed in Section 5.1. In the simulated system, sensors for detecting GSH concentration, as well as the intensity of light and mechanical stimuli are modeled. Since *Hydra* cannot determine the location of a stimulus, sensor readings are simply taken as the maximal stimulus intensity on the area of *Hydra*'s body. Alternatives to this approach were considered, such as modeling spatial

summation, as described in Section 3.2.1. Albeit a common reflex property, indication of spatial summation in *Hydra* was not found in the literature, and thus it did not seem motivated to use such an approach.

6.1.3 Proprioceptive sensory system

As mentioned in Chapter 5, the proprioceptive sensory system of *Hydra* monitors the nutritional state as well as the adapted light condition of the animal. Clearly, these properties should ideally be implemented also in a model of the animal. One problem with trying to model the adapted light condition of *Hydra* is that, as pointed out in Section 5.1.2, there is only indirect evidence of photoreceptors in *Hydra*. Also, the data found in the literature concerning the influence of adapted light condition on behavior does not contain information on background illumination conditions [46]. Due to the two factors just mentioned, the adapted light condition was left out in this project, and thus only the nutritional state of *Hydra* was modeled in the proprioceptive sensory system. During simulations the nutritional state, M_h , is updated according to

$$M_h(t + dt) = \max \left(0, M_h(t) - dt \frac{h_{max}}{f_{max}} \right) \quad (6.4)$$

during feeding, and

$$M_h(t + dt) = \min (h_{max}, M_h(t) + dt) \quad (6.5)$$

if another behavior is active. The upper limit imposed on M_h , h_{max} , corresponds to maximum allowed starvation time, and f_{max} corresponds to the maximum duration of feeding (see Section 5.1.3). Thus, the decrease in M_h during feeding is $\frac{h_{max}}{f_{max}}$ units per second. See Table 6.3 in Section 6.1.6 for parameter values.

6.1.4 Motivational state

As described in Section 3.1, it is the motivational state of an animal that give rise to behavior. Consistently with this theory, the motivational state variables of the simulated *Hydra*, are taken as the union of the animal's exteroceptive sensory system, as described in Section 6.1.2, its proprioceptive sensory system, as described in Section 6.1.3, and its current active behavior. The motivational state variables are summarized in Table 6.2 below.

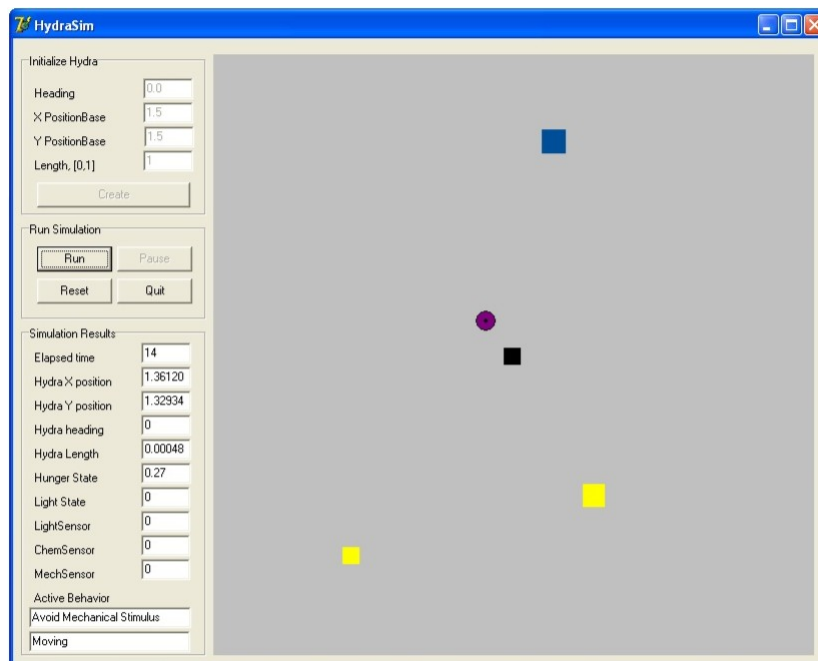
6.1.5 Environment

Simulations of the modeled *Hydra* were carried out in a 2D-environment. A square-shaped arena with side-length S and periodic boundary conditions were used. Two alternatives for the representation of light, mechanical, and chemical stimuli in *Hydra*'s

Table 6.2: Motivational state variables for the simulated *Hydra*.

Variable	Range	Description
M_l	[0, 1]	Reading of light sensor
M_m	[0, 1]	Reading of touch sensor
M_c	[0, 1]	Reading of GSH sensor
M_h	[0, h_{max}]	Nutritional state
M_{beh}	B1,B2,B3,B4	Current active behavior

environment were implemented: (1) Representation of stimuli in terms of objects, where the size, position and intensity of each stimulus is varied according to a pre-defined rule. (2) Sensor readings taken, during simulation, from a file containing stored values for each time step. In the first case, the environment of *Hydra* can be represented, e.g., in terms of stochastic processes, whereas the second alternative may be used for simulating *Hydra* during a pre-defined stimuli setting. A screenshot of the simulation environment can be seen in Fig. 6.2.

**Figure 6.2:** The simulated *Hydra* and environment.

6.1.6 Model parameters

The parameter setting used in this project for the simulated system are specified in Table 6.3.

Table 6.3: Parameter settings for the simulated system.

Parameter	Value	Description
dt	0.5 s	Simulation time step
S	3 l.u.	Side length of simulation arena
l_{max}	$S/30 = 0.1$ l.u.	Length of maximally extended <i>Hydra</i>
r	$l_{max}/2 = 0.05$ l.u.	<i>Hydra</i> radius
a, b	0.5	Rate parameters for movement during contractions
c_{LP}	1.5	Parameter determining the length of a locomotion step
h_{max}	$60.48 \cdot 10^4$ s	Maximum time of starvation (1 week)
f_{max}	1800 s	Maximum duration of feeding response (30 min)

6.2 Behavioral organizer

Consistently with behavior-based modeling, the overall behavior of the simulated *Hydra* emerges from the organization of constituent behavioral units. As mentioned in Chapter 4, many strategies have been suggested for behavior organization within the framework of BBR. Based on the literature study of *Hydra*'s individual, and overall behavior, as described in Section 5.1.3, an arbitration method was chosen. The model used in this project for coordination of *Hydra*'s constituent behaviors is based on the **colony-style architecture** (CSA), a direct descendant from the **subsumption method**¹. The CSA was developed by J. Connell [14], and its operating principles are the following [14, 22]:

- Arbitration method, see Section 4.2.
- Behaviors are arranged in layers, in a priority-based manner.
- Each behavior is associated with an **applicability clause** (AC), and a **transfer function** (TF). The AC determines whether the behavioral output should be active or not, while the TF determines what action the agent would take (typically the motor output), assuming the behavior is active.
- Behavior selection is achieved by switches of any of the following types: (1) Suppression, where the output from one behavior replaces the output from another behavior. (2) Inhibition, where one behavior is prevented to generate any output. (3)

¹A priority-based arbitration method, and one of the pioneering works in BBR, see [8].

Release type, where a higher-priority behavior enables the output of a lower-priority behavior to pass through the switch.

6.2.1 Applicability clause

The AC defines, for each behavior, whether the output from the behavioral unit should be active or not. An AC can be **situation-driven**, i.e. related to a goal state, where the present state (resulting from the agent's action) causes the AC to be either true or false. As an example, consider a robot that turns only when an obstacle appears in front of the robot (and only until the object is no longer in front of it). The AC can also be **event-driven**. An event is characterized by a very brief (point-like) occurrence, whereas situations typically are extended intervals of time [14]. An event-driven AC is of a set/reset type, where a certain event triggers the AC, and some other criteria resets it.

6.2.2 Transfer function

The TF determines the actual action (output) of the behavioral module, given the current motivational state. The output from the TF is only passed out from the behavioral module when the AC is true.

6.3 Constituent behaviors

Using the CSA, modeling each of the constituent behaviors consists of two steps, namely (1) defining an AC, and (2) defining a TF. The architectures of a behavioral module is illustrated in Fig. 6.3.

As described in Section 4.1, there are several different methods by which behaviors can be generated. In the case of *Hydra*, the constituent behaviors are relatively simple, and thus implementation by means of hand-coding was used in most cases. As a complement to hand-coding, biologically inspired methods such as RNNs and EAs were also used. A brief description of these methods is given in Appendices A and B, whereas specific implementation details for the resulting behavioral model of the simulated *Hydra* are stated in Chapter 7.

