The Role of Affect in Imitation: an Epigenetic Robotics Approach

Arnaud J. Blanchard

A thesis submitted in partial fulfilment of the
requirements of the University of Hertfordshire
for the degree of

Doctor of Philosophy

The programme of research was carried out in the School of Computer Science,
Faculty of Engineering and Information Sciences, University of Hertfordshire (UK).

May 2007

To my parents ....

Acknowledgements

I first would like to thank my main supervisor Lola Cañamero for her direction, support and help but also for the freedom she gave me. I would also like to thank Kerstin Dautenhahn and Carol Britton for having been my second supervisor.

When I arrived in england more than three years ago I have been very well welcome by a first generation of students and researchers. A second generation has been very nice to support me when I was finishing the thesis. I would like to thank them all. Last but not least I would like to thank my family for their essential practical and moral support.

I was supported by a research studentship from the Adaptive Systems Research Group, School of Computer Science, University of Hertfordshire. This research has also received partial support from the EU Network of Excellence HUMAINE, contract FP6-IST 507422 and the EU Research Project FEELIX GROWING, contract FP6 IST-045169.

Arnaud J. Blanchard

Abstract

In animals, humans and robots, imitative behaviors are very useful for acting, learning and communicating. Implementing imitation in autonomous robots is still a challenge and one of the main problems is to make them choose when and who to imitate.

We start from minimalist architectures, following a bottom-up approach, to progressively complete them. Based on imitation processes in nature, many architectures have been developed and implemented to increase quality (essentially in terms of reproducing actions with accuracy) of imitation in robotics. Nevertheless, autonomous robots need architectures where imitative behavior is well integrated with the other behaviors like seeking for stability, exploration or exploitation. Moreover, whether to express imitative behaviors or not should also depend on the history of interactions (positive or negative) between robots and their interactive partners.

In this thesis, we show with real robots how low-level imitation can emerge from other essential behaviors and how affect can modulate the way they are exhibited. On top of proposing a novel vision of imitation, we show how agents can autonomously switch between these behaviors depending on affective bonds they have developed. Moreover, with simple architectures, we are able to reproduce behaviors observed in nature, and we present a new way to tackle the issue of learning at different time scales in continuous time and space without discretization.

Contents

List of Figures

Chapter 1

Introduction

Robotics is promised to play a great role in our everyday life, actually more and more studies are done to design robots as our everyday companions (Dautenhahn, Woods, Kaouri, Walters, Koay and Werry 2005) and enterprises like Sony with its famous dog robot AIBO already propose companion robots for broad public (Kaplan 2005). Although artificial intelligence and robotics are not new disciplines, robots still have to be designed or programmed in order to perform well in precise environments and scenarios. It is probably utopia to imagine architectures allowing robots to perfectly adapt and behave to any kind of situation and in many cases we cannot even evaluate what a perfect behavior is. Actually, the uncertainty or unpredictability of the real world forbids omniscience and a behavior can be good or bad depending on unaccessible elements (e.g. behavior of another unknown agent).

Nevertheless, when we observe nature and especially human intelligence, we can only be impressed by the huge capacity of adaptation of human and animals in very different contexts. Therefore, it should be possible to improve capability of artificial intelligence and improve autonomy of robots. Autonomous robots should

adapt their behaviors to their environment with the minimum need of external intervention or supervision. Moreover for designing robots as generic as possible acting in open-ended environments, designers should avoid planning a-priori all the situations robots may encounter. They should base development of robotic behaviors on dynamical interactions of robots and their environment. Strong influences of the environment on development and emergence of behaviors allow these behaviors to be diverse and adapted to the environment even though they were not explicitly encoded. The study of emerging robotic properties not explicitly encoded is called "epigenetic robotics" and according to (Zlatev and Balkenius 2001) the goal of epigenetic robotics is to understand, and model, the role of development in the emergence of increasingly complex cognitive structures from physical and social interaction.

1.1 Motivation and Goals

The goal of our research[1] is not only to provide technical solutions but also to understand and to model processes observed in nature. On the one hand, observing nature and producing biologically plausible models could inform the design of efficient minimal architectures. These architectures can then be implemented and applied in order to solve engineering problems. On the other hand, these architectures are powerful tools to help us understand how the brain solves cognitive problems and robots are very useful for this (Guillot and Meyer 2001). We aim to explain the maximum number of phenomena with the minimum amount of complexity or assumptions in the models. In this context, we use a "bottom-up" approach,

[1]This thesis presents the work I have carried out under the supervision of Lola Cañamero, but for style reasons I use the term "we" although the work presented here is only my original contribution.

which advocates incrementing the complexity of the architectures step by step. We try to make each step as simple as possible taking advantage of emergent properties which issue from the dynamics (Steels 1994). Our structures are based on the Perception-Action (Per-Ac) model (Gaussier and Zrehen 1995) where there is not explicit representation of the world (Brooks 1991).

Other studies (Calinon and Billard 2007, Fellous and Arbib 2005, Alissandrakis, Nehaniv, Dautenhahn and Saunders 2005, Dautenhahn and Nehaniv 2002, Billard 2000) have shown that imitation can help make robots more autonomous. One of the big issues in imitation for autonomous robots is to modulate the imitative behaviors and answer the questions of when and who to imitate (Dautenhahn and Nehaniv 2002). For other issues, especially for action selection, researchers have proposed the use of emotion to modulate behaviors (Fellous and Arbib 2005, Avila-García and Cañamero 2004, Cañamero 2001, Cañamero 1997). Moreover, it has been observed that there are strong correlations between emotion, affect and imitation (Heyes 2001, Hatfield, Cacioppo and Rapson 1994).

Therefore, we want to answer the scientific question of when and who robots should imitate in order to increase their autonomy, especially autonomy of expressing the right behaviors at the right moment. We use biologically plausible architectures which help to understand the role of affect in different behaviors, in animals or humans (especially children). Within our epigenetic robotic approach we first present a way to increase synchronization in robotics and how a notion affect can be modeled using a learning process at different time scales. This notion of affect allows us to modulate the expression of behaviors such as seeking for stability, exploration or exploitation. Then we show that appropriate imitative behaviors can emerge from modulation by affect of other behaviors. Finally we present how we

can extend this work for more general learning problems.

1.2 Overview of the Thesis

Chapter 2 . In this first chapter we develop the problem of imitation, affect and robotics in order to set the background of our work. We also set important notions that we use throughout the thesis.

Chapter 3 . We propose an architecture to improve synchronization and reactivity during imitation for tasks where robots have to follow a target. We apply it on a real robot and present the results.

Chapter 4 . We present a simple way to model affect and we describe our architecture allowing robots to generate behaviors similar to those observed in nature. We present the result of its implementation on a real robot. This architecture allows us to simulate affective bonds and to set the core of our work.

Chapter 5 . Based on the work done in the previous chapter, we proposed a coherent integrated robotic architecture generating behaviors of seeking for stability, exploration and exploitation. Affect is used to balance these multiple behaviors and we show that low-level imitation can be an emergent property of this architecture. We present the result of a real-world implementation of this architecture.

Chapter 6 . We improve the architecture, in order to make robots able to handle more complex environments. We also show the result of a partial implementation.

Chapter 7 . We discuss the limitations of our work and propose solutions—that we have not implemented—to overcome them.

Chapter 8 . We summarize the dissertation and present new perspectives our work has raised.

Appendix A contains a glossary of some of the terms used throughout this dissertation, appendix B presents the publications based on the work of this thesis, and appendix appendix:software presents the programming framework that we have developed in order to control the robots.

We also include a CD-ROM with the sources of our programming framework and several videos that we will refer to throughout the dissertation.

1.3 Contribution to Knowledge

1. Our main contribution is our novel view of low-level imitation being an emergent property of simple architectures providing behaviors of seeking for stability, exploration and exploitation. Moreover, the imitative behavior is modulated by affect allowing agents to only imitate other beneficial agents and avoiding interactions with noxious agents.

2. Another important point is the introduction of the notion of "desired sensations" and their use in the context of remembering rewards or punishments without needing to discretize the world. The learning process is about to remember which are the situations corresponding to events like "best rewards", "best familiarity", "worst rewards", etc., and not the evaluation of expected rewards for each situation (or each kind of situation) as it is often the case.

3. Moreover, the possibility to learn simultaneously many "desired sensations" at different time scales and for different balances of the importance of reward, punishment and familiarity, leads to architectures where behaviors of robots can be smoothly modulated. Robots could therefore focus on reaching rewards, focus on avoiding punishments or try to reach familiar situations solely with a modulation.

4. We propose models of natural phenomena such as the imprinting phenomenon or exploration and imitation modulated by affect. Even if we are missing data to really argue that these phenomena are modeled as we propose, we can at least say that it is a possibility and that these phenomena do not necessarily need complex or high-level processes.

5. As secondary contribution, we have developed a new programming framework that is useful in developing robotic architectures using a bottom-up approach in open ended environment. It is very important with this approach for the designer to be able to monitor everything that happens in the architecture in order to be able to take advantage of unexpected events.

Our work allows to build better learning systems in robots, especially when the time between actions and rewards is not constant. It also allows to improve human-robot interactions through imitation and adaptation of behaviors with the interactive partners. Finally it helps to understand the development of affective bonds and imitative behaviors in animals and humans. The contribution to knowledge presented throughout this thesis has been published in, or submitted to, relevant conferences and journals. See Appendix B.

Chapter 2

Imitation, Affect and Robotics

The question of whether a computer can think is no more interesting than the question of whether a submarine can swim. — Edsger Dijkstra (1972).

2.1 Context

To survive in different environments, *autonomous agents*—animals, robots or actors of a simulation, see (Steels 1995)—must act, learn and communicate, and imitation (in its wide definition) is very useful to achieve these tasks (Bakker and Kuniyoshi 1996).

- Acting can be initiated merely by being in the appropriate context (local enhancement) or by focusing on the appropriate stimulus (stimulus enhancement) (Zentall and Akins 2001) thanks to the presence of another agent. The actions of the other agent can also be a means to focus attention and attention is an important issue in autonomous robotics (Arkin 1998, pp 276–279).

- Learning is a complex process which can be performed by trial-error or by imitating agents with more knowledge. This second solution is widely used to improve the learning of robots (Billard and Mataric 2001, Billard 2000, Hayes and Demiris 1994) and can even be used as a programming method (Alissandrakis et al. 2005, Kuniyoshi 1994) but it is also suggested to be an important phenomenon in the development of infants (Rochat 2002, Užgiris 1999).

- Communication seems to be rooted in imitation and synchronization processes (Nadel, Revel, Andry and Gaussier 2004, Kozima and Zlatev 2000, Nadel, Guérini, Pezé and Rivet 1999, Trevarthen, T. and Fiamenghi 1999, Billard, Dautenhahn and Hayes 1998). It can also be enhanced through emotion linked with imitation (Kugiumutzakis, Kokkinaki, Makrodimitraki and Vitalaki 2005, Heyes 2001, Hatfield et al. 1994).

Therefore imitation is a very important phenomenon for autonomous robots but also in the understanding of biological processes like the relation between imitation, affect and development of attachment bonds (Heyes 2001, Trevarthen et al. 1999, Hatfield et al. 1994). It is difficult to use a unique definition of imitation as numerous definitions of imitation have been proposed by authors such as (Dautenhahn and Nehaniv 2002, Mitchell 1987, Thorpe 1963)—see also (Breazeal and Scassellati 2002b, Dautenhahn and Nehaniv 2002, Galef 1998, Bakker and Kuniyoshi 1996) for reviews—and among the problems raised by imitation, five questions (the "Big Five") seem fundamental: what, how, who and when to imitate, and what makes a successful imitation (Dautenhahn and Nehaniv 2002). In (Breazeal and Scassellati 2002a), the authors raise again the question of when and what to imitate and they

also ask how a robot can map observed actions onto behavioral responses. The problem of the mapping and what to imitate is known as the "correspondence problem" where states of the system (objects or environment), actions or goals have to be considered for imitation (Alissandrakis, Nehaniv and Dautenhahn 2007, Nehaniv and Dautenhahn 2002). This is especially a problem when agents have dissimilar bodies or are in different contexts—for example manipulating different objects or using different effectors where exact copy cannot be done.

2.2 Imitation

In this context, the main question we raise is how to make robots more autonomous in imitation, especially in the process of deciding when to imitate. The question of improving the quality (essentially in terms of reproducing relevant actions with accuracy) of robots' imitation skills is very important and has been well studied (Saunders, Nehaniv, Dautenhahn and Alissandrakis 2007, Saunders 2006, Calinon and Billard 2005, Alissandrakis et al. 2005, Gaussier, Moga, Banquet and Quoy 1998) however, for autonomy it is also very important that robots can decide by themselves when to imitate the right behavior in the appropriate context.

Among the many definitions of imitation, we can extract a few important points.

Novelty Early in 1898 imitation is seen as "Learning to do an act from seeing done" (Thorndike 1898), novelty is also important for Thorpe who defines imitation as "the copying of a novel or otherwise improbable act or utterance, or some act for which there is no instinctive tendency" (Thorpe 1963).

Intention Thorndike defined that an imitated behavior needs to be understood for

true imitation (Thorndike 1911) and in the definition of Mitchell (Mitchell 1987) the copy has to be designed to be similar to the model for imitation.

Contagion Behaviors can be unconsciously elicited by actions of others (Heyes 2001) (yawning for example), or can be facilitated, inhibited or retarded because of others (Zentall and Akins 2001).

In animals and even in humans, it is difficult to evaluate what is really intentional and in behavior-based AI with low-level or reactive control systems (Brooks 1991) we cannot define what "understand" means in robotics. We do not argue in this thesis that we model "true imitation" and we will refer to "low-level imitation" for actions similar to the actions of another agent and induced by this other agent—i.e. actions which would not occur without the other agent. In our context, the context of autonomous agents, a successful imitation would be an imitation which increases the success of the agent—in its survival, learning or realization of specific tasks. We mainly focus on the question of when to imitate (i.e. in which context) and especially how to integrate, coordinate imitative behaviors with other behaviors (learning, exploring, exploiting situations, etc.). For this purpose, we want to study the role of affect in imitation as affect seems to have a strong effect on imitation and may be useful in modulating imitation (Kugiumutzakis et al. 2005, Heyes 2001, Hatfield et al. 1994). The quality of imitation is not considered as a critical point here and we will only use simple or low-level imitation, although we will care about the quality of the synchronization because, as we will see, synchronization is an essential issue in our problem.