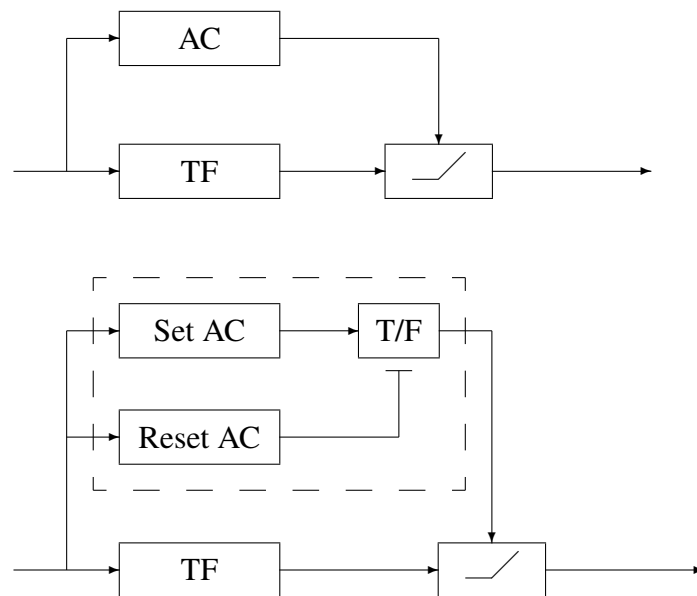


Figure 6.3: Behavioral modules in the CSA. The (motivational) input determines whether the AC generates a true or false signal. A true signal from the AC shuts the gate, and thus passes the signal from the TF to the output of the behavioral module. The top panel shows a situation-driven AC, where the state of the AC depends only on the present input. In the bottom panel an event-driven AC is illustrated. The event-driven AC (within the dashed box) is of set/reset character, i.e. one input configuration sets the AC, while another condition resets it.

Results and discussion

In this chapter the resulting behavioral model of *Hydra* is presented. The process of modeling the overall behavior of *Hydra* was carried out in two steps: (1) Generation of the constituent behaviors involved in *Hydra*'s overall behavior. Using the CSA, this amounts to defining an appropriate AC as well as a TF for each behavior. (2) Generation of the behavioral organizer, i.e. organizing the constituent behaviors into layers, and defining the interacting connections. The two steps were carried out independently of each other. This chapter starts with a presentation of the results concerning each of the constituent behaviors, and concludes with a description of the overall behavior of the simulated *Hydra*.

7.1 Generation of behaviors

The four constituent behaviors of *Hydra*, as described in Section 5.1.3, were generated independently of each other. However, as mentioned in Section 6.1.1, the behavioral output of B1-B3 share some characteristics, namely how *Hydra* should allocate its time between contracted and extended states, as well as in locomotion, depending on the animal's motivational state. Thus, the transfer functions of B1-B3 share the task of evoking a CP or an LP, whichever is applicable, at the appropriate point in time, assuming that the particular behavior is the active one.

It should be noted that each behavior is a self-contained unit, operating only at a local level. Hence, a behavioral unit is not supplied with any information concerning the other behaviors included in the implemented behavior repertoire. It is therefore worth stressing that, while the applicability clause of each behavior determines whether the output from its transfer function should be active or not, the selection of which behavioral output that is finally passed on to the actuators is carried out by the behavioral organizer. The results from the generation of the ACs as well as the TFs for each of the constituent behaviors in

the behavior repertoire of *Hydra* will now be described.

7.1.1 B1: Spontaneous actions

Hydra's spontaneous actions, as described in Section 5.1.3, depend only on its proprioceptive state, and not on any external stimulus.

Applicability clause

By default, spontaneous actions is an applicable behavior (since it does not require any triggering situation, event, or state to evoke activity). Thus, the AC for this behavior, B1, is always true.

Transfer functions

The behavioral output of the spontaneous behavior consists of CPs and LPs. The task of generating these pulses amounts to determining the time of occurrence and the duration of each such pulse, as well as how the CPs and the LPs should interact. To account for the irregularity of the evoked pulses, the CPs and the LPs were implemented as inhomogeneous stochastic pulse trains. The expected values, μ_{con} and μ_{loc} , were taken as functions of the proprioceptive state of *Hydra*, and the rates (in s^{-1}), $R_{con}(\mu_{con})$ and $R_{loc}(\mu_{loc})$, of the evoked pulses were taken as random variables drawn from the exponential distribution. Thus, the points in time for activation of CPs and LPs are modeled by inhomogeneous Poisson spike trains. The expected value of the CP rate was taken as

$$\mu_{con}(M_h) = \mu_{con0} - \frac{\mu_{con0}}{h_{max}} M_h, \quad (7.1)$$

where μ_{con0} is the expected contraction rate of newly fed animals. Here, μ_{con0} was set to $1/360$, since a newly fed animals contracts around 10 times per hour [46]. The contraction rate, $R_{con}(\mu_{con})$ is then exponentially distributed with expected value μ_{con} , i.e. $R_{con} \sim E(1/\mu_{con})$.

The expected value of the LP rate was taken as

$$\mu_{loc}(M_h) = \frac{1}{c_{1,loc}} e^{-x}, \quad (7.2)$$

$$x = \left(\frac{M_h - h_{max}/2}{c_{2,loc}} \right)^2.$$

Here, $c_{1,loc}$ was set to 300, and $c_{2,loc}$ to $2 \cdot 10^5$. The locomotion rate is distributed according to $R_{loc} \sim E(1/\mu_{loc})$. Fig. 7.1 shows the expected rates of CPs and LPs, respectively. The duration of a CP was set to 30 s, as described in [63], and the duration of an LP was

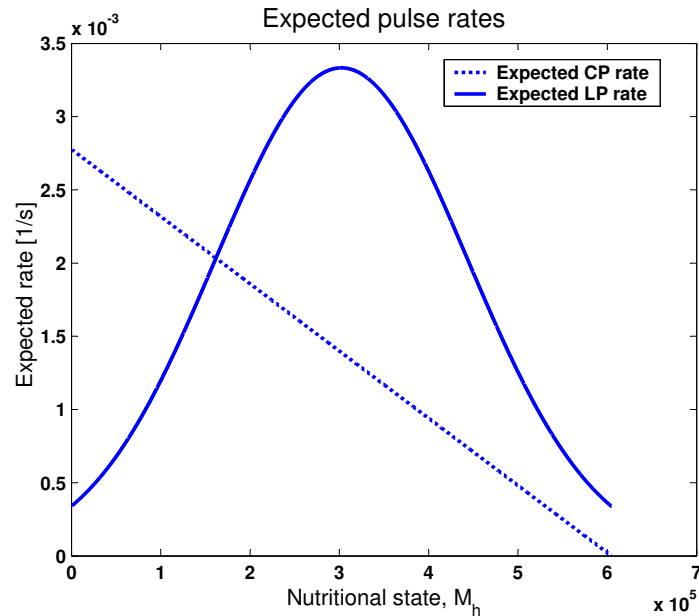


Figure 7.1: Expected rates of CPs and LPs.

set to 60 s. No indication of duration of LP pulses was found in the literature, see Chapter 8 for suggestions of improvements of the duration of an LP. To account for the mutual inhibition of CPs and LPs, whenever there is an overlap in time, only the first occurring pulse type is executed.

7.1.2 B2: Respond to mechanical stimulus

As described in Section 5.1.3, reaction to mechanical stimulus occurs by contraction or locomotion, and shows habituation to repeated stimulus. Test data obtained in [53] shows the response to repeated stimulus presentations, each lasting for 2 s, at 16 s **inter-stimulus intervals** (ISIs). The results from this experiment was used to model the habituation property of B2.

Applicability clause

An event-driven AC was used, and the habituation effect to mechanical stimulus was modeled to agree with the results obtained in [53]. No information concerning the recovery time from habituation was found in the literature, and it was here taken, arbitrarily, to 10000 s. This generates a training data set with inputs consisting of a stimulus with in-

tensity $M_c = 1$ and 2 s duration, presented every 16 s for 3 hours, followed by an absence of the stimulus for 10000 s, and then repeated presentation every 16 s for 4300 s. The outputs of the training data set were reconstructed from [53], and can be seen (later on) in Figs. 7.3 and 7.5.

For this model, two different approaches were used: (1) an RNN, and (2) a habituation model based on leaky integrators. Both models were optimized, parametrically and structurally, by means of an EA. The fitness measure was, in both cases, taken as $f = 1/e$, where e is the **root mean squared error** (RMSE) over the data set during periods of stimulus presentation¹:

$$e = \frac{1}{N} \sqrt{\sum_{i=1}^N (o(i) - y(i))^2}, \quad (7.3)$$

where $o(i)$ represents the i :th measured response, and $y(i)$ the corresponding response from the model. On the occurrence of a stimulus the output from the habituation unit, the **response strength**, s , is related to the probability of a stimulus evoking a response [53]. To ensure a valid range of the response probability, p_{B2} , regardless of habituation model, the following conversion was carried out:

$$p_{B2} = \begin{cases} s, & 0 \leq s \leq 1, \\ 0, & s < 0, \\ 1, & s > 1. \end{cases} \quad (7.4)$$

Activation of the AC occurs if $M_c > 0$ and $p_{B2} > X$, where X is a random variable drawn from the uniform distribution, $X \sim U(0, 1)$. In general, the latency depends on stimulus strength, as described in Section 3.2.1. However, no such information was found in the literature for *Hydra*'s response to mechanical stimulus, therefore it was set to be constant. Reset of the AC occurs at completion of the response, which may be a CP or an LP, as described below. The results from each test case of evolving a habituation unit will now be described.

Test A: Evolving an RNN to represent habituation

As a first experiment it was tested whether an RNN, parametrically and structurally optimized by means of an EA to fit experimental data obtained in [53], could represent the habituation to mechanical stimulus. A continuous time RNN was used. After applying Euler's method for numerical integration of the network equations, the dynamics of neuron i in the network is governed by the following equation (see also Appendix B):

$$y_i(t + \Delta t) = y_i(t) + \frac{\Delta t}{\tau_i} \left[-y_i(t) + \sigma \left(b_i + \sum_{j=1}^n w_{ij} y_j(t) + \sum_{j=1}^m w_{ij}^I I_j(t) \right) \right], \quad (7.5)$$

¹Note that no contribution to the RMSE occurs during the recovery period.

where b is the bias term, τ the time constant, I the input signal(s), and w and w^I are the synaptic weights from other neurons and input signals, respectively. The integration time step, Δt , was set to 0.2 s. The simulation time step (time between two consecutive input signals) is 2 s, which gives 10 integration steps between data points used in the calculation of Eq. 7.3. A background to the RNNs used here is given in Appendix B. The RNN was evolved using an EA with the following properties (see Appendix A for an explanation of the terms): explicit encoding, elitism, no crossover, tournament selection, and structural as well as parametrical mutations. After an investigation of various mutation operators, six different operators were used, as described below. The EA properties are summarized in Table 7.1.

- m_1 - *Creep mutation*: The value of the gene is updated according to $w^{new} = w^{old}(1 - 2rc + c)$, where $r \sim U[0, 1]$ and c is the *creep rate*. Since this mutation may generate values outside the allowed range, the gene was scaled into its proper interval using $w \rightarrow w^{max}$ if $w > w^{max}$, and $w \rightarrow w^{min}$ if $w < w^{min}$.
- m_2 - *Full-range mutation*: The gene is given a new, random value within the allowed parameter interval.
- m_3 - *Add connection mutation*: A connection between two units (either between two neurons, or between an input signal and a neuron) with a randomly chosen synaptic weight (within the allowed range), is added.
- m_4 - *Remove connection mutation*: Removal of a connection between two units.
- m_5 - *Add neuron mutation*: One neuron is added to the RNN (at a randomly chosen location). To avoid a macromutation, i.e. a mutation that alters the performance of the resulting individual in a significant way, the neuron is added with all weights set to zero (and with randomly selected bias and time constant). Thus, only the *possibility* of new connections is established as a result of this mutation operator.
- m_6 - *Remove neuron mutation*: Removal of one randomly chosen neuron and all its incoming and outgoing connections.

In Fig. 7.2, the structure of the best evolved RNN is shown. The network consists of three neurons and, as can be seen from the network parameters in Eq. 7.6, all connections but one input signal connection were established. The evolved RNN performs adequately on the test data, as can be seen in Fig. 7.3. However, its generalization ability to other ISIs and stimulus intensities is poor, an example of which is also shown in Fig. 7.3. In general, it should be possible to improve this ability by including such stimulus properties in the training data set. However, no such data was found in the literature, and it was decided to use another approach rather than creating a new, somewhat arbitrary data set. As a result, a solution based on a model of habituation was investigated, which will be described next.

Table 7.1: EA properties and settings for the best evolved RNN-based habituation unit. For all three mutation cases, if a mutation occurs, one out of two mutation operators is selected. If, e.g., a connection mutation occurs, there is a 50% chance of a connection being added (m_3), and a 50% chance of a connection being removed (m_4).

Population size	50
Initial RNN size	2
Crossover	Not used
Tournament size	5
p_{tour}	0.65
$p_{mut1,2}$	0.10
$p_{mut3,4}$	0.15
$p_{mut5,6}$	0.10
Creep rate, c	2
Range, weights and biases	$[-5, 5]$
Range, time constants [s]	$[10, 2000]$

$$w = \begin{pmatrix} 0.63 & -5.00 & 5.00 \\ -0.39 & 3.17 & -5.00 \\ 2.30 & -5.00 & 3.51 \end{pmatrix}$$

$$w^I = (0.51 \quad 0.00 \quad 3.71)$$

$$b = (-0.75 \quad -1.17 \quad -3.72)$$

$$\tau = (1107.21 \quad 95.50 \quad 509.15) \tag{7.6}$$

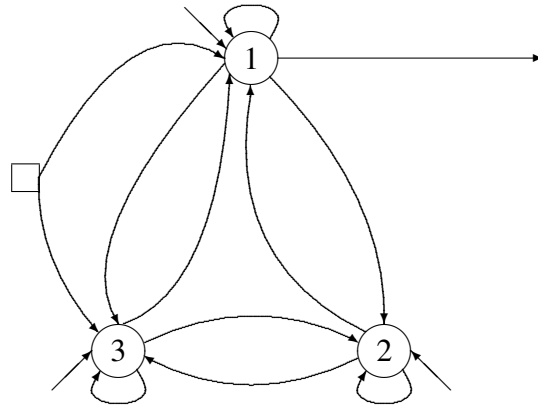


Figure 7.2: Evolved RNN habituation model.

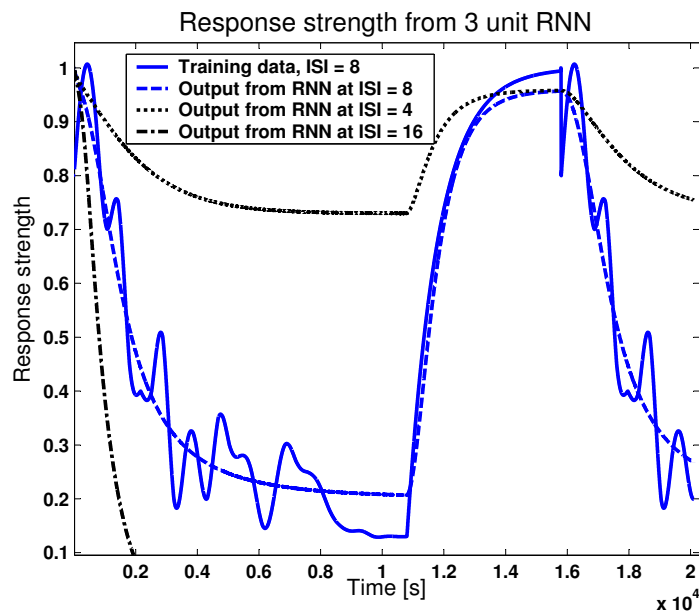


Figure 7.3: Response strength for best evolved RNN-based habituation model. ISIs are presented here in time-step units (1 time-step equals 2s, as described above). While the RNN performs well on training data (ISI = 8), note that the recovery from habituation is not adequate for stimulus of ISI = 4.

Test B: Evolving a habituation model based on leaky integrators

One way of modeling habituation is to use cascaded leaky integrators, as proposed e.g. in [62], and illustrated in Fig. 7.4. With the stimulus input as Y_1 , the equations for unit j in a cascade of n integrators become

$$V_j(k+1) = a_j V_j(k) + b_j Y_j(k), \quad (7.7)$$

$$Y_{j+1}(k) = \begin{cases} Y_j(k) - V_j(k), & Y_{j+1}(k) > T_j \\ 0, & \text{otherwise.} \end{cases} \quad (7.8)$$

Here all thresholds, T_j , were set to 0. The response strength, s , is taken as the output from the last (N :th) unit, Y_{N+1} , on the occurrence of a stimulus, as proposed in [62]. An EA was used to evolve the size (N) and parameters $\{a_1, a_2, \dots, a_N, b_1, b_2, \dots, b_N\}$ of a habituation model on this form to fit the data previously described. For the EA, chromosomal real number encoding was used, with genes taking values in $[0, 1]$. Two-point crossover² and tournament selection was used, as well as two mutation operators, m_1 and m_2 . Carrying out parametrical mutations, m_1 assigns a new (random) value to a gene. Mutation operator m_2 adds or removes, with equal probability, two genes (corresponding to one integrator unit), to an individual. The EA properties, along with the performance of the obtained model, are shown in Table 7.2 and Fig. 7.5, respectively.

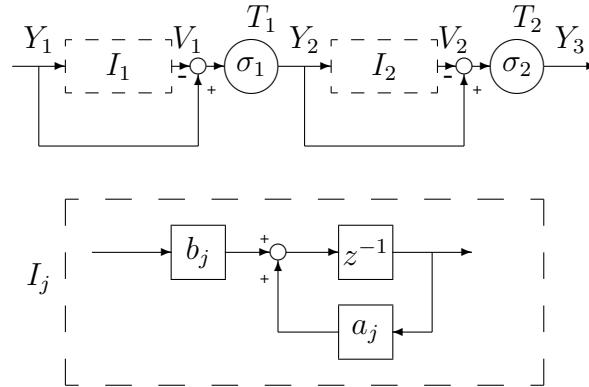


Figure 7.4: Habituation model based on leaky integrators. The top panel shows two cascaded units in compact form. Here, σ_j is a threshold unit with parameter T_j . The bottom panel shows the contents of each integration unit, I_j .