2.3 Affect

In the first sentence of his thesis (Avila-García 2006), Avila-García warns us about the risk to use the word "emotion": *It is exceptionally dangerous to use the term emotion in a scientific environment, for many prejudices arise with its mere mention.* We have essentially the same issue with "affect" and this is mainly due to the fact that there is no clear definition and affect has a wide vague definition. Actually, affect is often used in literature treating emotions, imitation or robotics (Arbib and Fellous 2004, Heyes 2001, Velasquez 1999, Blaney 1986), nevertheless it is not defined. One could wonder why we should use words or notions if they do not apply to reality; we will never be able to say that robots feel emotions or have affect. Actually our architectures are only equations and probably—and it is euphemism—do not feel emotions or have affects.

We need principles helping us to modulate imitation and we need a clear functional definition of them to work with. However, similar principles have already been studied in psychology, especially the relation between affect, emotion and imitation (Kugiumutzakis et al. 2005, Heyes 2001, Hatfield et al. 1994) therefore, even though in robotics we only need mathematical definitions, we use psychological terms in order to have intuitive notions of what we are talking about and to make analogies. One of the notions that we will need is the notion expressing the fact that an agent evaluates its situation and in function of this evaluation will change its behavior, especially imitative behavior. Following the proposals of (Arbib and Fellous 2004, Likhachev and Arkin 2000, Dunn 1977) we use the notion of *comfort* to represent this evaluation and we propose a working definition of *comfort*:

comfort: global evaluation of the "goodness" of a situation based on the evaluation

of exogenous factors (affect) and endogenous factors (well-being).

The definition of affect in the Oxford American dictionary is: *emotion or desire, especially as influencing behavior or action.* And emotion is *an instinctive or intuitive feeling as distinguished from reasoning or knowledge.* Therefore we propose a functional definition of affect:

Affect: immediate subjective (based on exogenous factors, external stimuli) evaluation of a situation (expectancy of well-being) without direct or logical explanation.

We use affect to describe the exogenous component of *comfort* as it is described in (Likhachev and Arkin 2000, Dunn 1977) and in Chap. 5 we make precise how we use and compute this definition in robotics. The endogenous component of comfort is "well-being" which represents satisfaction of the primary needs (Avila-García 2006, Likhachev and Arkin 2000, Kahneman, Diener and Schwarz 1999, Dunn 1977) and we make precise our robotics use of this notion in Chap. 4. We therefore present the working definition of well-being.

Well-being : endogenous (from internal factors) component of comfort given by measuring the satisfaction of the primary needs.

2.4 Robotic Approach

Studies in psychology and neuroscience tend to show that action and perception are encoded at the same level (Matarić 2002, Vogt 2002, Prinz 1997, Decety 1996). These observations converge to the *ideomotor theory* originally proposed by William James (1980) explaining that voluntary actions are in fact the result of the "will"

of sensory feedback or sensory consequences rather than the "will" of performing actions themselves. An extension of this theory has been used to explain imitation (Prinz 2005) using the fact that perceptions and action are tightly coupled in the brain. Observing the consequences of an action can raise the "will" of obtaining the same perception and therefore producing the corresponding action producing imitation of the initial action. We can see a successful implementation in robots of the extended theory of imitation in (Saunders 2006). The extended ideomotor theory uses notions of ideas or wills requiring cognitive functions of high level and does not help in solving the problem of autonomy in order to decide whether an agent should have the "will" to imitate or not. Nevertheless, the principle is very interesting and it can be used at a lower level in case of simpler imitative behaviors called "low-level imitation". In low-level cases, the process is more related to sensory-motor associations than perceptual-motor associations. The difference may be subtle, however in the bottom-up approach to robotics, perception implies a higher level of treatment than sensations. We detail this difference in the coming subsection.

2.4.1 Sensation and Perception

Gibson in (Gibson 1979) proposed that perceiving an object for example corresponds in fact to perceiving all the actions we can do with the object. He called all the possibilities raised by an object "affordances", and actually it has been observed that the vision of an object like a hammer activates the part of the brain corresponding to the action of handling the hammer even if subjects do not want to handle the hammer (Ellis and Tucker 2000). This definition of perception is very interesting for robotics, because it does not imply cognitive functions of high level like recognizing or understanding. Therefore the term "sensation" is used to express the raw data

coming from the sensors like the sense vector in (Chesters and Hayes 1994).

The distinction between perceptions and sensations in robotics is stressed in (Maillard, Gapenne, Gaussier and Hafemeister 2005) where a perception is a *dynamical sensory-motor attraction basin* and this notion is used to generate appropriate behaviors when "perceiving" (e.g. moving toward the object). However, the sensations produced by an object can activate several sets of possible actions and as in optical illusions (see examples in Fig. 2.1) there is not always a true or unique possible interpretation. Similarly, several perceptions of movement can be raised by the variation of visual sensation. Actually the agent can either perceive itself moving or perceive the observed object moving. It needs other sensors to select the correct perception. We define notions of sensation and perception even though we will barely use the notion of perception.

Sensation: set of raw data coming from the sensors.

Perception: interpretation of the ongoing action or affordance raised by a sensation.

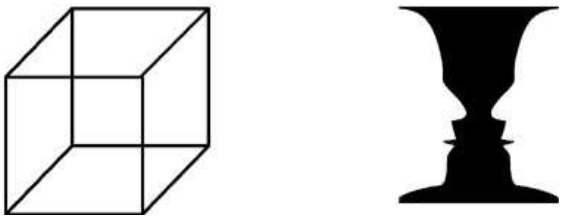


Figure 2.1: Examples where the same visual sensation leads to different perceptions

We can find applications of low-level imitation (i.e. very automatic) by tightly coupling sensations and actions in (Demiris and Johnson 2003, Andry, Gaussier and Nadel 2003, Gaussier et al. 1998) and we detail the experiment of Gaussier and colleagues in the coming subsection.

2.4.2 An Example of Low-level Imitation

Although imitation behaviors can be very complex, the researchers (Andry et al. 2003, Gaussier et al. 1998) modeled imitation at a low level merely by merging vision and action. In their experiment, they made a robot learn the sensory-motor association between the position of its arm (proprioception) and its vision. Once the associations had been learned, the experimenter turned the camera and directed it on his own arm—instead of the robot's arm—and started to move his arm. The robot confounding the arm of the experimenter with its own arm, perceived an error between its proprioception and the input from its camera. Therefore, it tended to reduce this error (hardcoded homeostatic control) by moving its arm to adapt the proprioception with the vision. The result is that the robot reproduced—and then imitated—the experimenter's movements.

This elegant way of producing low-level imitation circumventing the correspondence problem is very interesting as it does not need complex architectures or strong assumptions. However, it needs an external action from the experimenter—turning the camera—to switch the robot between the phase of learning to the phase of imitation. Therefore the problem for an agent of being autonomous even in the decision of imitating or not is still not solved.

2.4.3 Our Vision of Low-level Imitation

In Chap. 5 we will present a new vision of imitation based on the same kind of approach but where low-level imitation is not produced by the reduction of errors between the usual sensation and the actual sensation, but on the contrary by trying to amplify the difference—in the sensory-space—between the usual sensation and

the current sensation. With this vision, imitation of a moving arm is executed because this moving arm produces an unexpected sensation of movement in an area of the visual field and the robot moves its own arm in the same area to increase the sensation of movement in this area using the sensory-motor association. Therefore it would produce low-level imitation as well, but because of a different reason (trying to increase an error instead of trying to decrease an error). In this case, we would not need to turn the camera in another direction in order to make the robot imitate and as we will see imitation is in the continuity of other behaviors of the robot.

2.5 Summary

The literature review has first shown the importance of imitation for autonomous agents. Then we have seen than imitation is complex and one of the main difficulties is to decide when and who to imitate. Independently to this problem, it has been observed that affect and imitation are linked and we wonder whether affect can be useful—and maybe necessary—to have autonomous agents. Recent works on perception-action coupling allow us to generate architectures combining affect and imitation inspired by natural phenomena, and to present a new vision of low-level imitation.

The main advantage of this new vision of low-level imitation is that it allows us to produce imitative behaviors modulated by affect within an architecture producing other useful behaviors for autonomous agents. We will present the successive behaviors of imprinting in Chap. 4, exploration, exploitation and imitation in Chap. 5, but before we start by presenting a method we have developed in order to improve synchronization in robots.

Chapter 3

Visual Velocity Detection

We want to study the correlation between affect and imitation, and synchronization seems to have strong effects on affective bonds (Hatfield et al. 1994), therefore we need systems able to provide good synchronization in order to test and use this phenomenon. For example, we may need to make robots dance together, but to synchronize a dance between two robots we need temporally accurate systems. Actually, with repetitive movements such as a dance, a small temporal lag may suggest no synchronization or even opposition. In robotics, object-following strategies are usually achieved either by efficient but complex systems which do not fit with our approach and our very simple hardware, or by adequate systems with detection of movements (e.g. (Gaussier et al. 1998)) but which are not accurate enough for synchronization.

In their studies of imitation tasks using robots, (Andry et al. 2003) use the amount of movement (temporal variation of intensity of light) to perceive the target position. This technique is efficient and simple as it does not need complex visual tasks such as object recognition. However, the problem with this mode of imitation

in robotics is that there is always a delay between the object agent (the agent to be imitated) and the subject agent (the agent imitating). The subject agent starts to move only after the object agent has moved into a new position.

In this chapter, we present a first step towards the development of suitable mechanisms based on visual velocity detection with respect to our general approach. This system is not only applicable in cases of precise reproductions of movement (e.g., when mirroring a movement) but also in cases where movements do not need to be precise but must be very well synchronized, when dancing for example. In this latter case, our velocity detection system alone (without positional information) allows an agent to copy beats perfectly even if the amplitude of movements is not perfect. After presenting the problem to be addressed, we show how to solve it with a classical position detection system and compare it with the use of our visual velocity detection system. Our experimental results show how our system outperforms systems based on position detection but raises the problem of the quality of the imitation. We therefore propose in the last part, a way to combine advantages from both methods.

3.0.1 Problem addressed

In the context of an autonomous mobile robot that has to interact with other agents in its environment, we have addressed the problem of achieving natural and fast reactions by the robot to detected changes in its environment. By this, we mean that we want the agent to react as quickly as possible to the changes in its environment (in our case, motions in the visual field). Minimizing the reaction time to respond to environmental changes is very important, in particular when the limited perceptual and computational resources of the agent impose severe constraints, as is the case

with the Hemisson robot (www.k-team.com) we have used in our experiments. The camera available for this robot can only process fewer than ten images per second, while an average camera can process about 30 images per second. Moreover, the camera is only able to provide "images" of a line of 102 grayscale pixels, although this is sufficient to detect the position and the velocity of object in only one dimension.

We have therefore designed four architectures to allow a robot to follow a target or to be synchronized with the movement of a target:

1. position detection with Winner-Take-All (WTA) (Sec. 3.1 method 1),
2. position detection without WTA (Sec. 3.1 method 2),
3. velocity detection with focalization[1] (Sec. 3.2 method 3),
4. and velocity detection without focalization (Sec. 3.2 method 4).

The first two are based on position detection and the last two are based on velocity detection. We have implemented these architectures in a Hemisson robot (the "subject" robot) fitted with a camera. The target is composed of two vertical strips or a pattern of strips drawn on a white paper attached to an object Koala robot, as shown in Fig. 3.1. We use a Koala instead of a Hemisson as target because we need to calculate the position of the target; Koala is fitted with odometers, giving us access to the real velocity of the target and allowing us to drive it with a sinusoidal movement. We describe the architectures based on position detection in Sec. 3.1 and the architectures based on velocity detection in Sec. 3.2. Finally we will discuss the possibilities and advantages associated with combining both methods in Sec. 3.3.3 before concluding in Sec. 3.4.

[1] selection of an area where the velocity detection is applied.

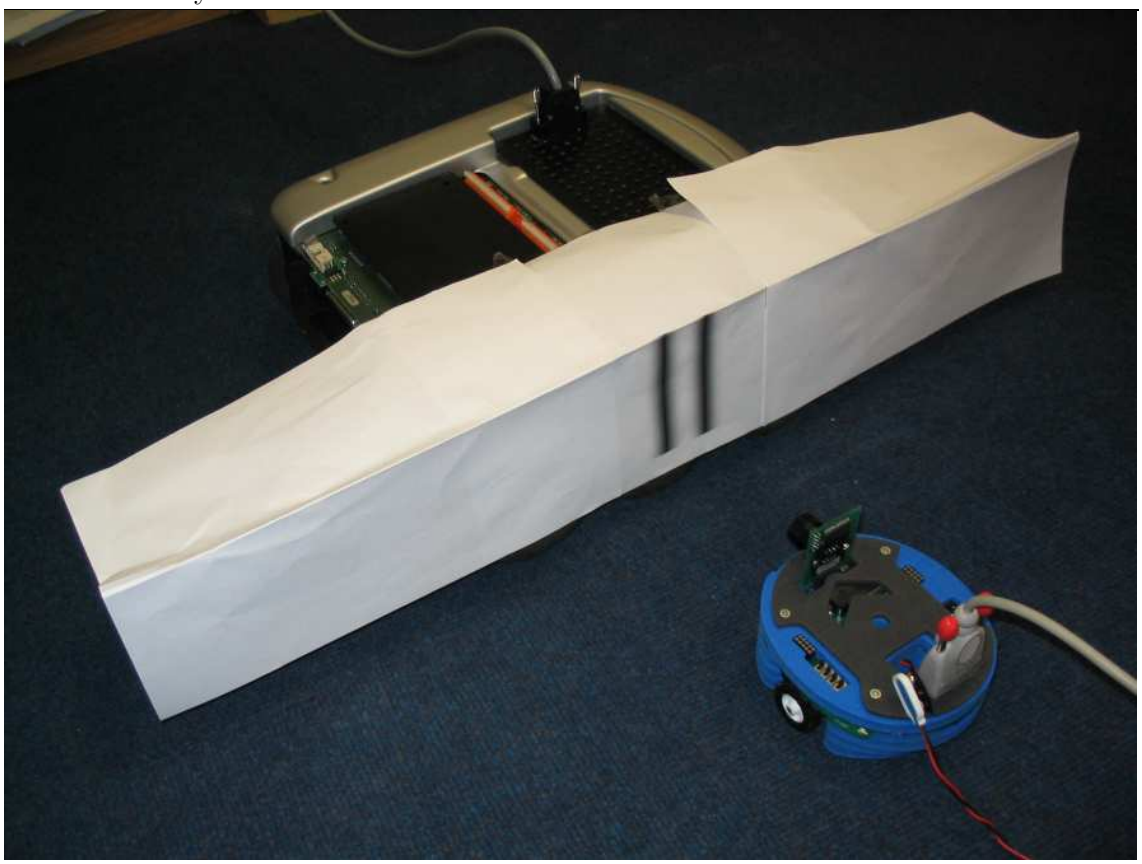


Figure 3.1: Experimental setup. On the left the Koala robot (object) moves the target observed by a Hemisson robot (subject) on the right.