²Since each integrator unit is represented by two genes, a valid individual consist of an even number of genes. To avoid generating individuals with odd number of genes, the crossover points were restricted to every two genes.

Table 7.2: EA properties for the best evolved cascaded integrators habituation unit.

Population size	100
Initial model size	5 units
Tournament size	10
p_{cross}	0.5
p_{tour}	0.70
p_{mut1}	0.10
p_{mut2}	0.05
Parameter range (a,b)	[0, 1]

The best model obtained consists of 10 units, the parameter values of which are given below.

$$a = (0.001 \ 0.001 \ 0.001 \ 0.001 \ 0.001 \ 0.001 \ 0.002 \ 0.001 \ 0.998 \ 0.003)$$

$$b = (0.124 \ 0.010 \ 0.105 \ 0.012 \ 0.711 \ 0.204 \ 0.107 \ 0.549 \ 0.004 \ 0.010)$$

Transfer function

The transfer function is initialized when the AC is set, and its task is to generate a CP or an LP. One problem with determining whether locomotion or contraction is the adequate response is that only qualitative descriptions of this relationship were found in the literature [28, 45, 46]. These descriptions implies that there is a relative increase in locomotion responses in starved animals. Here, the following relationship was used to determine the response type

$$\begin{aligned} f_{con} &= a_{con}, \\ f_{loc} &= a_{loc}M_h, \end{aligned} \quad (7.9)$$

for $0 < M_h < h_{max}$. If $M_h = h_{max}$, neither a CP nor an LP is evoked. Here, a_{con} was set to 300 and a_{loc} to 0.001. The probability of locomotion and contraction response, respectively, was taken as

$$\begin{aligned} p_{con} &= \frac{f_{con}}{f_{con} + f_{loc}}, \\ p_{loc} &= \frac{f_{loc}}{f_{con} + f_{loc}}. \end{aligned} \quad (7.10)$$

The constant latency was set to 2 s, the duration of a CP to 10 s, and the duration of an LP to 60 s. Fig. 7.6 shows an example of the output from the transfer function during a contraction response to mechanical stimulus. The AC, being of set/reset character, causes the behavioral output to be active between the time of set and reset.

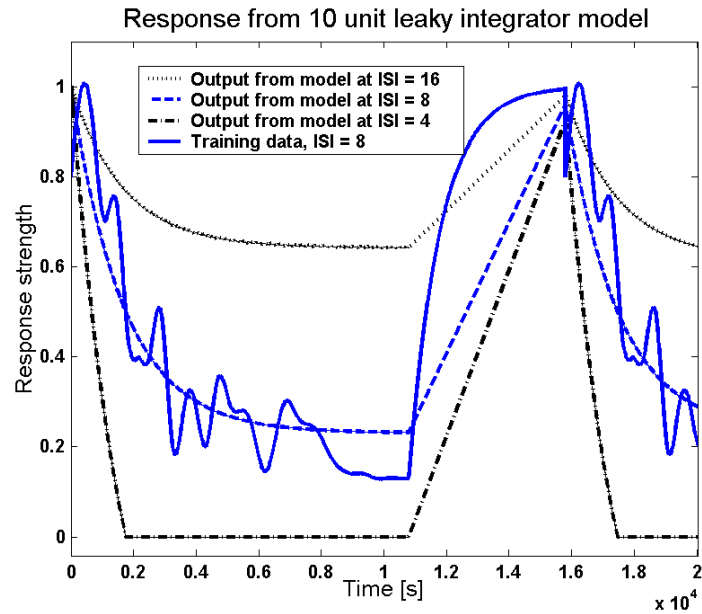


Figure 7.5: Response strength for best evolved integrator habituation model. ISIs in time-steps.

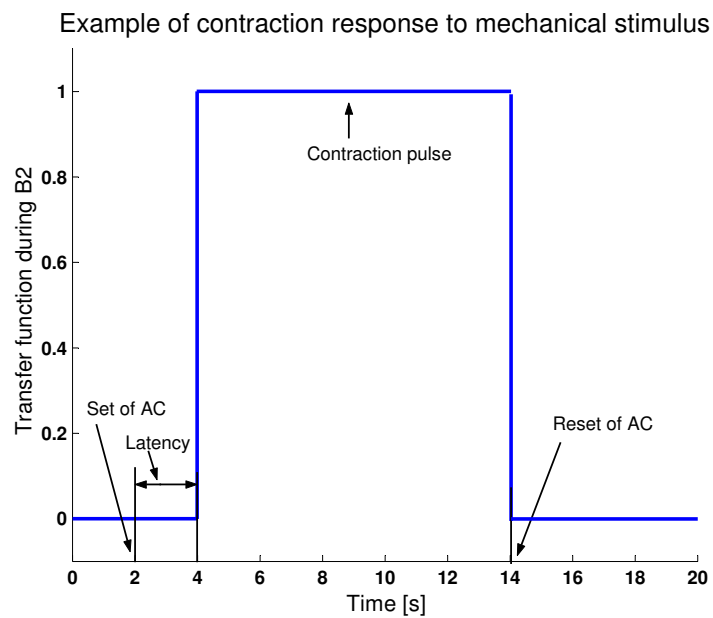


Figure 7.6: Example of output from transfer function during response to mechanical stimulus.

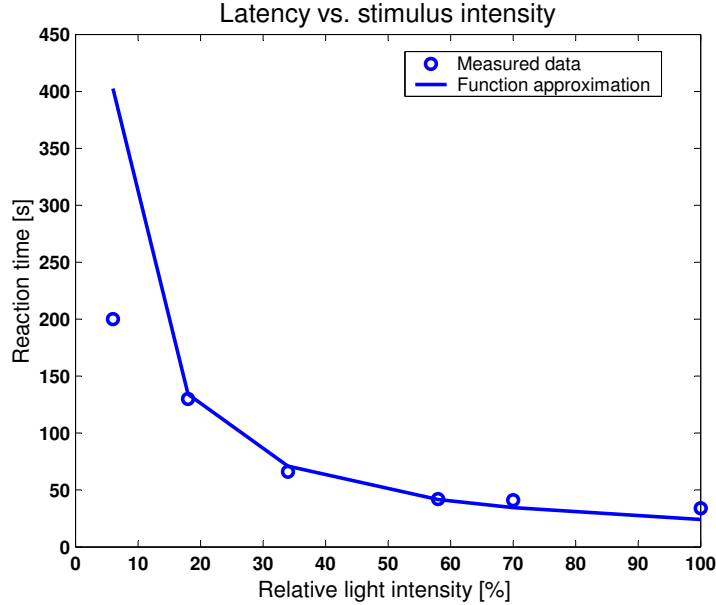


Figure 7.7: Latency of response to light stimulus. In the function fitting, the measured response to 6% relative stimulus intensity was considered an outlier.

7.1.3 B3: Respond to light stimulus

Similar to B2, response occurs by contraction or by locomotion. As discussed in Section 5.1.3, upon exposure to light, there is an immediate inhibition of any ongoing contraction. Following a latency which depend on stimulus intensity, a response is evoked. There is no habituation to repeated stimulus.

Applicability clause

As with B2, an event-driven AC was used. Activation of the AC occurs if $M_l > T_l$, where T_l was set to 0.2. The latency was modeled to fit the data obtained in [59]. Least square function fitting was used, and the latency obtained was

$$L_l = \frac{24.15}{M_l}. \quad (7.11)$$

Fig. 7.7 illustrates the result obtained. Note that the measured data is a function of relative stimulus intensity in percent, whereas readings from the light sensor in the modeled *Hydra* are taken in $[0, 1]$. The activation threshold, T_l , also ensures that division by zero does not occur in the calculations of L_l .

Transfer function

The TF of B3 share the same task as the one of B2, namely to evoke a CP or an LP, whichever is applicable. Eqs. 7.9 and 7.10 were applied to determine response type also in this case. Similar to B2, the time for the applicable response was set to $t + L_l$, where t is the time for activation of the AC, and L_l is the latency as determined above. Also in this case, the duration of a CP was set to 10 s, and the duration of an LP to 60 s.

7.1.4 B4: Feeding

The feeding behavior of *Hydra* is activated by the presence of GSH and, as discussed in Section 5.1.3, both activation threshold and duration of the response depend on the nutritional state of the animal. Throughout the feeding response, there is a total inhibition of other behaviors [52].

Applicability clause

Also for B4, an event-driven AC was used. The activation threshold and duration of response were determined with respect to measured data from experimental studies of the animal.

The threshold for activation of the feeding behavior in *Hydra* depends on time of starvation. In [31], a quantitative study of this relationship was conducted. The results indicates an exponential decrease in the feeding threshold, T_f , for an increase in starvation time. In a least square function fitting to the measured data, Eq. 7.12 was obtained. The results are shown in Fig. 7.8.

$$T_f = \frac{1.29 \cdot 10^4}{M_h^{1.95}}, \quad (7.12)$$

where M_h is increased by 1 per second while the animal is not feeding, see Eq. 6.5.