3.1 Position detection

The basic principle underlying position detection is the one described in (Gaussier et al. 1998). S_t is an array—one dimensional hypercube in our software—of the intensity of the pixels from the sensor (camera) at time t. The area of maximum variation of intensity of light corresponds to the area where the object is moving and to compute the "map" of the intensity differences, we apply a square function on the intensity difference, $(dS_t = S_t - S_{t-1})$, between two successive images. In

addition to being continuous and revealing the absolute values of differences, the square function applies a kind of filter by neglecting small differences (which could be noise) and emphasizing the important ones. We obtain an image of the moving contrasted areas of the target ($M_t = (dS_t)^2$) and to obtain a smooth and stable activity bubble focussed on the center of the target, we spatially and temporally filter it. Spatial filtering is realized by convolving the signal with a gaussian function, whereas temporal filtering is realized by smoothing temporal changes (see Eq. (3.1))

$$\overline{M_t} = \frac{\overline{M_{t-1}} + \tau M_t}{1 + \tau} \tag{3.1}$$

with τ a time constant. Temporally smoothing the signal brings stability and also makes it possible to maintain the position of the target even if it does not move for a small period. Then we use a winner take all (WTA) to set to one, the value of the array at the position with the maximum quantity of movement among all the positions of the visual field. This activation can therefore be used in order to activate motor commands in order to follow the target. The subject robot needs to slow down when it is about to reach the target in order to be smooth and to avoid over shooting. In addition, the robot has to stay on the same target and neglect perturbations appearing on the sides of the visual field, therefore it is better to ignore extreme positions. Integrating the position activity signal with the derivative of gaussian is simple, biologically plausible (Amari 1977) and produces an activity suitable for driving a robot that is tracking. The produced activity is negative when the moving target is on the left and positive when it is on the right, moreover, the produced activity is null or low when the activated position is either in the center or on a extreme position of the visual field.

This is a classical method (method 1) to allow robots to follow a moving target. However, we can again simplify the architecture by removing the WTA (method 2). The new resulting behavior of the robot is slightly different but still interesting: now, the virtual speed of the subject robot does not depend only on the position of the target, but also on its contrast and activity. As there is no WTA to trigger the value M_t (position of the moving target) to a threshold independent of the input activity, the speed of movement of the subject robot will directly depend on the quantity of movement on the perceived position (M_t) of the moving target. Therefore, the agent does not respond to the target's position exactly as it would with the previous architecture; however, this new behavior is closer to the behavior we expect from an agent—animal, human or even robot. In fact we naturally have this kind of expectation when testing it. During the tests we moved the target with the hand and we automatically try to augment the quantity of movement of the target when the robot does not react fast enough. This even though we know that it is useless when there is a WTA as the quantity of movement does not matter. We present the global architecture in Fig. 3.2 where we can see the part not needed to apply the method 2.

3.2 Velocity detection

Position detection methods work fine (the robot virtually follows the target), however, there are unavoidable delays between the movement of the target and the movement of the subject robot. In order to increase the reactivity of the subject robot to changes in its environment, we proposed to use the velocity, instead of the position, of the target as the input information for synchronization. As can be seen

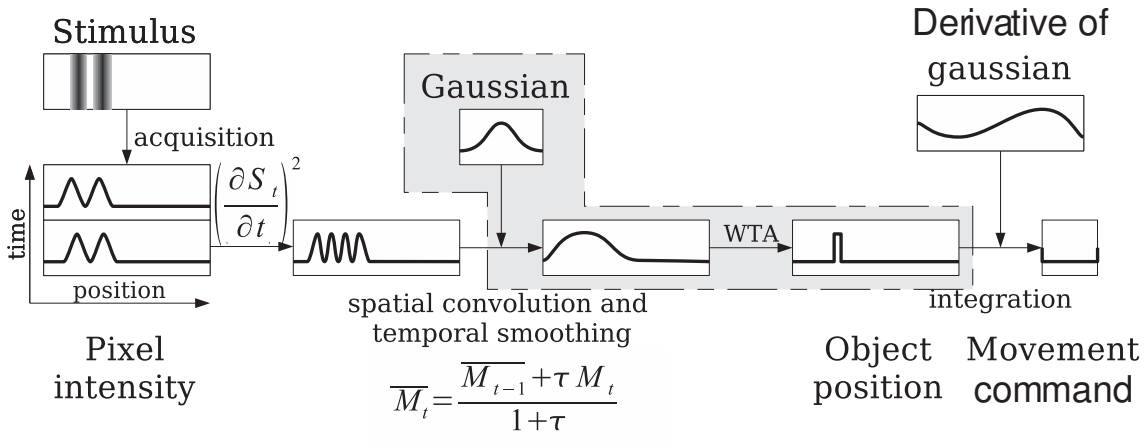


Figure 3.2: Position-following architecture. The gray part corresponds to the part that we can remove in order to obtain a simpler architecture with slightly different behavior.

in the experiments, this method allows the subject robot to start to move as soon as the object robot (target) moves without having to wait for a significant change in the position of the target. Following the study of (Barron, Fleet and Beauchemin 1994) comparing different methods, a simple and efficient method for detecting velocity is the spatio-temporal gradient technique. This technique has been used by (Anderson 2003), who has shown that the technique is biologically plausible in several of its aspects and raises the same perceptual errors that we observe in people. Another advantage of this technique is that it is independent of the contrast and intensity of light of the stimulus (providing that the contrast and intensity are not too low). This velocity detection method, proposed by (Johnston, Benton and Morgan 1999), is based on the hypothesis that each point of an object has a constant intensity. Therefore, any variation in the luminosity of an image can be assumed to

be due only to movements of its objects. By considering P the perceived velocity, k a constant (essentially depending on the distance to the object), and $S_{t,x}$ the light intensity at the time t at the position x, we have :

$$P_{t,x} = k\frac{\frac{\partial S_{t,x}}{\partial t}}{\frac{\partial S_{t,x}}{\partial x}} \tag{3.2}$$

In Eq. (3.2), dividing the variation of intensity by the contrast, is a problem when the contrast is almost null. It is however not surprising because without contrast we cannot estimate the movement of an object. To solve this problem we use a threshold which allows the perceived velocity to be neglected when the contrast is too low. (McOwan, Benton, Dale and Johnston 1999) propose a nice solution based on the observation that the visual system measures at least three orders of spatial derivative. They therefore use multiple orders of spatial and temporal derivatives which are unlikely to be all null at the same time. We can see an example of a second order in Eq. (3.3).

$$P_{t,x} = k\frac{\frac{\partial^2 S_{t,x}}{\partial x \partial t}}{\frac{\partial^2 S_{t,x}}{\partial x^2}} \tag{3.3}$$

If we want the robot to consider the entire visual field we only need to sum the velocity of each pixel (the video 'synchronization with moving strips.avi' of the CD-ROM presents an application of this technique). However, if we want to focus its attention on a precise area, we must first isolate this area using an adequate filter. If we only want the robot to adapt its velocity to that of the target, we can use the position detection method from the previous section to select the area to consider. Figure 3.3 describes the architecture implementing method 3 with selectivity of

the target and method 4 without selectivity of the target; both methods use visual velocity detection. We have implemented these different architectures on a Hemisson

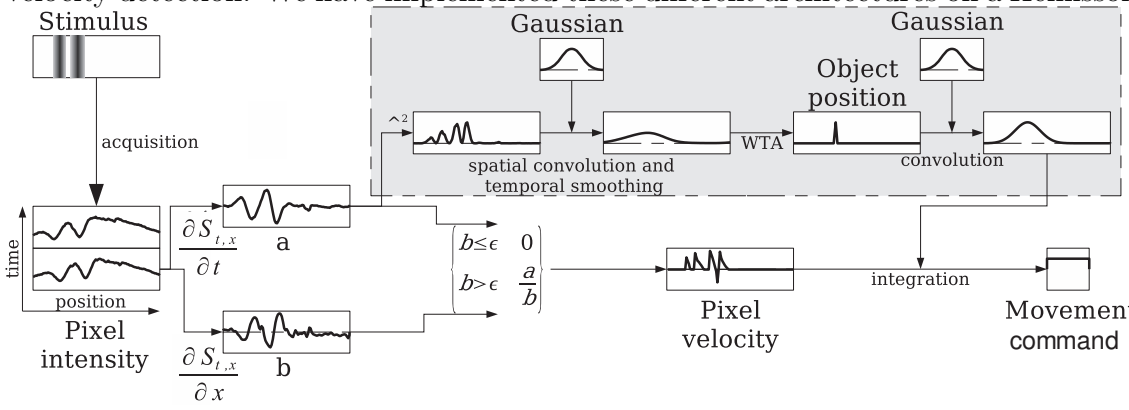


Figure 3.3: Architecture to detect the velocity of a focussed object (method 3). The gray part could be replaced by a large static gaussian allowing the architecture to take care of the entire visual field (method 4). In this scheme, the curves are the result of real data.

robot (K-Team 2002-2006) and we present and compare the resulting behaviors in the next section.

3.3 Experiments

3.3.1 Setup

In all experiments, we have used the Hemisson robot as the subject robot and the Koala robot to carry the target. We also measure the magnitude of rotation command that the Hemisson would send to its wheels. Since it is impossible to know the exact position of a Hemisson robot (it does not have odometer sensors), we have had to design experimental setups which account for this constraint: all the computation is carried out normally but we inhibit the action and instead record

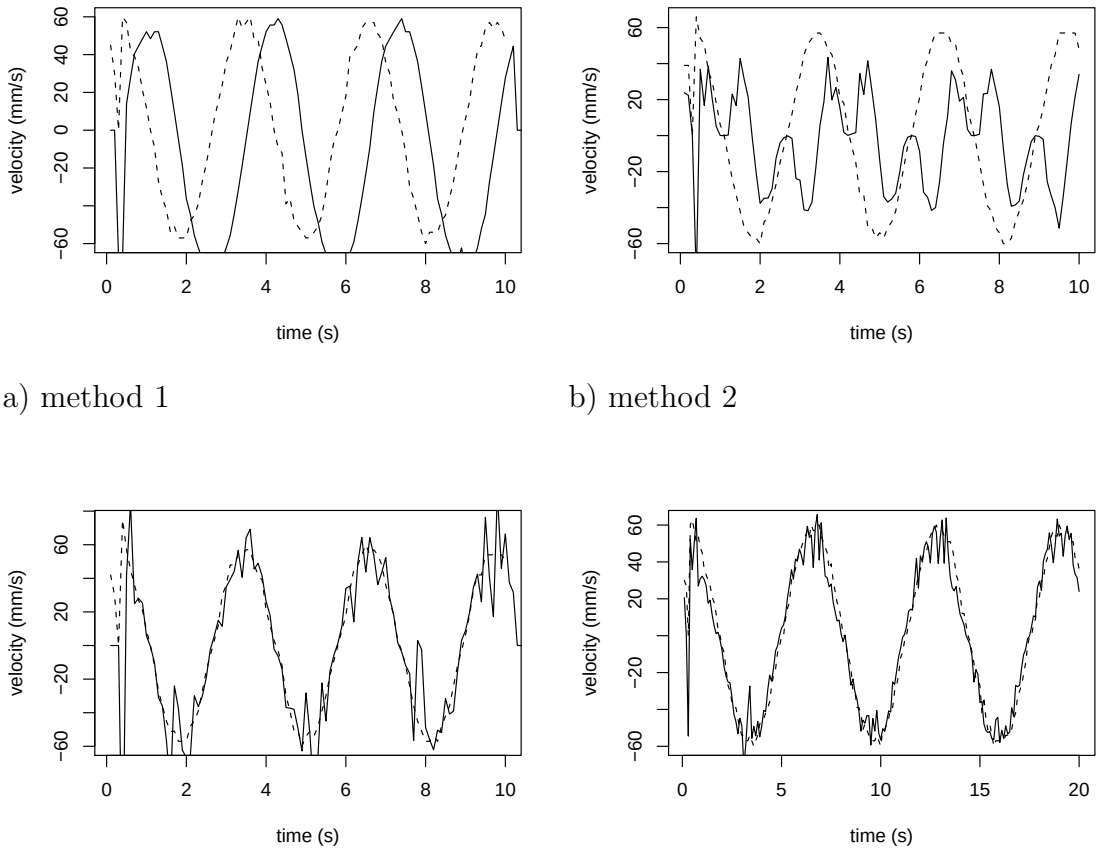
the commands of rotation given (but not executed) to the wheels.

We use the same setup (see Fig. 3.1) for each of the first three methods: the Koala robot moves forward and backward with a sinusoidal velocity carrying two vertical strips drawn on the target. The subject Hemisson observes (without moving) the target from a distance of about 3.5 inches. To test the final method 4 we use a similar setup but this time the target is a wider pattern of vertical strips covering the entire visual field.

3.3.2 Results

In Fig. 3.4, we present the result of one experiment among many experiments which produce similar results. The first graph on the left (a) shows the results of the synchronization task using position detection with WTA (method 1) and the second graph on the right (b) shows the results without WTA (method 2). The two graphs on the bottom show the results of the synchronization task using detection of velocity: focused target (method 3) is shown on the left graph (c), and no focalization with a wide target covering the entire visual field (method 4) is shown on the last graph (d).

On each graph, the singularities observed over the first two iterations have no meaning. They are due to the sudden start of the Koala and the fact that there is no previous image in memory to calculate the temporal light derivative. The dashed line corresponds to the velocity of the object agent and the solid line corresponds to the velocity command given by the subject agent. Each time-step is 100 ms in length (about 10 frames per second).



a) method 1

b) method 2

c) method 3

d) method 4

Figure 3.4: Results of the four methods tested. The dash line corresponds to the velocity of the object robot (target) and the solid line corresponds to the command of velocity of the subject robot. At the top we see the velocities of the robots during a task of tracking using the technique of position detection, with the WTA (method 1) on the left and without the WTA (method 2) on the right. At the bottom we see the results of the task of tracking using the technique of velocity detection, with a focused target (method 3) on the left, and with no focalization, and with a wide target covering the entire visual field (method 4) on the right.

3.3.3 Discussion

All the methods we have presented here have, depending on the tasks needed during agents' interactions, particularly in imitation and synchronization, some interesting

properties and advantages.

The first method, using position detection, is very useful for following the trajectory of a target. However, the delay that it produces is not convenient for synchronization tasks or for frequently changing situations. For example, during a dance, a small lag will de-correlate two agents.

The second method, which uses a simpler version of the same principle, is well suited for following a target's position even with a small embedded system—little computational power is needed—and also for some specific behaviors. However, the fact that the velocity of the reaction does not depend solely on the position of the target but also on its quantity of movement can be a problem. Although, this can also be an advantage if we want to make the perceived behavior more natural to a human observer—it seems natural to have more response when the target is more active regardless of its position. This also corresponds to the fact that we only see motion in the peripheral vision.

The last two methods both use velocity detection, with one focusing velocity detection on the area of the target and the other applying velocity detection to the entire visual field. The focalization is achieved by using the position detection to first define the area where the velocity detection should be applied. The advantage here is that the reaction is both fast and proportional to the real velocity of the target and the image of the background does not influence the results. However, this method is more complex requiring the previous position detection system on top of the system with detection of velocity.

The final method, which integrates each pixel's velocity without any focalization, allows us to achieve pure synchronization. The position of the target is irrelevant as the entire visual field is considered. If the target is moving in the visual field,

the observing robot will move in the same direction but with a non-proportional velocity since the background is considered. This method is especially useful when the entire visual field is moving—e.g. when the camera itself is moving. It allows the movements of an agent to be stabilized, in much the same way as observed in a fly (Holst and Mittelstaedt 1950). Holst and Mittelstaedt (Holst and Mittelstaedt 1950) investigate this property in flies by putting a fly in a rotating drum with black and white strips. The fly moves to stay in the same relative position to the drum. We have been able to reproduce this fly phenomenon with our robot. We put the robot in a drum with black and white strips and, when we move the drum, the robot turns with the same velocity in the same direction (see the video 'rotating drum.avi' of the CD-ROM). Only the optical flow is used, and we therefore observe some drift (the robot does not face exactly the same strips) . It is necessary to manually define a gain coefficient to correlate the perceived optical flow with the wheels' velocity. It seems that the gain must be genetically defined in the fly since this insect will turn indefinitely in the wrong direction if the head is turned and glued by 180 degrees. The robot exhibits exactly the same behavior when the camera is turned by 180 degrees too.

It can see that both kinds of methods (position detection or velocity detection) have advantages and disadvantages. Position detection does not produce drifts but is not very reactive. Velocity detection is very reactive but produces a drift which prevents a prolonged interaction due to the target becoming lost. This dilemma is also commonly encountered in engineering disciplines. To drive a system it is possible to use either the position (first order) with a stable but slow system, or the velocity (second order) with a fast but unstable system. The best results are obtained by combining both methods (Jones 1977) and this leads us to think that

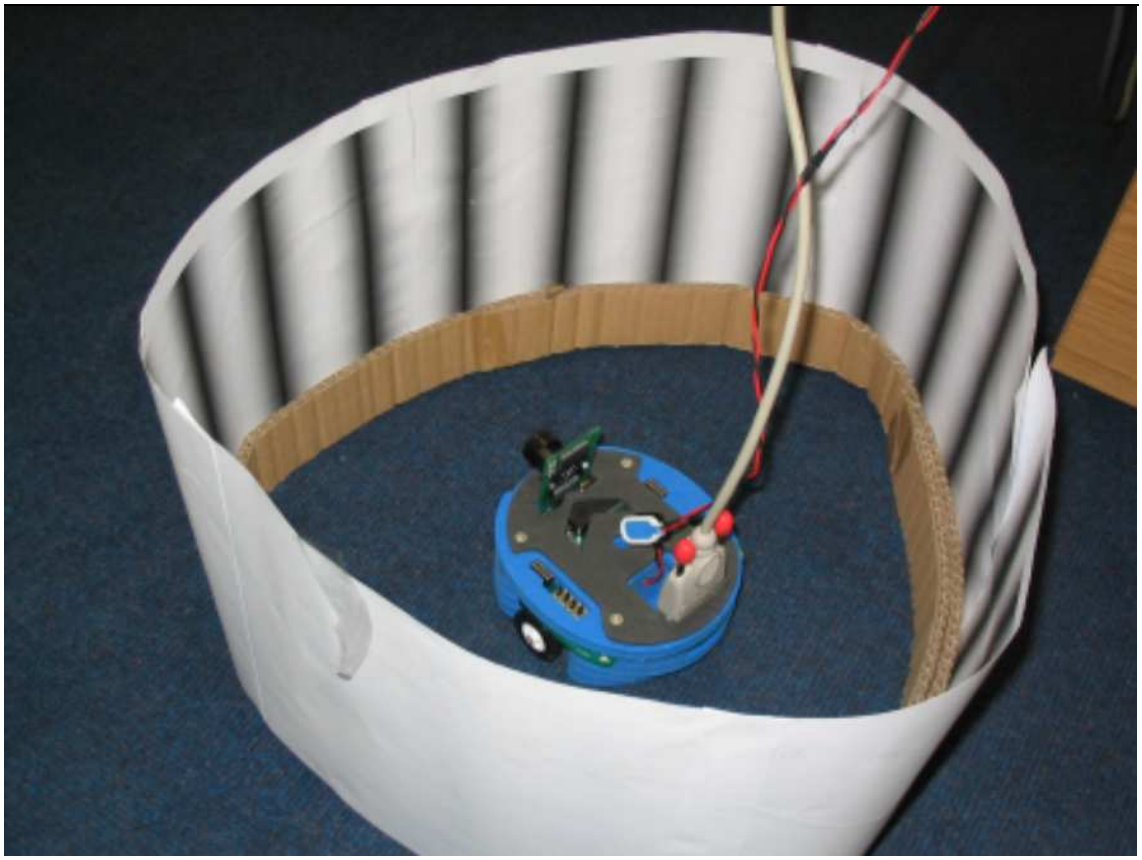


Figure 3.5: When we make the arena turn, the robot turns in the same direction, at the same time. Like this, the robot stays relatively to the arena at the same place.

we should do the same. The resulting architecture that we propose is simpler than the sum of the both because a lot of operations are shared between the two methods (see Fig. 3.6). The detection of velocity can be seen as a way to anticipate the position of the target, and the detection of position as a way to correct errors and drifts.

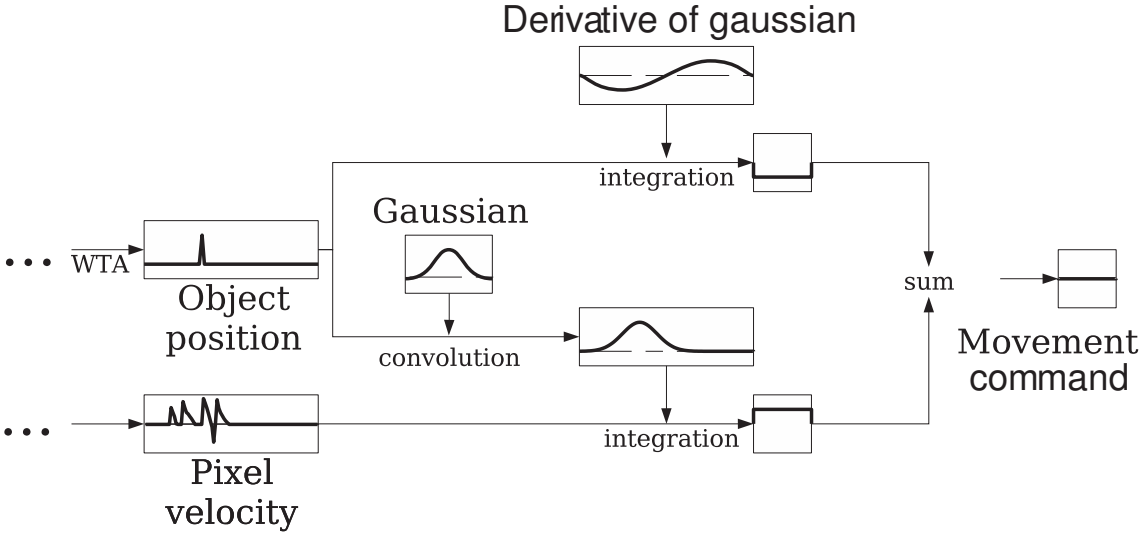


Figure 3.6: Architecture that combines the method 1 and 3 to take advantage of both. The left part of the figure has to be linked to the Fig. 3.2 and Fig. 3.3.

3.4 Conclusion

We have presented different methods which allow us to increase the level of interaction (synchronization-imitation) by imitating biologically plausible processes. These processes are both simple and easy to implement. If we want to synchronize a dance, velocity detection is very useful, while the detection of position is more useful when following a moving target. If we consider different kinds of imitation we can see the position as the goal of imitation and the movement (velocity) as the means of imitation. In fact using combined methods, imitation is done in order to reach the same position observed, but this is achieved by using the same velocity as the perceived one.

We have also proposed to use detection of velocity for tracking a target by anticipating its position (see Fig. 3.6). We can imagine a robot learning by association the relationship between detection of velocity and position to progressively antici-

pate position of the target from its perceived velocity. Studies such as (Hofsten and Rosander 1996) and (Richards and Holley 1999) investigate how babies develop the capacity of smooth tracking with the same kind of protocol and they have shown that babies progressively develop a better coordination between the movement of the eyes and the head. Our work can be extended to reproduce this phenomenon with robots.

In order to continue this investigation into synchronization and imitation further, we need to solve several problems common to most robotic studies. One of these requires a robot to learn to anticipate and therefore neglect changes in sensations which are produced by its own actions (e.g. perception of movement because the robot itself is moving). If its learning is done during exploratory phases, how can the robot choose to explore its environment or not ? The environment can sometimes be hostile and the robot should only explore when the conditions are ideal for exploring without risks. Even if the robot is capable of synchronization, how can it know that it should be synchronized or not ? It should not simply blind copy and the transitions between the behaviors should be smooth. In the next chapters (Chap. 4, and Chap. 5), we will propose solutions for solving these problems, basing our work on affect allowing robots to adapt their behaviors to the context. We start with the development of affective bonds.

Chapter 4

Affective Bonds

"They rear a very large number of chicks by an amazing device. For the hens do not sit on the eggs. Instead they keep a great number of eggs warm with even heat and so hatch them. As soon as the chicks come out of the eggs, they follow the men and recognize them as if they were their mothers". — St Thomas More *Utopia* (1516)

In the introduction, we have presented why imitation is important for autonomous agents in order to survive, to learn, and to socialize. In the previous chapter we have presented a way to improve synchronization during imitation, however, whatever the quality (in terms of accuracy and relevance of actions) of imitation, an autonomous agent should not imitate all the time; other behaviors are very important like exploring the environment or exploiting what it has already discovered. In this context, a successful imitation will not necessarily be very accurate or sophisticated but will be done at the "right" moment (appropriate context) with the "right" partner. To be efficient, imitative behaviors have to be integrated and balanced with other behaviors. Better understanding of the process of balancing behaviors in animals and

humans can greatly improve the treatment of pathological problems, but it can also improve the functionalities of autonomous robots.

The general efficiency of a robot can be improved not only in terms of autonomy and learning, but also in terms of human-robot interactions (Cañamero, Blanchard and Nadel 2006). If robots are to be truly integrated in humans' everyday environment, they cannot be simply (pre-)designed and directly taken "off the shelf" and embedded into a real-life setting. Too much autonomy in social robots might also carry risks if they behave "selfishly" and are detached for their human users, especially if the users are not used yet to the robots. A balance depending on the partner and progressive autonomy of the robots seems to be needed.

In 1958, Bowlby and Harlow (Harlow 1958) introduced the attachment theory, where they respectively showed that infants and rhesus monkeys modify their behaviors—especially play and exploration—depending on the relationships they have with the persons or even the objects in presence. This effect is so strong that later in the 1960s, Ainsworth (Ainsworth 1969) observed the differences of infants' behaviors in a scenario called "strange situation" (when the mother leaves unexpectedly) in order to diagnose the relationship between a child and his caregiver (usually the mother). Studies like those from (Hatfield et al. 1994) show that there is a strong correlation between the affective bonds with someone and the probability to imitate and to be synchronized with him/her; the symmetrical phenomenon has been observed as well: affective bonds change depending on the ease and frequency of imitation and synchronization. These phenomena are very interesting from the point of view of balancing behaviors in autonomous agents, therefore we studied how we can produce a Per-Ac architecture to model attachment and create affective bonds. In order to bootstrap the affective bonds, we propose to first model a simpler

phenomenon of attachment which is the "imprinting" phenomenon and we will see that this phenomenon is not only interesting in terms of attachment but also in terms of learning.