The duration of B4 have been studied in [33], and indicates a gradual recovery of the ability to respond following a prior feeding response. The following relation was determined for the ability of feeding duration

$$D_f = 1800 (1 - e^{-0.2x}), \quad (7.13)$$

where $x = \frac{M_h}{3600}$. Fig. 7.9 shows the gradual recovery.

Activation of the AC occurs if $M_c > T_f$, while the AC is reset if $t_{B4} > D_f$, or if $M_c = 0$, where t_{B4} is the time for which B4 has been active and D_f is the ability of duration of the feeding response at the time of activation of B4. Thus, feeding occurs until repletion or until the presence of food is no longer detected by the sensory system.

Transfer function

Feeding in *Hydra* consists of a sequence of actions, and involves e.g. tentacle movement. However, in the simplified model of *Hydra* used in this project it does not seem motivated to model any movement of the animal during B4. Thus, the transfer function was omitted in the sense that no physical state variable is affected by the feeding behavior.

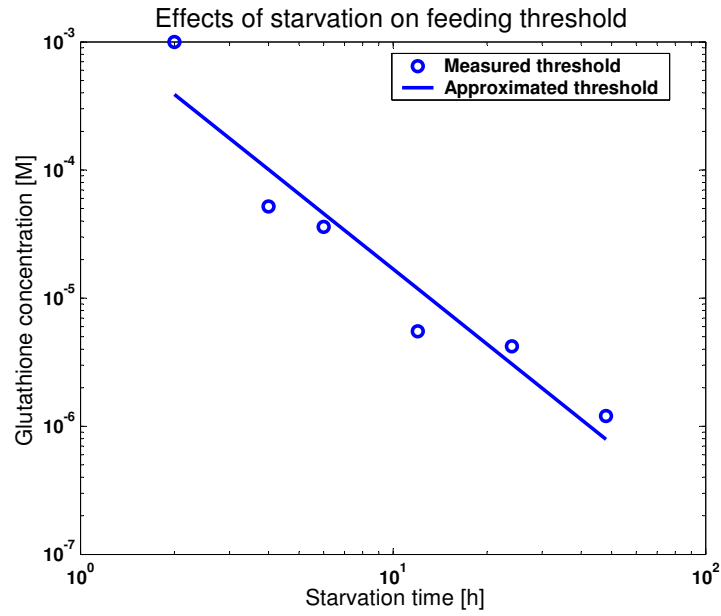


Figure 7.8: Effects of starvation on threshold for evoking the feeding response in *Hydra*.

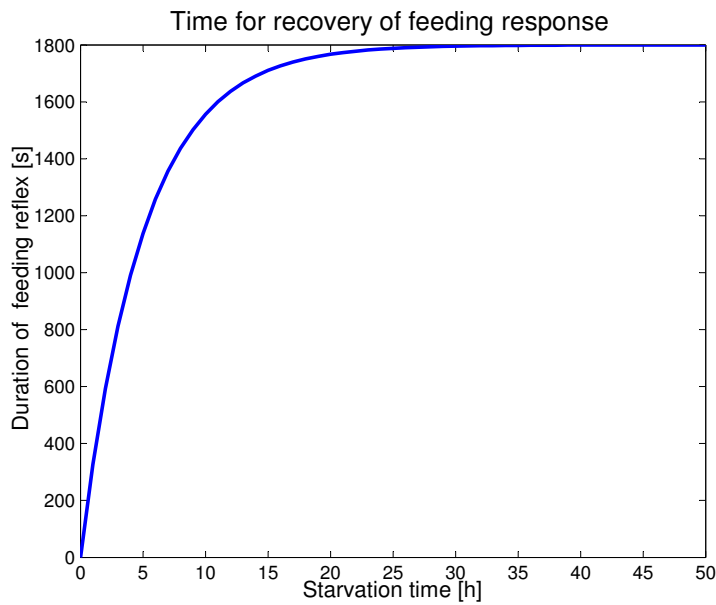


Figure 7.9: Effects of starvation on duration of the feeding response in *Hydra*.

7.2 Generation of behavioral organizer

As previously described, a priority-based interaction of the behaviors in *Hydra* is suggested in the literature. Thus, behavioral organization using the CSA is straight-forward, and can be accomplished by organizing the behaviors in layers with respect to priorities, using switches of suppression type for the behavioral inter-connections. Fig. 7.10 shows the resulting behavioral model.

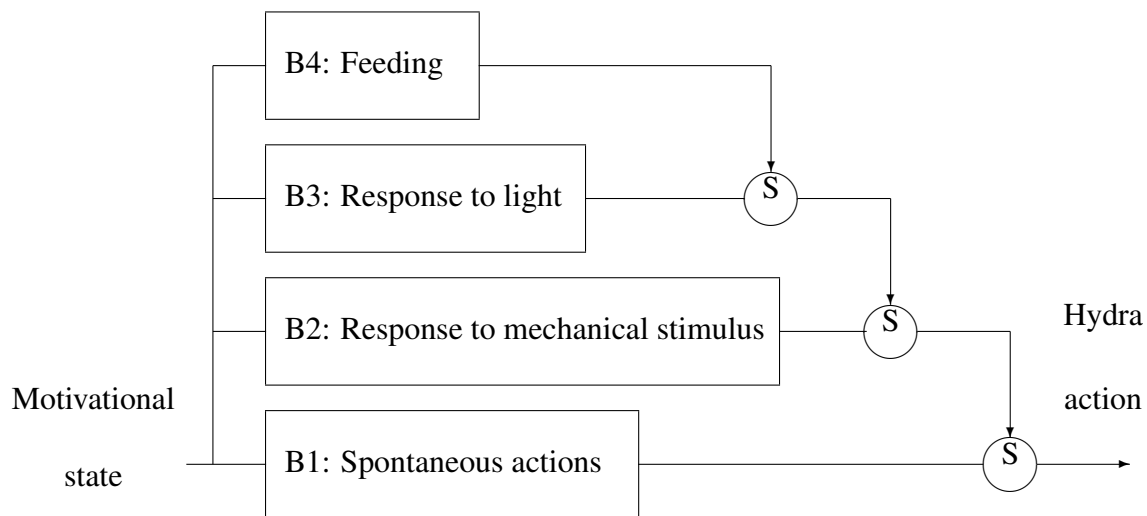


Figure 7.10: Structure of colony-style behavioral organizer for the behaviors of *Hydra*. Here, *S* represents switches of suppression-type.

In behavior selection situations, in biological organisms as well as in artificial systems, it is common that some trade-off is necessary [40]. Thus, it would be desirable to investigate to what extent the suggestion of priority-based behavior organization in *Hydra* holds. However, not enough results from experiments on the physical animal were found in the literature to enable such an investigation at this point. In Chapter 8, recommendations for further experiments with respect to this are discussed. To validate the model in other aspects, a set of simulation experiments were carried out, the results of which will now be described.

7.2.1 Experiment A: Spontaneous locomotion patterns

Wagner recorded the position of *Hydra*, when the animal was left undisturbed for several days. His results suggest that the spontaneous locomotion behavior of *Hydra* is much like

a random walk [66]. Jennings used the results from Wagner's experiments in his detailed description of spontaneous activities in *Hydra* [28].

To investigate the spontaneous movement patterns of the modeled *Hydra*, simulations of the animal in the absence of any external stimulus during three days (72 hours) were carried out. The foot position, (x_f, y_f) , of the simulated *Hydra* was recorded during this time. Evident from the implementation details of LPs, as stated in Section 6.1.1, the results agrees with Wagner's study in the sense that *Hydra* moves about in a random walk. Additional test data would allow for an expansion of this experiment, as is discussed in Chapter 8. In the modeled *Hydra*, all locomotion steps are equally large, as discussed in Section 6.1.1. This differs from Wagner's observations, in which the *Hydra* shows locomotion steps of varying lengths. This limitation of the modeled *Hydra* is due to the simplifications made in the representation of the animal's body, and a possible improvement is discussed in Chapter 8. The result from one realization of this simulation experiment is shown in Fig. 7.11.

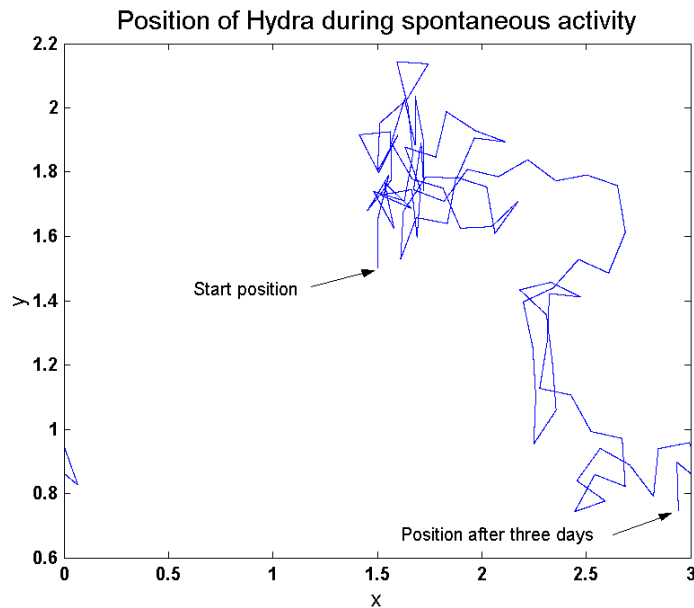


Figure 7.11: Position of *Hydra* during three days of spontaneous activity.