Animals and robots without any experience (when they are just born or just switched on) in an open environment can not evaluate their situation—they have nothing to compare with, or to use as references. Luckily most of the time, animals and robots start their "lives" in a safe situation. Many animals give birth birth in safe places (nests or dens protected from predators, warmness and food provided by parents ...) and users naturally switch on robots far from dangers (water, stairs, extreme temperatures ...). Therefore a simple but safe and useful behavior is to stay in the same situation they have started with.

Interestingly, in the 1930's Lorenz showed that birds follow the first thing they saw and called this phenomenon *imprinting* (Lorenz 1935). Animals (especially birds) form special attachment bonds with objects to which they are exposed very early in life. Usually young animals are imprinted to the mother—usually, the first thing they see—but can also be imprinted to other objects ("unnatural" imprinting, see Fig. 4.1 on the left). According to our approach we interpret this as the fact that they try to "keep" the first sensation they had and we have developed an architecture to make our robots reproduce this phenomenon like in Fig. 4.1 on the right).

The imprinting phenomenon was for a long time considered to be instantaneous and irreversible, as the term "imprinting" suggests. In the mid 1930's, ethologist Konrad Lorenz made this phenomenon well-known through his studies of greylag geese. Lorenz raised these animals from hatching, becoming the imprinting—parent-like—object for them. This "unnatural" imprinting to an individual of a very dif-

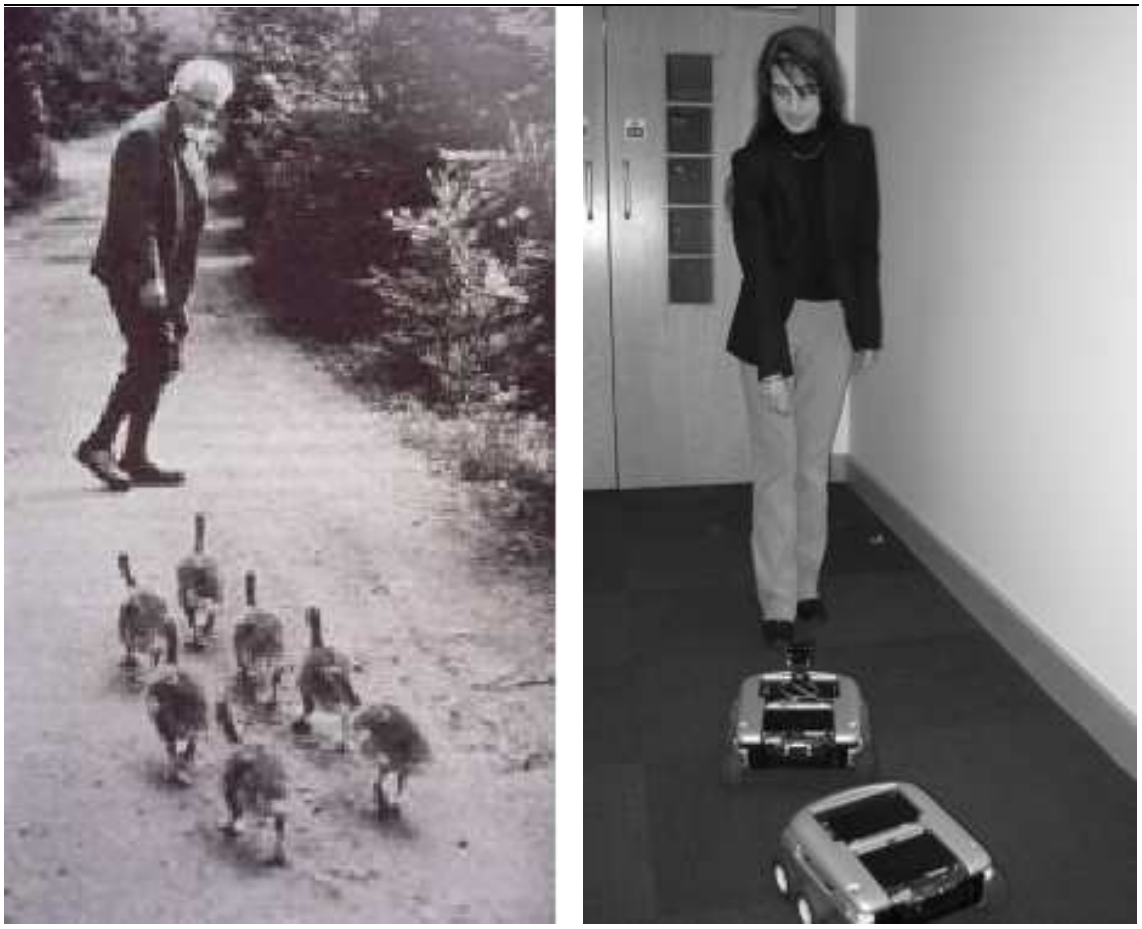


Figure 4.1: On the left Lorenz followed by its imprinted geese, on the right Lola Cañamero followed by our imprinted robots.

ferent species initially suggested that the animals had become attached to the first "eye-catching" object they had perceived immediately after hatching. This form of perceptual learning was also considered to be very different from (and unrelated to) other types of learning arising later in life, such as conditioning or associative learning. Such view has been more recently questioned as over-simplistic. Bateson (Bateson and Barron 2000), for example, postulates a model in which imprinting is not an instantaneous and irreversible process but a much more flexible and less

peculiar phenomenon. The main points of this view can be summarized as:

- Imprinting does not necessarily occur immediately after birth but has a more flexible sensitive period (Bateson and Martin 2000) affected by both experience and species-specific features. This provides some flexibility regarding the exact point in time in which the mother is first "perceived" and imprinted.

- Imprinting is not a monolithic capability but is composed of several linked processes (Bateson and Barron 2000): (1) "analysis" or detection of a "relevant" stimulus guided by predispositions of what the animal will find attractive; (2) recognition of what is familiar and what is novel in that stimulus, which involves a comparison between what has already been experienced and the current input; and (3) control of the motor patterns involved in imprinting behavior.

- Although imprinting can be functionally distinguished from learning involving external reward, both types of learning are deeply connected, as suggested by the possibility of transfer of training after imprinting.

We present a novel Perception-Action architecture and experiments to simulate imprinting phenomenon in a robot following this latter approach. Starting with a basic architecture that simulates imprinting in the more traditional sense (Sec. 4.1), we incrementally modify and extend this architecture to achieve further adaptation, also integrating reward-based learning (Sec. 4.2). This adaptation is achieved in the context of a history of "affective" interactions between the robot and a human, driven by "well-being" responses in the robot. In the following of this dissertation, we call "caretaker" the interactive partner of the robot. The caretaker interacts with the robot and can induce low or high well-being on the robot.

4.1 Robotic Architecture for Imprinting

The architecture we have used to implement imprinting follows a "Perception-Action" approach rooted both in psychology (Prinz 1997) and in robotics (Gaussier and Zrehen 1995), and that we have already successfully applied to synchronization of movements in robots (see chapter 3). This approach postulates that perception and action are tightly coupled and coded at the same level.Action is thus executed as a "side-effect" of wanting to fit the desired sensation with the actual sensation. Actions that allow to correct different perceptual errors are selected on the grounds of sensorimotor associations that can be "hardcoded" by the designer (e.g., with static coefficients, as it is our case) or learned from experience by the robot—see e.g., (Andry et al. 2003) or (Dearden and Demiris 2005) for examples applied to robots imitating hands' movements. We have used this general approach to model imprinting as an attempt to make the robot act in order to reduce the difference between the *current sensation* (S) and a *desired sensation* ($\overline{S}$) using sensory-motor associations.

Learning the Desired Sensation

Intuitively, the most obvious way to implement imprinting in a robot would be to have it learn the first sensation that it has when it is switched on (the equivalent of "hatching" in birds) as being his desired sensation—the sensation it will memorize and try to maintain after imprinting. This could be implemented:

$$\overline{S_t} = S_0 \tag{4.1}$$

where $\overline{S}$ is the desired sensation (the "goal" values for all sensor readings), t is the time elapsed from "hatching" , S is the current sensation (the current values for all sensors). This corresponds to the view of imprinting as "stamping" or developing instantaneous and irreversible affiliative bonds with the first "eye-catching" stimulus. This approach helps to guarantee that the desired sensation of the robot will be reachable, since it corresponds to a sensation that has already been reached at least once. On the contrary, using some sort of predispositions to code a-priori a desired sensation does not guarantee this sensation is reachable or can exist.

However, if by accident the first sensation is noise (quite common when we start a robot) or in the case of birds, the mother is not present at the hatching time, the first sensation is not relevant. To improve the robustness of the process we propose that the robot memorizes its average sensation from the beginning of its life (see (4.2)).

$$\overline{S}_t = \frac{S_0 + \ldots + S_t}{t+1} \tag{4.2}$$

At the beginning, when the robot has few experiences, the average sensation will be almost equal to its current sensation, but with time, experiences will accumulate in its memory and the influence of the current sensation on the desired sensation will decrease. To implement this equation, the agent needs to store all the sensations at all the time steps which is virtually impossible and moreover it is not biologically plausible. However, we show how it can be equivalent to use an incremental rule (4.3) similar to the learning rule of Rescorla and Wagner (Rescorla and Wagner 1972) used for conditioning.

$$\overline{S}_t = \frac{S_0 + \ldots + S_{t-1} + S_t}{t+1}$$

$$
\begin{aligned}
&= \frac{\frac{S_0+\ldots+S_{t-1}}{t} \times t + S_t}{t+1} \\
&= \frac{\overline{S}_{t-1} \times t + S_t}{t+1} \\
&= \frac{\overline{S}_{t-1}(t+1) - \overline{S}_{t-1} + S_t}{t+1} \\
&= \overline{S}_{t-1} + \frac{1}{t+1}\left(S_t - \overline{S}_{t-1}\right) \\
&= \overline{S}_{t-1} + \eta_t\left(S_t - \overline{S}_{t-1}\right) \qquad (4.3)
\end{aligned}
$$

The learning rate $\eta_t = \frac{1}{\widetilde{T}_t}$ and in this case we only need a variable increasing with the time ($\widetilde{T}_t = \widetilde{T}_{t-1} + 1; \widetilde{T}_0 = 1$) and a variable memorizing the current average sensation ($\overline{S}$). The complexity of the calculation is very low and biologically plausible. The learning rate at "hatching" or imprinting time is 1; therefore, learning

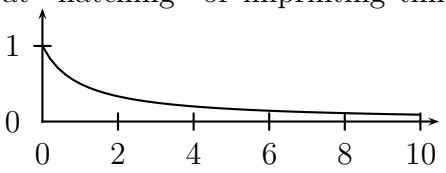


Figure 4.2: Decrement of the learning rate (y-axis) as a function of time (x-axis). At the beginning, a learning rate of 1 means that the desired sensation is equal to the current sensation.

is instantaneous at that moment and the desired sensation is equal to the initial sensation. Our first architecture used to model imprinting is described Fig. 4.3.

4.1.1 Experiments

Apparatus

We have implemented and tested our architecture using a Koala robot (www.k-team.com). Only the ring of infrared proximity sensors located around the robot was used to provide sensation input in these experiments. The average of all the

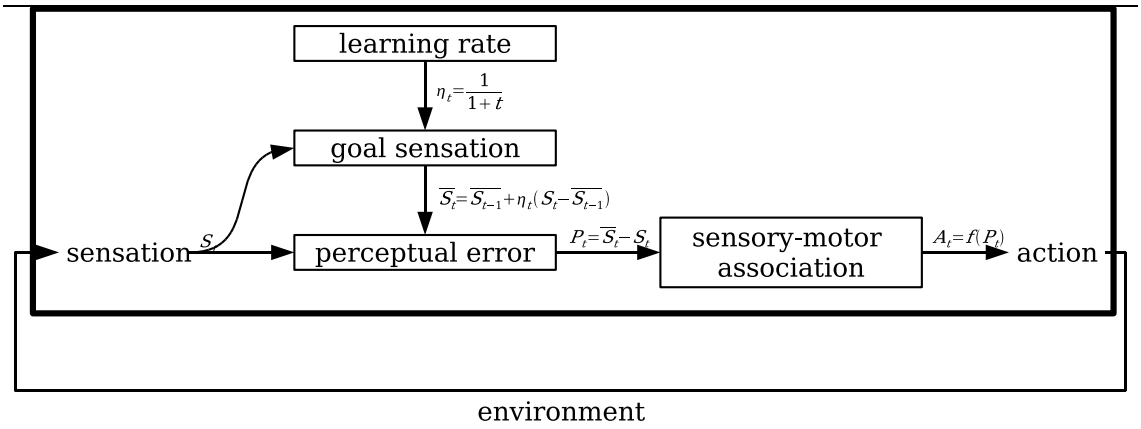


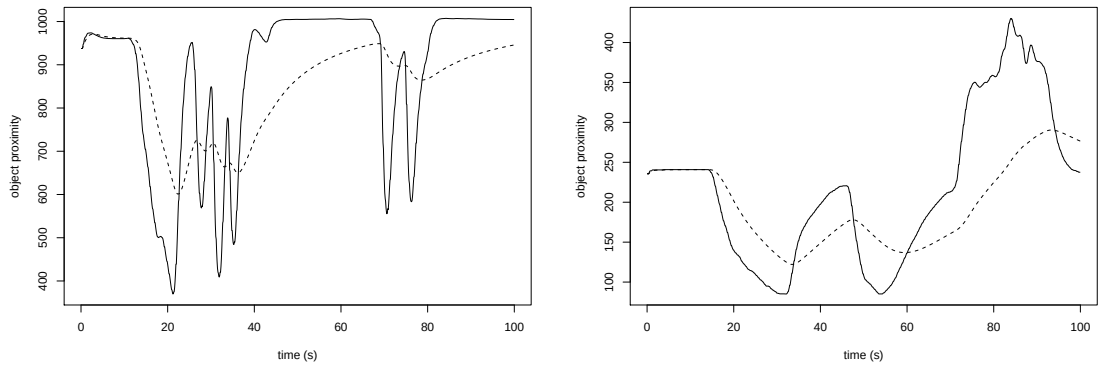
Figure 4.3: General architecture used to model imprinting.

infrared front sensors was used to detect (the proximity of) objects at the front of the robot—we will refer to this averaged reading as "the proximity sensor". Distance to frontal stimuli is the only modality used to form the desired sensation—i.e., for imprinting. The only actions of the robot after "hatching" (and therefore imprinting) are forward and backward movements as side-effects of its attempts to achieve the desired sensation acquired at imprinting time. As a consequence of this, the robot "approaches", "follows" or "avoids" (reverses if approached at a distance smaller than the imprinted distance) the frontal object as this moves around. We used two types of imprinting stimuli: near objects (high activity of the proximity sensors) and distant objects (lower activity of the proximity sensors). Two types of objects—a human and a cardboard box moved by a human, as shown in Figure 4.4—were used as near and distant stimuli. Although the experiments worked very satisfactorily with different types of objects, only the results obtained with the cardboard box were used for analysis purposes due to their higher clarity.

Figure 4.4: Experimental setting. In this case, a box located close to the robot is used as imprinting object.

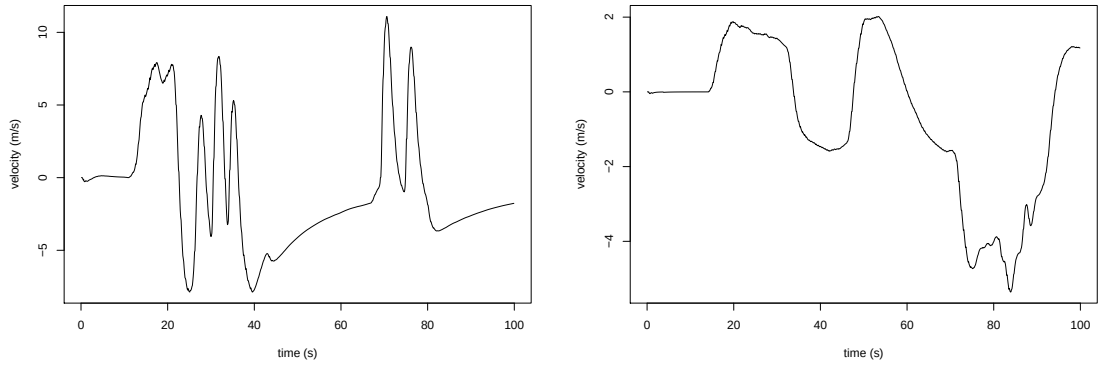
Results and Discussion

Using the box, 10 tests were run for each "hatching" condition—near or distant imprinting object. Figure 4.5 shows one representative example of each condition, with graphs on the left side of the figure (a1 and b1) corresponding to the "near hatching" case, those on the right (a2 and b2) to the "distant" one. Top graphs (a1 and a2) show current (solid line) and desired (dashed line) sensations, bottom graphs (b1 and b2) show the speed of the robot responding to the random movement of the box. In both conditions, the desired sensation (dashed lines in a1 and a2) fluctuates at the beginning and becomes more stable with time in both cases, even though the imprinting stimulus moves at different distances at the front of the robot. As a consequence of homeostatic control, the velocity of the robot (graphs b1 and b2) changes in order to decrease the difference between desired and current



a1

a2



b1

b2

Figure 4.5: Results of two experiments testing imprinting to near (left graphs) and to distant (right graphs) stimuli. The y-axis shows averaged readings of the proximity sensors on the top graphs and the speed at which the robot moves to correct the perceptual error on the bottom graphs, the x-axis shows time from "hatching" in all graphs.

sensations. Motor speed is directly proportional to (a fraction of) the magnitude of the difference.

We have done a variant of this experiment where we have implemented this ar-

chitecture on both sides of the robot, using the left front sensor and the left wheel on one architecture, and the right front sensor and the right wheel on the other architecture. The two architectures are not interconnected and are exactly identical to the one described Fig. 4.3. Without doing anything else the robot is able to turn in order to better follow or avoid the caretaker, in two dimensional space.

We see how the robot learns the imprinting stimulus using a very simple function. Such learning can take place even when the imprinting object (i.e., an object detected at a particular distance within the range of the infrared sensors) is absent at "hatching" time, although learning becomes more slow and difficult with time, corresponding to the limited time window during which the imprinting process is possible in animals. However, this model still implements the simple view of imprinting as "stamping" a permanent and irremovable trace, while, as Bateson points out, "the process is not so rigidly timed and may indeed be undone" (Bateson and Barron 2000). It also disregards the connection between imprinting and other types of (reward-based) learning. For an autonomous robot living in a changing and social environment, being able to modify or undo what was learned during imprinting is also very important, since (a) it is virtually impossible for the designer to define a priori a time window for the imprinting process that works in all possible environmental conditions, and (b) if the environment (including the social partner) changes, the robot has to adapt to the new features.

4.2 From Imprinting to Adaptation

In algorithms employed in autonomous robots and neural networks research, it is very common to use a learning rate that decreases with time in order to achieve a good level of stability in memory that consolidates learning. The learning rate must vary with time since, if it were constant, everything that is learned would be replaced by new events, memory contents would change constantly. However, learning should change not only as a function of time but also of the *relevance* of the stimulus. The problem now is thus how to make the robot assess what is relevant.

4.2.1 Assessing Relevance

To assess the relevance of external stimuli, we use the notion of well-being: since under normal circumstances, the evolutionary advantage of becoming attached to a caretaker is to foster security, beneficial interactions with the environment, and generally well-being, stimuli that carry some rewards associated with them are thus those stimuli relevant to become attached to. Drawing on Ashby's view of survival as viability (Ashby 1952) or stability of the internal environment, in our robot well-being is related to the values of its internal homeostatic variables in a range of ideal values, following (Cañamero 1997). Closely related architectures have used a similar notion of well-being (also termed "comfort" or "satisfaction" in those architectures) and "discomfort" to assess and compare the performance of different behavior selection policies in autonomous robots (Avila-García and Cañamero 2004), and to learn affordances through the interactions of a robot with objects in the environment (Cos-Aguilera, Cañamero and Hayes 2003). There are different ways to calculate well-being when the internal environment consists of several internal homeostatic

variables, such as the inverse of the average of the errors (deviations between the actual value and the ideal value or setpoint) of all the variables, the variance, etc—see e.g., (Avila-García and Cañamero 2002) for a presentation and discussion of different metrics. In our model, well-being corresponds to the endogenous factor of comfort described by Dunn (Dunn 1977); it is a measure (taking values between 0 and 1) of the viability of its internal state, i.e., the distance of the variables composing the internal state of the agent from their ideal values. A high level of well-being corresponds to a zone of good viability in its physiological space, as depicted in Fig. 4.6.

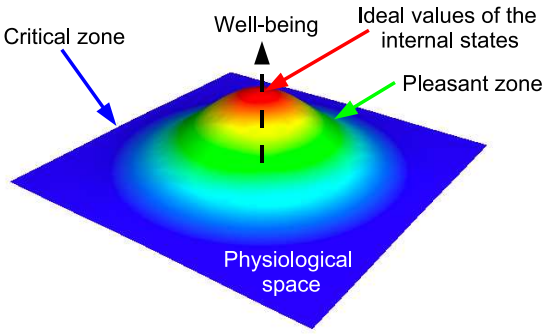


Figure 4.6: Well-being as a function of the distance of the internal states (here 2 are represented) to the ideal values.

A simple way of calculating well-being (Wb) at each point in time t given n variables, by taking the inverse of the sum of the deficits (distances $d_1, \ldots, d_n$, between the real values of the internal states and the ideal values), is:

$$Wb(t) = \frac{1}{1 + d_1(t) + \ldots + d_n(t)} \tag{4.4}$$

Later, we will use tactile contact as a source of well-being. We will thus try to make our robot learn to remember the stimulus (the sensation) that gives it most well-

being. To do that, we modulate the learning rate with the well-being but we will not just weight the learning rate with the well-being (it will NOT be: $\eta_t = \frac{k.Wb}{\tilde{T}_t}$ with k, a static parameter) because it does not show the interesting property of instantaneous learning at "hatching" time (i.e. $\eta_0 = Wb_0$ and not 1) if the well-being is not equal to 1 at the "hatching" time. Moreover, making the learning rate depend on the well-being can present disadvantages because it is very difficult to know in advance the average well-being of the robot and the ideal value of k is impossible to determine. If the environment is "difficult" or "hostile" (producing very low levels of well-being) and k is small, adaptation will be very slow, but learning will be unstable if the environment is highly "positive" (i.e., providing very high levels of well-being) and k is large. The strong influence of the well-being can thus be problematic because this level has to be chosen depending on the hostility of the environment, and neither the robot nor the designer has this information in advance.

The method that we propose does not present this problem, since the learning rate is not modulated by the absolute value of well-being but by its relative value We propose to define the learning rate as the ratio between the value of the current well-being and the accumulation of the value of the well-being at all the time steps already past:

$$\eta_t = \frac{Wb_t}{\widetilde{Wb}_t} \tag{4.5}$$

with $\widetilde{Wb}_t = \widetilde{Wb}_{t-1} + Wb_t$ and $\widetilde{Wb}_0 = Wb_0$. This allows the robot to learn to adapt its desired sensation depending on what is considered the best at that moment, and as a result of this learning, the robot will adapt its interactions to different environments or interaction styles of caretakers. This adaptation is not something

that only takes place "then and there", but it also depends on the history of the interaction. When the well-being is constant, the learning rate decrease with time exactly in the same way as learning in the simple imprinting algorithm, but now it can also depend on the relevance of the stimulus as measured by the well-being provided to the robot. The more well-being a stimulus provides, the faster the robot will develop an attachment link to it and the stronger this link will be—i.e., the more relevant the stimulus will be for imprinting.

We can now reproduce the imprinting phenomenon described in Sec. 4.1, this time taking into account the relevance that the observed stimulus has for imprinting, since the well-being produced by some stimuli (e.g., a caretaker stroking the robot) amplifies the memorization of these relevant stimuli over non-relevant ones (e.g., a static wall). As we will see, in addition to the homogeneity and simplicity of the equations, using this algorithm presents some other advantages for learning.

4.2.2 Multiple Time Scales

With the function described in Eq. (4.5), after some time interacting with the environment learning becomes very slow, as $\widetilde{Wb}$ becomes very large. Intuitively, this would correspond to a situation in which the robot has formed an attachment bond with the caretaker but cannot learn anything else. However, robots like animals have to learn new things while interacting with their environment and they have to learn which of these things are "beneficial" for them while remembering what was learned during imprinting. We are thus facing the problem of how imprinting relates to later forms of learning. A possibility is to consider further learning as a completely different process that starts once imprinting has finished and for which we can use a learning rate that depends solely on the well-being—e.g., as in (Cos-Aguilera et

al. 2003)—but not on the time from "hatching". However, this would erase useful memories and produce conflicts with the imprinting phenomenon.

As Bateson points out (Bateson and Barron 2000), imprinting should not be regarded as an irreversible process that was completed once and for all when the appropriate "time window" closes to the world. Even if learning about the features of the imprinting object becomes more difficult after the "sensitive period" (Bateson and Martin 2000), the effects of imprinting are not irreversible. Moreover, he adds that although imprinting can be functionally distinguished from learning involving external reward, both types of learning are deeply connected, as suggested by the possibility of transfer of training after imprinting. Therefore, to make these different types of learning compatible, we can consider them as related processes to learn what is relevant (beneficial) for the individual at different time scales. For example, learning to be with the caretaker serves a goal that is beneficial in the long term, whereas learning about the usefulness of an object to satisfy an urgent need serves an immediate goal. Instead of learning a single desired sensation that the robot will try to achieve or maintain through its interactions with the environment, it could thus simultaneously learn different desired sensations that it will try to reach depending on the time scale used to remember (seconds, hours, days, etc). We will see later how these multiple desired sensations can be selected. We present now our method to make robots learn different desired sensations at different time scales.

From the perspective of learning, this implies trying to reconcile imprinting and reward-based learning, and this presents problems such as conflicting requirements regarding the learning rates needed for each process. Our approach thus differs from reinforcement learning algorithms such as Q-learning and TD-learning since it deals with several learning rates and makes a selective use of memory—only the "best"

sensation related to each time scale is kept. To provide a common framework for imprinting and reward-based (in our case well-being based) learning, we have to reconcile the following ideas:

- At the beginning (i.e., during the imprinting process) we want the learning rate to decrease with time to consolidate memory and "protect" what was learned about the imprinting object.
- It is useful to continue learning new things. Since we don't know in advance which is the best learning rate for each particular case, it might be useful to remember desired sensations at different time scales. However, this process cannot work at the beginning (during the imprinting process) since the robot has not accumulated enough experiences. Also the learning rate needed (closer to a constant rate) seems in conflict with the decreasing learning rate above.

To keep a continuum and avoid conflict between the two parts of the needed process, we keep the shape and therefore the interesting properties of the decay of the learning rate, presented in (4.5), but we amplify or at the opposite we attenuate it. To do so, we apply the power function to the previous calculation of learning rate, with a exponent depending on the time scale of the desired sensation learned. Each desired sensation S^γ at the time scale γ is defined by the value γ of the exponent of the power function, which can take values between 0 and $+\infty$:

$$\overline{S_t^\gamma} = \overline{S_{t-1}^\gamma} + \eta_t^\gamma (S_t - \overline{S_{t-1}^\gamma}) \text{ and } \eta_t^\gamma = \left(\frac{Wb_t}{\widetilde{Wb_t}} \right)^\gamma \tag{4.6}$$

If γ tends to ∞, the learning rate tends to 0—after "hatching", there is no further adaptation of what has been learned about the imprinting object. If γ tends to 0,

the learning rate tends to 1—there is no stability and the desired sensation tends to correspond to the current sensation. Between these two extremes, we have different intermediate learning modes available and when γ is equal to 1, we have exactly the same learning properties as previously. Examples are provided in Fig. 4.7, which shows the evolution of the learning rate (with a constant well-being) under three different time scales. Our robot is now able to memorize different desired sensations

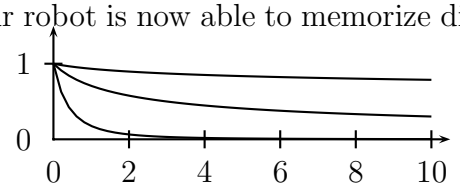


Figure 4.7: Evolution of the learning rate on three different time scales. The y-axis shows learning rate values, the x-axis time from "hatching". Parameter values defining the time scale of the learning rates are $\gamma_0 = 0.1$ for the top curve, $\gamma_1 = 0.5$ for the middle curve, and $\gamma_2 = 2.5$ for the bottom curve.

related to different time scales. Let us see how to select among them the sensation that it will actually try to reach.