7.2.2 Experiment B: Effect of starvation on contraction frequency

In [46], Passano and McCullough studied the effect of starvation on the contraction frequency of *Hydra*. The average contraction rate of animals fed daily, as well as every third

day, were recorded three times per day. In order to investigate this property of the modeled *Hydra*, simulations of the animal during similar feeding conditions were carried out. First, *Hydra* was fed during 6.5 minutes every 24 hours, and secondly, the animal was fed for the same period of time every 72 hours. In each case ten simulation runs were conducted, from which the mean and standard deviation of the contraction frequency were calculated. The results from simulations, as well as from the experimental study in [46], are presented in Table 7.3.

The contraction frequency of animals that are fed every third day is described in [46] as being reduced during the days following feeding. The decrease in frequency, however, is not significant for all recordings. The results from simulations shows a more rapid decrease than the experimental results, but its qualitative properties agrees with those shown by the physical *Hydra*: Animals fed daily contracts at a higher frequency than animals fed less often, even immediately after feeding. Also, there is a decrease in the contraction frequency during the days following feeding in animals fed every three days.

Table 7.3: Contractions per hour (mean and standard deviation) in *Hydra* fed daily and every third day, respectively.

	Experimental data, evening count	Simulation results
Fed daily	7.2 ± 0.5	7.98 ± 0.25
Fed every third day		
First day	5.1 ± 0.4	6.15 ± 0.44
Second day	4.9 ± 0.4	4.36 ± 0.63
Third day	4.8 ± 0.4	3.69 ± 0.14

Summary and conclusions

In this thesis work methods from BBR have been used to model the behavior of a simple biological organism. Literature studies were conducted concerning suitable model organisms, as well as methods for behavior generation and organization. Based on the literature surveys, the cnidarian *Hydra* was selected as the model organism for this project, and the CSA was selected as the framework for modeling the behavior of *Hydra*. The two main reasons for choosing *Hydra* as the model organism are the animal's complexity level, and the number of measurement data available from prior ethological experiments conducted on the animal. The reasons for choosing the CSA are that it supports implementation of priority based behaviors (as is the case with *Hydra*'s behaviors according to the literature), as well as allows for modeling the reflex properties found in *Hydra* such as latency, habituation, and duration of response.

The general conclusion of the work conducted in this project is that it is possible to model several of the behavioral properties of *Hydra* using the CSA as a framework. The modeled latency of *Hydra*'s response to light stimulus, the habituation effect of its response to mechanical stimulus, the duration of a CP and the feeding response, respectively, as well as the variable feeding threshold are consistent with experimental data. Specifically, it was found that a habituation model based on cascaded leaky integrators can represent the habituation properties of the animal. Simulations of the overall behavior of the animal shows that the spatial movement pattern of undisturbed *Hydra*, as well as the effect of starvation on contraction frequency, agrees with results from experiments conducted on the physical animal. In order to obtain conclusive results concerning the overall behavior of *Hydra*, however, some improvements remain.

Several future contributions would be of interest: Expanding the model to include the morphology of *Hydra* would allow for more accurate modeling, e.g. of the duration of an LP, as well as the distance of movement, in the case of locomotion by means of somersaulting, as discussed in Section 7.2.1. Generation of more test data for validation purposes would also be of interest. For example, a recording not only of the spatial

patterns of movement, but also of the time of occurrence for each movement, would allow for an expansion of the simulations carried out in Section 7.2.1 to include also the frequency of locomotion. If such experiments were also to include a recording of the contraction frequency, more insight of the interaction between CPs and LPs could be gained. Furthermore, experiments concerning the adaption to ambient light conditions would be of interest, since such experimental data could be used to model an additional internal process of the animal. Also of interest for further work is an investigation of the extent to which priority-based behavior organization in *Hydra* holds. For this purpose, experiments on how the animal reacts to light and mechanical stimuli during feeding, not only initially, but also after some time has passed, are recommended.

Bibliography

- [1] ARKIN, R., *Behavior-Based Robotics*. Cambridge, MA: The MIT Press, 1998.
- [2] ARKIN, R., KAHLED, A., WEITZENFELD, A., and CERVANTES-PEREZ, F., “Behavioral models of the praying mantis as a basis for robotic behavior”, *Journal of Robotics and Autonomous Systems*, vol. 32, no. 1, pp. 39–60, 2000.
- [3] ARKIN, R. C., “Reactive robotic systems”, in *Handbook of Brain Theory and Neural Networks* (ARBIB, M., ed.), pp. 793–796, Cambridge, MA: MIT Press, 1995.
- [4] BEKEY, G., *Autonomous Robots: From Biological Inspiration to Implementation and Control*. Cambridge, MA: The MIT Press, 2005.
- [5] BERGREN, C., *Anatomy of a Robot*. New York, NY: McGraw-Hill, 2003.
- [6] BEST, J. and RUBINSTEIN, I., “Maze learning and associated behavior in planaria”, *Journal of Comparative and Physiological Psychology*, vol. 55, pp. 560–566, 1962.
- [7] BIOMEDIA ASSOCIATES, “<http://www.ebiomedia.com>”, Oct. 13th, 2006.
- [8] BROOKS, R., “A robust layered control system for a mobile robot”, *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 1, pp. 14–23, 1986.
- [9] BROOKS, R. A., “Intelligence without reason”, in *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)* (MYOPOULOS, J. and REITER, R., eds.), (Sydney, Australia), pp. 569–595, Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1991.
- [10] BUASON, G., BERGFELDT, N., and ZIEMKE, T., “Brains, bodies, and beyond: Competitive co-evolution of robot controllers, morphologies and environments”, *Genetic Programming and Evolvable Machines*, vol. 6, no. 1, pp. 25–51, 2005.

-
- [11] BÄCK, T., FOGEL, D., and MICHALEWICZ, Z., eds., *Handbook of Evolutionary Computation*. Bristol: Institute of Physics Publishing and Oxford University Press, 1997.
- [12] CHIEL, H. and BEER, R., “The brain has a body: Adaptive behavior emerges from interactions of nervous system, body and environment”, *Trends in Neurosciences*, vol. 20, no. 12, pp. 553–557, 1997.
- [13] CLARK, W. and GRUNSTEIN, M., *Are We Hardwired? The Roles of Genes in Human Behavior*. Oxford University Press, 2000.
- [14] CONNELL, J., “A colony architecture for an artificial creature”, Tech. Rep., 1151, MIT Artificial Intelligence Laboratory, June 1989.
- [15] DAWKINS, R., *The Selfish Gene*. Oxford: Oxford University Press, 2nd ed., 1989.
- [16] DAWKINS, R., *Climbing Mount Improbable*. New York, N.Y.: W.W. Norton and Company, 1996.
- [17] DAWKINS, R., *The Blind Watchmaker*. London: Penguin Books, 2000.
- [18] EIBL-EIBESFELDT, I., *Ethology: The Biology of Behavior*. New York, NY: Holt, Rinehart and Winston, 2nd ed., 1975.
- [19] GIBSON, J., *The Ecological Approach to Visual Perception*. Boston, MA: Houghton Mifflin, 1979.
- [20] GROOME, D., *An Introduction to Cognitive Psychology : Processes and Disorders*. London: Psychology Press, 2nd ed., 1999.
- [21] HAEFNER, J., *Modeling Biological Systems: Principles and Applications*. New York, NY: Chapman & Hall, 1996.
- [22] HARPER, C. and WINFIELD, A., “A methodology for provably stable behaviour-based intelligent control”, *Robotics and Autonomous Systems*, vol. 54, pp. 52–73, 2006.
- [23] HAYKIN, S., *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ: Prentice Hall, 2nd ed., 1999.
- [24] HICKMAN, C., ROBERTS, L., LARSON, A., I’ANSON, H., and EISENHOUR, D., *Integrated Principles of Zoology*. New York, N.Y.: McCraw-Hill, 13th ed., 2006.
- [25] HOLLAND, J., *Adaptation in natural and artificial systems*. Cambridge, MA: MIT Press, 1992.