4.2.3 Selecting the Time Scale

It is advantageous to be able to learn more than one sensation the robot should try to reach, some learned on the long term and more related to the familiarity and stability of the sensation and some other learned on the short term, more related to the immediate well-being associated to them. However, these different desired sensations may not be compatible, and the robot has to choose between them if we want to avoid the Buridan's ass paradox illustrated by the fact that a hungry and thirsty ass placed in between a source of food and and a source of drink will hesitate until it dies. We call the chosen sensation that the robot will finally try to reach, the *goal sensation*. This goal sensation should be based on desired sensations

promising high well-being even if they are unfamiliar—less secure—when the well-being of the robot is high as the robot can take risks, and based on familiar but less promising sensations when the well-being is low as the robot can not afford to take risks. Therefore, the goal sensation will be a desired sensation in a short time scale (γ_k small) when the well-being is high, and it will be a desired sensation in a long time scale (γ_k large)—of which the imprinting sensation is an example—when the well-being is very low. The use of a short time scale allows the robot to be very reactive to external changes, which in principle is advantageous for its survival, but on the other hand the lack of experience puts it in an "insecure" position that should be avoided when the well-being is already low (Avila-García and Cañamero 2004). Intuitively, when the robot "feels secure", it will have a more open stance towards the external world and will tend to "live in the present". On the contrary, in a situation of discomfort it will be more closed to the world and the present situation, to look back for past memories; observations of infants (Dunn 1977) and monkeys show similar behaviors (Drea 1998). The goal sensation is a combination of desired sensations learned at different time scales weighted by a "filter" in a gaussian shape where the position of the maximum value depends on the well-being (see Fig. 4.8).

4.2.4 Openness to the World

Our architecture now allows the robot to keep learning after the initial imprinting, and to be able to try to keep its "best" sensation. However, if the robot is continually trying to achieve its "best" sensation looking into its multiple time-scales memory (the multiple desired sensations), it will avoid any new sensation and therefore will not be able to learn from new experiences. This can be seen as an instance of the well known "exploitation/exploration" dilemma (Wilson 1996) in autonomous

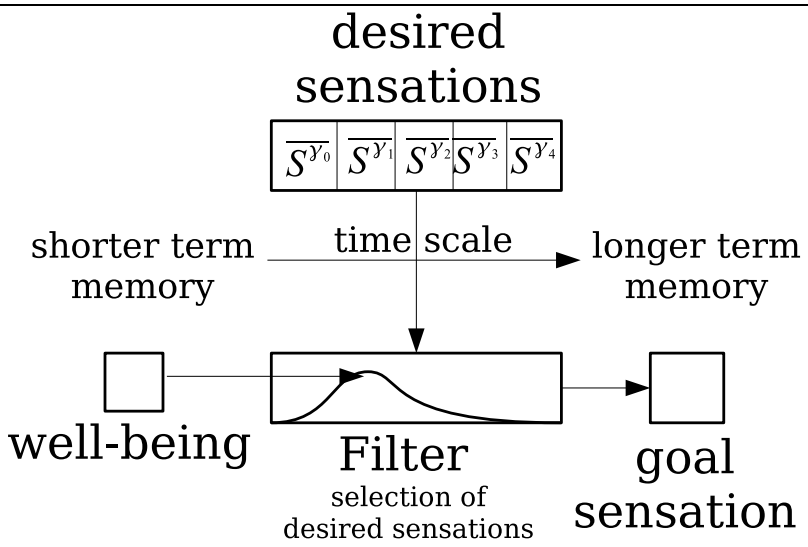


Figure 4.8: The final desired sensation is formed from desired sensations at different time scales by means of a filter that weights the contributions of these desired sensations. In this filter, the maximum value is defined by the value of the well-being.

learning, i.e., how to decide between using the knowledge already acquired in order to solve a problem, or continuing exploring to acquire new knowledge. We thus need a mechanism to solve this problem.

Well-being can also be used to provide such mechanism, since there is evidence that a good level of well-being or comfort—e.g, postural comfort (Kugiumutzakis et al. 2005)—facilitates learning in infants, and this also makes sense in our architecture. When the robot has a low level of well-being, it should look for the most familiar (i.e. secure) sensation it knows, but if it is not efficient and the well-being continues to decrease, it should stop rather than trying hopelessly to reach an impossible or not beneficial sensation; at least external actions can be done by a human to help it without the robot resisting. Conversely, in a situation of high well-being the robot will have no reason to change its current sensation but on the contrary,

should associate this new sensation as a desired sensation. A good strategy seems thus to let the current sensation change (i.e., to "pay attention" to new sensation) when the robot has a good level of well-being; this is achieved by inhibiting (modulation by "activity" in Fig. 4.9) its attempts to attain its desired sensation. On the contrary, when the well-being is low, the robot will try to actively reach memorized desired sensations, unless the well-being is really too low. The activity of the robot to changes, would thus have an inverted-U shape as a function of well-being allowing the robot to accept external stimulation in extreme situations. Figure 4.9 summarizes our global Perception-Action architecture described in this section, combining imprinting and reward-based (well-being based) adaptation.

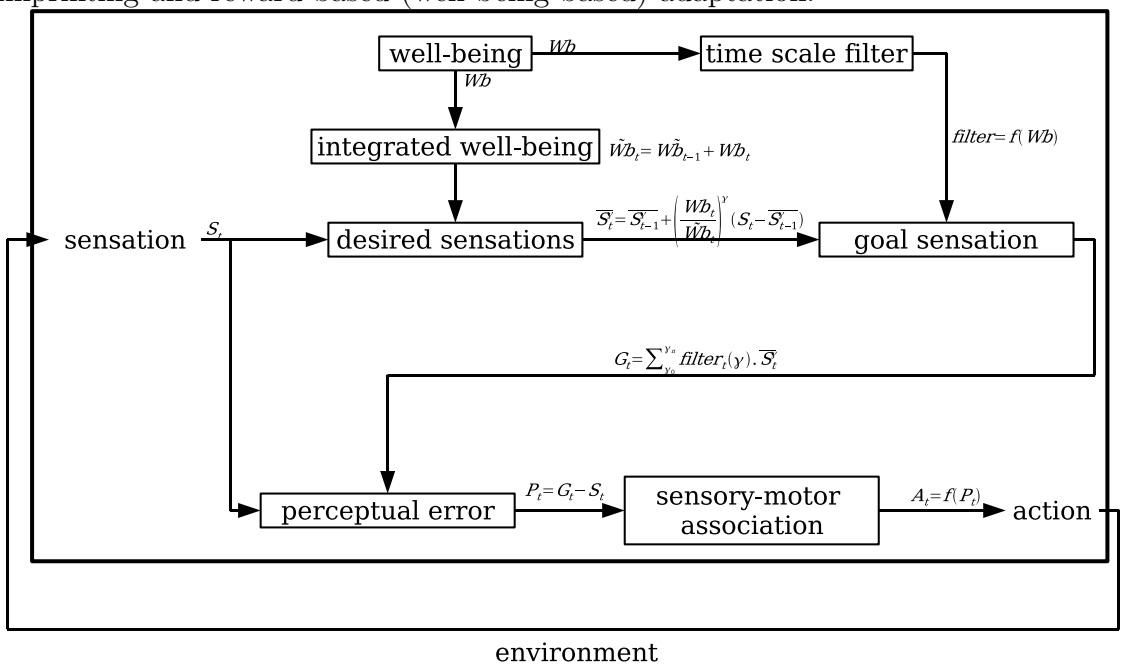


Figure 4.9: Global Perception-Action architecture for imprinting and adaptation.

4.2.5 Experiments

Apparatus

The setting of these experiments is very similar to the one presented in Section 4.1.1, but this time we need to add some new features to manage well-being. The robot receives reward increasing the well-being as a result of tactile contact on its leftmost infrared proximity sensor. We added to the architecture an internal homeostatic variable, "tactile contact" that the robot must keep close to an ideal value (as high as possible) and that decays with time in the absence of contact on the infrared sensor mentioned above. We adapt Eq. (4.4) to calculate the robot's well-being using this variable:

$$Wb = \frac{1}{1 + e^{-contact}} \tag{4.7}$$

The robot will try to keep Wb as high as possible given its present circumstances and the history of its interactions. To facilitate interaction with humans, the robot emits beeps with a frequency that depends on the level of "distress", i.e., the frequency of the beeps increases as the well-being decreases. This is akin to a "separation distress" response in animals, and is intended to "flag" the need for action on the part of the human—tactile contact that will increase well-being in the robot—see e.g., (Panksepp 1999)—for a discussion of the separation distress and comfort responses in animal).

Results

Figure 4.10 shows the results of one example of interaction with the robot among 10 similar tested interactions, with an architecture including 20 time-scales. We began

the interaction (the moment of "hatching") without any object at the front of the robot. The robot therefore starts with an imprinting situation in which there is no imprinting object—point 'a' at the top of Figure 4.10. Therefore, when we try to approach it (point 'b0') it moves backwards (marked as 'b1' in the bottom graph of the figure). We then increase its well-being (point 'c1' in the middle graph) by touching its side sensor and we observe (also in the middle graph) that the activity level decreases. When we approach the robot again (point 'c0') it does not reverse (avoid us) anymore, as we can observe in the "plateau" in the lower graph. We then remain close to the robot for some time, touching its sensor simultaneously in order to make it learn that in fact, and contrary to its initial experience, it is beneficial to have a stimulus in front of it. When this stimulus disappears, we also stop touching its side sensor; the well-being then starts to decrease while the activity level increases (d1), and the robot will select, as goal sensation, the long-term desired sensations (d0) and therefore it will try to reach them (e0): it will move forwards (d2) to try to find something at its front. When it finds it, it stops (e1). It is interesting to note a very stable shape (denoted by 'f') on a rather long time scale of the desired sensations graph. This means that, globally, the presence of something at the front of the robot is positive even if locally (on a short time scale) it is not always the case. In fact, continuing this experiment (approaching an "object" to the robot and giving it comfort) for a longer period, we would assist to a slow propagation of that stable shape (f) to the very long-term scales, eventually modifying the memory of the imprinting stimulus.

4.3 Conclusion

We have presented a Perception-Action architecture and experiments to simulate imprinting in a robot. Following recent theories about imprinting in animals, we do not consider imprinting as rigidly timed and irreversible but as a more flexible phenomenon that allows for further adaptation as a result of experience. Our architecture reconciles two types of learning traditionally considered as different, and even incompatible, due to apparently conflicting features and functions: the establishment of an initial attachment to a situation or a caretaker (an imprinting object) and reward-based learning as a result of experience, that we have grounded in the notion of internal well-being. Adaptation is achieved in the context of a history of interactions between the robot and a human, driven by variation of well-being in the robot (see video 'affective bonds.mp4' of the CD-ROM for an illustration of this chapter).

Our implementation is still simple and we have made simplifications but we would like to improve it in the future. Firstly, we only used one feature (distance to the perceived stimulus) to learn about the caretaker. Proper treatment of learning about the imprinting object would require considering multiple features that the robot would have to analyze in order to recognize the caretaker from different perspectives and in different situations. Secondly, at present the robot only stores a desired sensation per time scale in its memory. However, taking into account other contextual factors would necessitate learning and handling different desired sensations within each time scale. Third, desired sensations provide the robot with a mechanism to decide what it should reach, but further development would also require a mechanism to decide what it should avoid ("avoided sensations"), some-

thing like the basis of "fear" system. Finally, the robot can now actively try to reach known situations in order to stay in a stable environment, or passively accept new situations, but to be more autonomous, it should also be able to explore new situations by itself and express opportunism behaviors if appropriate. The balance between these different possible and necessary behaviors will be the subject of the next chapter, and we will show that it can lead to low level imitation depending on affect.

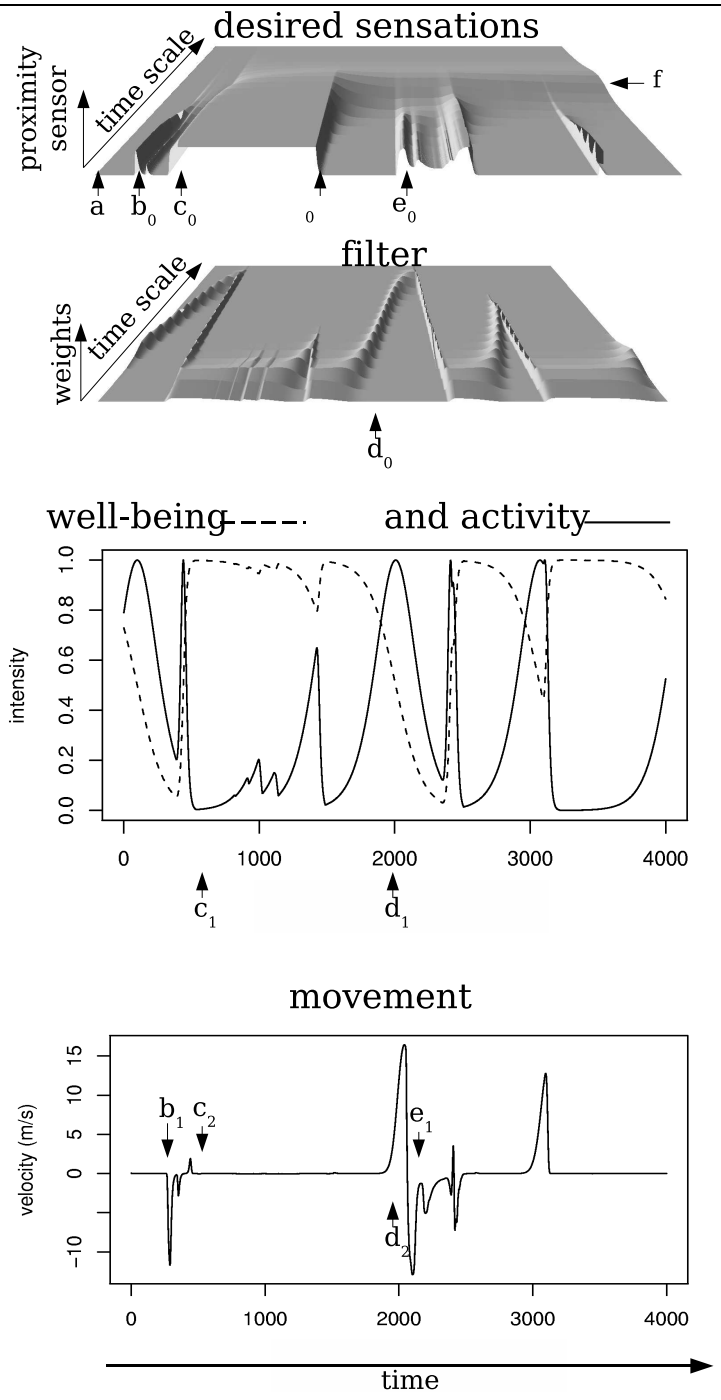


Figure 4.10: Evolution of the different internal states and movements of the robot during an interaction of about 2 minutes. 20 time scales have been used simultaneously. See text in the results part for explanations.