- [26] HOLYOAK, K. and MORRISON, R., eds., *The Cambridge Handbook of Thinking and Reasoning*. New York, N.Y.: Cambridge University Press, 2005.
- [27] JENNINGS, H., *Contributions to the Study of the Behavior of Lower Organisms*. Washington: Carnegie Institution of Washington, 1904.
- [28] JENNINGS, H., *Behavior of the Lower Organisms*. Bloomington: Indiana University Press, 1962.
- [29] KASS-SIMON, G., PANNACTIONE, A., and PIEROBON, P., “GABA and glutamate receptors are involved in modulating pacemaker activity in Hydra.”, *Comparative Biochemistry and Physiology A*, vol. 136, pp. 329–342, 2003.
- [30] KOCH, C. and SEGEV, I., eds., *Methods in Neuronal Modeling, from Ions to Networks*. Cambridge, MA: MIT Press, 2nd ed., 1998.
- [31] KOIZUMI, O. and MAEDA, N., “Rise of feeding threshold in satiated Hydra”, *Journal of Comparative Physiology*, vol. 142, pp. 75–80, 1981.
- [32] LENHOFF, H., “Behavior, hormones, and Hydra”, *Science*, vol. 161, pp. 434–442, 1968.
- [33] LENHOFF, H. and LOOMIS, W., eds., *The Biology of Hydra and Some Other Coelenterates*. Miami, FL: University of Miami Press, 1961.
- [34] MAES, P., “How to do the right thing”, *Connection Science Journal, Special Issue on Hybrid Systems*, vol. 1, no. 3, pp. 291–323, 1989.
- [35] MANGEL, M. and CLARK, C., *Dynamic Modeling in Behavioral Ecology*. Princeton, NJ: Princeton University Press, 1988.
- [36] MANNING, A. and DAWKINS, M., *An Introduction to Animal Behaviour*. Cambridge: Cambridge University Press, 1998.
- [37] MATARIĆ, M., “Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior”, *Trends in Cognitive Science*, vol. 2, no. 3, pp. 82–87, 1998.
- [38] MAYNARD SMITH, J., *Evolution and the Theory of Games*. Cambridge, MA: Cambridge University Press, 1982.
- [39] MCFARLAND, D., *Animal behavior: Psychobiology, Ethology and Evolution*. Prentice Hall, 3rd ed., 1998.
- [40] MCFARLAND, D. and BÖSSER, T., *Intelligent Behavior in Animals and Robots*. MIT Press, 1993.

-
- [41] MITCHELL, M., *An Introduction to Genetic Algorithms*. Cambridge, MA: The MIT Press, 1996.
- [42] NOLFI, S. and FLOREANO, D., *Evolutionary Robotics*. Cambridge, MA: The MIT Press, 2000.
- [43] NORDAHL, M., 2006. Private communication.
- [44] NORTHROP, R., *Introduction to Dynamic Modeling of Neuro-Sensory Systems*. CRC Press, 2001.
- [45] PASSANO, L. and MCCULLOUGH, C., “Pacemaker hierarchies controlling the behaviour of Hydras”, *Nature*, vol. 199, pp. 1174–1175, 1963.
- [46] PASSANO, L. and MCCULLOUGH, C., “Co-ordinating systems and behavior in Hydra, I. pacemaker system of the periodic contractions”, *Journal of Experimental Biology*, vol. 41, pp. 643–664, 1964.
- [47] PETTERSSON, J., *Generation and Organization of Behaviors for Autonomous Robots*. PhD thesis, Chalmers University of Technology, Göteborg, 2006.
- [48] PIRJANIAN, P., “An overview of system architectures for action selection in mobile robots”, Tech. Rep., Laboratory of Image Analysis, Aalborg University, Aalborg, May 1997.
- [49] PIRJANIAN, P., “Behavior coordination mechanisms - state-of-the-art”, Tech. Rep., IRIS-99-375, Institute of Robotics and Intelligent Systems, School of Engineering, University of Southern California, Los Angeles, CA, October 1999.
- [50] RIDDLE, D., BLUMENTHAL, T., MEYER, B., and PRIESS, J., *C. Elegans II*. New York: Cold Spring Harbor Laboratory Press, 1997.
- [51] RUGGIERIA, R., PIEROBONB, P., and KASS-SIMON, G., “Pacemaker activity in Hydra is modulated by glycine receptor ligands”, *Comparative Biochemistry and Physiology A*, vol. 138, pp. 193–202, 2004.
- [52] RUSHFORTH, N., “Modification in temporal patterning of pacemaker activity in Hydra by sensory input”, *The Journal of General Physiology*, vol. 50, p. 2484, 1967.
- [53] RUSHFORTH, N., BURNETT, A., and MAYNARD, R., “Behavior in Hydra: Contraction responses of Hydra Pirardi to mechanical and light stimulation”, *Science*, vol. 139, pp. 760–761, 1963.
- [54] RUSHFORTH, N., KROHN, I., and BROWN, L., “Behavior in Hydra: Inhibition of the contraction responses of Hydra Pirardi”, *Science*, vol. 145, pp. 602–604, 1964.

- [55] RUSSELL, S. and NORVIG, P., *Artificial Intelligence: A Modern Approach*. Upper Saddle River, N.J: Prentice-Hall, 2nd ed., 2003.
- [56] SAFFIOTTI, A., “The uses of fuzzy logic in autonomous robot navigation”, *Soft Computing*, vol. 1, no. 4, pp. 180–197, 1997.
- [57] SEVERIN, E., *Robot Companions: MentorBots and Beyond*. New York, NY: McGraw-Hill, 2003.
- [58] SIEGWART, R. and NOURBAKHSH, I., *Introduction to Autonomous Mobile Robots*. Cambridge, MA: MIT Press, 2004.
- [59] SINGER, R., RUSHFORTH, N., and BURNETT, A., “The photodynamic action of light on Hydra”, *Journal of Experimental Zoology*, vol. 54, no. 2, pp. 169–173, 1963.
- [60] SLATER, P., *Essentials of Animal Behaviour*. Cambridge: Cambridge University Press, 1999.
- [61] STADDON, J., *Adaptive Behavior and Learning*. Cambridge, MA: Cambridge University Press, 1983.
- [62] STADDON, J., *Adaptive Dynamics*. Cambridge, MA: MIT Press, 2001.
- [63] TADDEI-FERRETTI, C. and MUSIO, C., “The neural net of Hydra and the modulation of its periodic activity”, in *Foundations and Tools for Neural Modeling, International Work-Conference on Artificial and Natural Neural Networks, IWANN '99* (MIRA, J. and SÁNCHEZ-ANDRÉS, J. V., eds.), vol. 1606 of *Lecture Notes in Computer Science*, pp. 123–137, Springer, 1999.
- [64] TADDEI-FERRETTI, C. and MUSIO, C., “Photobehavior of Hydra (Cnidaria, Hydrozoa) and correlated mechanisms: A case of extraocular photosensitivity”, *Journal of Photocmemistry and Photobiology*, 2000.
- [65] TOATES, F. M., *Animal Behavior: A Systems Approach*. New York, NY: John Wiley & Sons, 1980.
- [66] WAGNER, G., “On some movements and reactions of Hydra”, *Quarterly Journal of Microscopical Science*, vol. 48, pp. 585–622, 1905.
- [67] WAHDE, M., *An Introduction to Adaptive Algorithms and Intelligent Machines*. Göteborg: Chalmers Reproservice, 4th ed., 2005.
- [68] WAHDE, M. and SANDHOLT, H., “Evolution of complex behaviors on an autonomous robot”, in *Proceedings of the 7th UK Mechatronics Forum International Conference*, (Oxford), Pergamon Press, 2000.

- [69] WATSON, J., *Behaviorism*. New York, N.Y.: Norton, 1925.
- [70] WEBB, B., “Can robots make good models of biological behavior?”, *Behavioral and Brain Sciences*, vol. 24, pp. 1033–1050, 2001.
- [71] WOLFF, K., *Generation and Optimization of Motor Behaviors in Real and Simulated Robots*. PhD thesis, Chalmers University of Technology, Göteborg, 2006.
- [72] XUE, X. and HENDERSON, T., “Video based animal behavior analysis”, Tech. Rep., UUCS-06-006, School of Computing, University of Utah, Salt Lake City, UT, 2006.

Evolutionary algorithms

In this appendix, a brief introduction to EAs is given. Numerous literature is available on the subject, for a more thorough review see e.g. [11, 41, 67].

EAs is an umbrella term for search algorithms inspired by evolution in nature. Differing somewhat in representation and implementation details, versions of EAs include genetic algorithms (GAs), evolution strategies (ES), and genetic programming (GP) [67]. Suited for problems with large search space and many local optima, EAs have been used in a wide variety of application areas, such as function optimization, scheduling, construction of ANNs, system identification, and time series prediction. An EA operates on a **population**, which consists of several **individuals**. An individual can be seen as a candidate solution for the task at hand, and its information is contained in one or more **chromosomes**, forming the **genome** of the individual. Each chromosome, in turn, consists of a set of **genes**, in which the information is encoded. During the flow of an EA, which can be seen in Fig. A.1, each individual is evaluated and assigned a **fitness**, i.e. a measure of goodness, based on its ability to solve the given task. The EA operators **selection**, **crossover**, and **mutation** are then applied, forming the individuals of the next **generation**. During selection, two individuals are chosen for reproduction, and fitter individuals stand a better chance of being selected than less fit ones. The crossover procedure combines information from the selected individuals, and mutations randomly change pieces of information encoded in the genome of an individual, inserting new genetic material in the population.

A.1 Encoding scheme

There are several ways of encoding the information contained in the genome of an individual. In this project, **explicit encoding**, and **real-number encoding** is used. In explicit

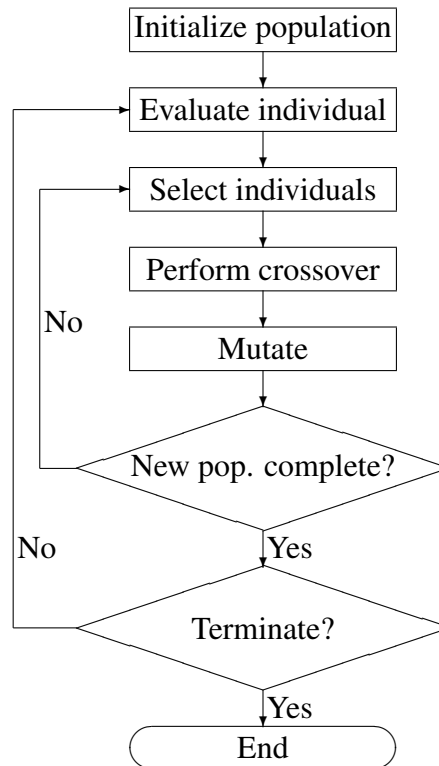


Figure A.1: Flow of a general EA.

encoding, which was used for parametrical and structural optimization of RNNs, there is no encoding/decoding, but the EA operates directly on the structure of the individual (the RNN in this case). In real number encoding, the information is stored in chromosomes, where each gene consists of a real number within a pre-defined range.

A.2 Fitness function

Based on its performance each individual is assigned a fitness value. The choice of fitness function depends on the problem at hand. In this project, the fitness measure was taken as the inverse of the RMSE over the training data set, as described in Section 7.1.2.

A.3 Selection

Selection of two parents can be accomplished in several ways. In this project, **tournament selection** is used. In tournament selection N individuals are randomly chosen from

the population, where N is referred to as the tournament size. Out of the N selected individuals, the best one (having the highest fitness) is selected with probability p_{tour} . With probability $1 - p_{tour}$ another, randomly chosen, individual is selected. The selection procedure is carried out twice, resulting in two individuals selected to generate offspring (through crossover and mutations).

A.4 Crossover

Through the use of crossover, genetic material from two parents are combined. The crossover operator is highly dependent on the encoding scheme, and often care must be taken in order to ensure that the resulting offspring actually encodes valid solutions to the problem at hand. In this project, crossover was not used in the experiment involving an RNN. In the case where an EA was used to optimize the parameters and size of a habituation model consisting of leaky integrators, **two-point crossover** was used. In two-point crossover, two crossover points are randomly chosen for each parent. The genes between the crossover points on one chromosome swaps places with the genes between the crossover points on the other chromosome. In this way, the length of chromosomes may be altered from parent to offspring, by the crossover operator.

A.5 Mutations

Mutations provide the EA with new material to work with. The mutation operator is problem specific, and several mutation operators may be used in one EA. In this project a number of mutation operators were used. Standard mutation operators, such as **full weight mutation**, where a gene simply is given a new, random value within its allowed range, and **real number creep mutation**, where the change of a gene is limited in magnitude, were used to alter the parameters of the problems. In addition structural mutation operators were implemented, to allow for optimization of the structure of the solution as well as its parameters. Implementation details of the mutation operators used in this project are stated in Section 7.1.2.

A.6 Elitism

In elitism a copy of the best individual is inserted in the next generation without any modification. Hence, through the use of elitism the maximum fitness of the population will never decrease during the run of an EA.

Appendix **B**

Recurrent neural networks

ANNs are, as the name implies, networks of computational units, called neurons. An **artificial neuron** has three basic properties: (1) A number of **synaptic weights** (weighted links) which connects input signals (and bias) to the neuron. (2) An **adder** that sums the weighted inputs. (3) An **activation function** that limits the output amplitude of the neuron. The activation function can also serve to add non-linearity to the neuron. A model of an artificial neuron is shown in Fig. B.1.

ANNs have been used in various application fields, including image recognition, time series prediction, non-linear control problems, and as robotic brains [67]. In cases where input-output data is available, specifically in cases where it is difficult to arrive at an analytical model for the problem, or when such a model simply does not exist, ANNs can be advantageous, and is then used as a **black-box model** of the system. RNNs is one kind of ANNs, that make use also of feedback connections between neurons (as opposed to feedforward neural networks, FFNNs). While FFNNs always generate the same output for any given input, the feedback connections provide RNNs with some memory, allowing its output to depend also on previous inputs. ANNs can operate in continuous time or in discrete time. In this project only continuous time RNNs were used, and the relevant equations for these networks will now be presented. For a more thorough review of ANNs, see e.g. [23].

B.1 Network equations

As previously mentioned, the RNNs used in this project operate in continuous time. The architecture of the RNN is such that the neurons are not arranged in layers (as is the

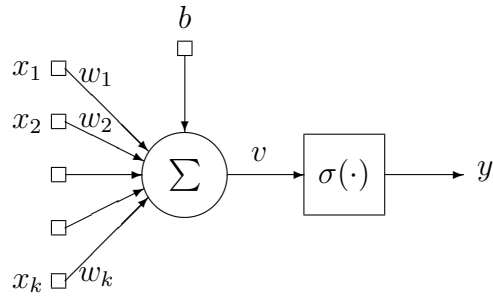


Figure B.1: An artificial neuron. The bias term, b , and input signals, x_1, x_2, \dots, x_k , are scaled by the synaptic weights, w_1, w_2, \dots, w_k , and then summed into v . The output, y , from the neuron is obtained by passing v through the activation function σ .

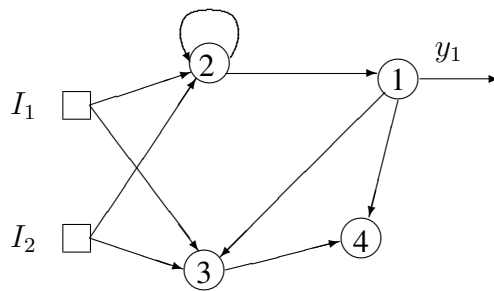


Figure B.2: An RNN with four neurons, one of which has a self-coupling (neuron 2). The network has two external input signals, I_1, I_2 , and one output signal (the output from neuron 1).

case with FFNN). Instead, the network simply consists of a set of neurons with possible connections between all neurons, as well as self-couplings. An example of an RNN is shown in Fig. B.2. For neurons connected in a network, the weighted incoming connections (denoted x_1, \dots, x_k in Fig. B.1) consists of both external inputs, I , and output signals from neurons in the network, y). The dynamics of neuron i in a network consisting of n neurons and m input signals, is governed by Eq. B.1.

$$\tau_i \frac{dy_i}{dt} + y_i = \sigma \left(b_i + \sum_{j=1}^n w_{ij} y_j + \sum_{j=1}^m w_{ij}^I I_j \right), i = 1, \dots, n, \quad (\text{B.1})$$

where τ_i is a time constant, b_i the bias term, y_i is the output of neuron i , w_{ij} the synaptic weight connecting the output of neuron j to neuron i , w_{ij}^I the weight connecting input signal j to neuron i , and I_j is the external input from input signal j to neuron i . Several alternative activation functions, $\sigma(\cdot)$, are common. In this project, it was taken as

$$\sigma(z) = \frac{1}{1 + e^{-cz}}, \quad (\text{B.2})$$

which restricts the output of any neuron to the range $[0, 1]$. For numerical (computer) calculations, the model is discretized, using Euler's method, according to

$$\frac{dy_i}{dt} \approx \frac{y_i(t + \Delta t) - y_i(t)}{\Delta t}. \quad (\text{B.3})$$

Using Eq. B.3 in Eq. B.1 gives

$$\tau_i \frac{y_i(t + \Delta t) - y_i(t)}{\Delta t} + y_i(t) = \sigma \left(b_i + \sum_{j=1}^n w_{ij} y_j(t) + \sum_{j=1}^m w_{ij}^I I_j(t) \right), \quad (\text{B.4})$$

so the discrete model can be expressed as:

$$y_i(t + \Delta t) = y_i(t) + \frac{\Delta t}{\tau_i} \left[-y_i(t) + \sigma \left(b_i + \sum_{j=1}^n w_{ij} y_j(t) + \sum_{j=1}^m w_{ij}^I I_j(t) \right) \right]. \quad (\text{B.5})$$

B.2 Learning algorithms

The parameters of an ANN is assigned in a process known as **training**, where a **learning algorithm** is applied in order to optimize the network parameters, i.e. the synaptic weights, bias terms, and time constants, in the case of RNNs. Some learning algorithms also allows the size and structure of the network to be optimized as well. For FFNNs, **backpropagation** is a commonly used learning algorithm, see e.g. [23, 67]. RNNs are, in general, more complex to train than FFNNs, but one way of accomplishing parametrical and structural optimization of RNNs is to use EAs, which was also done in this project.