Chapter 5

From Balancing Behaviors to Low-level Imitation

In the previous chapter we have proposed a robotic architecture to develop affective bonds to specific situations by autonomously forming desired sensations. The robot was acting in order to reach these desired sensations and stay in familiar situations. However, in that architecture the robot was not able to discover any new situation in the absence of external stimulation, as it only tried to stay safe in the same situation. For autonomous agents (children, animals or robots), exploring the environment is essential as it allows them to learn new things and also offers opportunities to consolidate past experiences. Exploration is also a risky activity as it exposes the agent to unknown, potentially overwhelming or dangerous situations. Therefore, exploration must be done with care, and a trade-off should exist between activities such as seeking stability, exploring, imitating novel actions performed by another agent, or trying to take advantage of opportunities offered by new situations and events. Endowing a robot with this capability poses three main problems: (1) generating all

these behaviors from the same underlying architecture; (2) autonomously switching among them; and (3) achieving a good balance in the execution of these activities. In this chapter, we extend our robotic architecture to achieve these three goals. In addition, trying to solve the exploration, exploitation problem leads to low-level imitation as a side-effect.

Taking an incremental approach to design our architecture, we first recall the architecture presented in the previous chapter adopted here to seek stability to later add elements that progressively induce capabilities of exploration, exploitation, and finally low-level imitation. At each step, new behavioral capabilities are added while preserving the existing ones, and an important issue is to make the robot achieve a good balance among all its activities. Autonomously achieving an adaptive execution of activities can be seen as an instance of the behavior selection problem. In our case, however, changes in observable behavior are not achieved by "switching" among a set of discrete behaviors, but differential execution of activities relies on a modulatory mechanism based on notions of "well-being" that we already use in the previous chapter and "affect" that we will define. In our model, well-being depends on the internal (physiological) states of the agent, whereas affect depends on the values of its sensors (sensations). Well-being thus corresponds to the endogenous factor of comfort described by Dunn (Dunn 1977), whereas affect corresponds to its exogenous factor. We use the following definitions:

- The *well-being* of an agent is a measure (taking values between 0 and 1) of the viability of its internal state, i.e., the distance of the variables composing the internal state of the agent from their ideal values. A high level of well-being corresponds to a zone of good viability in its physiological space, as depicted in Fig. 5.1 (left).

- *Affect* in this model is the evaluation (expressed in values between 0 and 1) of the "goodness" or "safety" of a situation based on the familiarity (in terms of frequency) and the past well-being (pleasantness) of the associated sensation. A high level of affect corresponds to the fact that the agent evaluates the situation in the world as highly safe; as depicted in Fig. 5.1 (right), this can be represented in the agent's sensory space as a function of the mismatch between the actual sensation and the desired sensations to which the robot would tend to access in case of danger. Using affect, agents are able to evaluate how far they are from a safety zone that corresponds to a familiar zone or to a zone where they expect to maximize their well-being and therefore their life time (Likhachev and Arkin 2000).

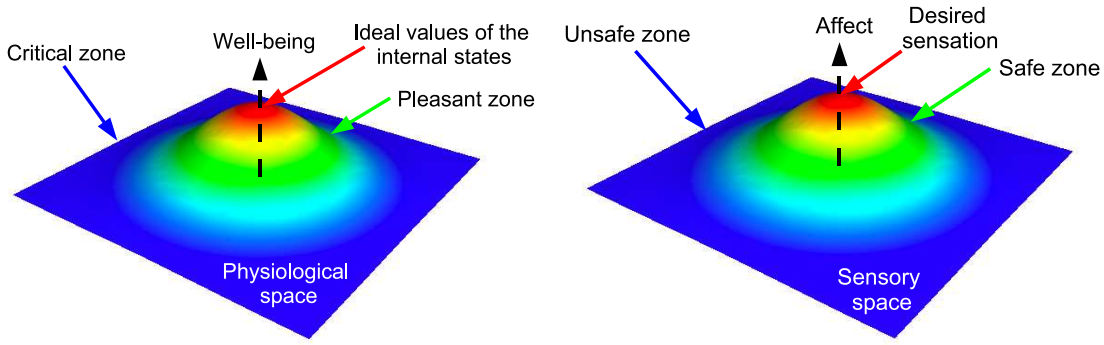


Figure 5.1: The two components of the robot's comfort: well-being can be represented in the physiological space (left) and affect in the sensory space (right).

As we will show in the remainder of the chapter, modulation of the architecture based on well-being and affect achieves adapted execution of different activities. This makes the robot explore if nothing happens, take advantage of opportunities in the presence of novelty while avoiding danger, imitate another agent that is performing a novel action, or return to familiar situations ("safe zones") if nothing

happens but the well-being is low.

In Sec. 5.1 we present a system that we add on the previous architecture to make the robot progressively explore its world. Then in Sec. 5.2 we add another system to allow the robot take advantage of what it has discovered. Before concluding in Sec. 5.4 we show in Sec. 5.3 how a side effect of the previous systems produces low-level imitation moderated by affect.

5.1 Exploration

Reproducing the imprinting phenomenon and adding the possibility of adaptation allows the robot to memorize which are the sensations it should try to reach in order to have good stability and well-being. However, in the absence of external stimulation, the robot described in the previous chapter will stay all the time in the same situation as it will never have the opportunity to encounter other sensations—it will never experiment with new situations. On the contrary, animals often look for novelty (Panksepp 1999, Power 2000) and, as already pointed out by (Oudeyer and Kaplan 2004, Kaplan and Oudeyer 2004, Steels 2004), it would be very beneficial for robots to look for novelty—in our case, unfamiliar sensations. Obtaining new sensations can however be dangerous and not always be useful as too much novelty does not produce efficient learning (Kaplan and Oudeyer 2007). Moreover, Dunn (Dunn 1977) observed that children explore more when they are in a familiar environment; Likhachev and Arkin (Likhachev and Arkin 2000) use the zone of comfort in order to modulate exploration in robots.

To generate spontaneous exploration, in (Blanchard and Cañamero 2006) we proposed to increase the effect of an exploratory behavior while the robot does

not have any other specific motivation. Exploratory behavior can consist in the execution of different actions selected randomly (Andry et al. 2003, Demiris and Dearden 2005), or selected in order to lead to a maximum of learning (Kaplan and Oudeyer 2004), or as in our case a simple exploratory behavior of moving forward, represented by Be, a positive variable defining the positive speed of the robot for exploration. To add this new behavior without interfering with the previous behavior making the robot seek for desired sensations, we need to inhibit the exploratory behavior when the robot has specific motivations—in this case when the robot is seeking for desired sensations. Therefore, in order to modulate the exploratory behavior, we introduce a notion of *apathy* with a variable Ap representing the fact that the robot does not have any specific motivation. Finally, the action to explore (Ae) based on the exploratory behavior (Be) is amplified when Ap is large, and becomes null when Ap is small:

$$Ae_t = (Ae_{t-1} + Be_t)Ap_t \tag{5.1}$$

For many reasons that we will see later, the robot can have the motivation to stop, inverse or amplify the ongoing variation of sensation. We use a variable Mc to denote the motivation to continue or amplify these actions. Mc is negative when the robot tries to oppose itself to the ongoing actions, and positive when it tries to amplify them. The apathy (Ap) leading to exploration represents the case in which the robot has low motivation to amplify or oppose itself to the ongoing actions:

$$Ap_t^\gamma = e^{-r(Mc_t^\gamma)^2} \tag{5.2}$$

where r is a parameter defining the decay rate of apathy as a function of motivation.

The robot can now theoretically explore (move forward in order to experiment with non-desired or novel sensations) when it does not have any other motivation. However, the behavior described in the previous chapter, always carries the motivation to avoid new sensations and therefore the robot would always be opposed to the exploratory behavior. Our proposed solution to this problem is to apply the notion of *affect* presented earlier to dynamically modify the motivation to continue. Affect reflects the subjective (i.e. without reasoning or exact predictions) evaluation of the situation in term of exogenous comfort (i.e. familiarity and past good experiences with the situation). Affect measures the proximity (in the sensory space) of the agent to its desired sensations corresponding to the zone of exogenous comfort (or safety zone, see Fig. 5.1). The physical distances are not relevant to the agent, what is really important for the agent is the amount of action (moving far or not) it would have to do in order to reach its desired sensations. This appreciation of distance is given by Pa^γ representing the perceived actions needed to make the robot reach its desired sensation at the time scale k. The variable affect (Af^γ) varies from 1 (close to the desired sensation) to 0 (far to the desired sensation) and is defined in:

$$Af_t^\gamma = e^{-s(Pa_t^\gamma)^2} \tag{5.3}$$

where s is a parameter indicating the decay rate of the affect as a function of the perceived actions needed to reach the desired sensation of the time scale k.

We recall that when the affect is high, the robot can afford to increase its distance from the desired sensations, whereas it should try to decrease it when the affect is low. The motivation to continue (Mc^γ) which tends to amplify or stop the ongoing actions (perceived actions) must then be positive when the affect is high,

and negative when the affect is low. The threshold between what should be considered as "low" or "high" affect is subjective and we set this threshold using a static parameter q defining the characteristic behavioral profile (or in a very restricted sense the "personality") of the robot and q can be interpreted as the *timorousness*. If q is high, the robot will often oppose itself to the perceived actions and try to stay in familiar situations, whereas if q is low, the robot will more often try to continue and amplify the perceived actions (it will try to increase the distance to its desired sensations). This helps to define the degree of openness to the world (Op^γ) which is a variable for each time scale representing the amplification of any perceived actions for "curiosity":

$$Mc_t^\gamma = Op_t^\gamma = Af_t^\gamma - q \tag{5.4}$$

In this part of the architecture, we equate the motivation to continue to the openness to the world ($Mc^\gamma = Op^\gamma$); the architecture is similar to the one presented in the previous chapter but the motivation to continue is computed as in Fig. 5.2 instead of being constant and equal to -1 corresponding to a permanent opposition to novelty.

When we apply this architecture to a robot in a static environment, it does not move at the beginning but when the action to explore (Ae) increases, it starts to move forward; in this case, the robot is not in a familiar situation anymore (Af^γ is low), and therefore it starts to have the motivation to oppose itself to the ongoing actions ($Mc^\gamma = Op^\gamma < 0$). As the robot has motivation the level of apathy (Ap^γ) will become low and this will stop the actions to explore (Ae). It is interesting to note that, therefore, the robot moves by small steps; such"cautious approach" behavior is not only useful to control the level of unfamiliarity during exploration, but it also reproduces the approach of an animal to a new stimulus—it moves forward,

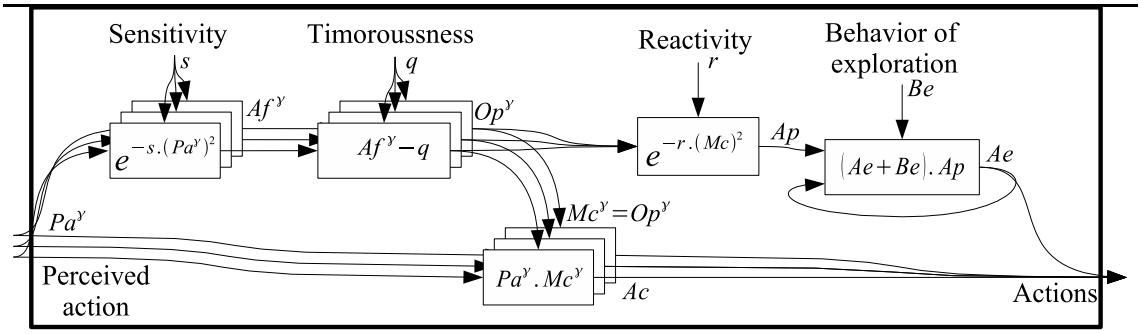


Figure 5.2: Actions are the result of spontaneous generation of actions for exploration (Ae) and actions (Ac) generated in order to amplify the perceived ongoing actions. The different layers correspond to different time scales. We take the average when they are grouped together.

stops, waits a bit, moves forward again and so on (see video:'animal avoidance behavior.mpg'). However, if the caretaker moves, the robot will perceive novelty and inhibit its exploratory behavior and only try to reach stability, i.e., retrieving its initial distance to the caretaker. Figure 5.3 shows typical positions of our Koala robot during this "cautious approach" exploratory behavior when tested with three different values of q : 0, 0.5 and 1, s set to 0.1, and r to 1 (see video: 'exploration and avoidance with negative affect.mov').

We can observe that the robot moves confidently (smoothly) when q is low, whereas it moves with hesitation (with some backwards movements) when q is high. In all cases, exploration is slower when the robot is farther away from its initial (and therefore familiar) situation (initial distance to the caretaker). The decay rate of exploration as a function of unfamiliarity depends on the parameter s and therefore can be changed.

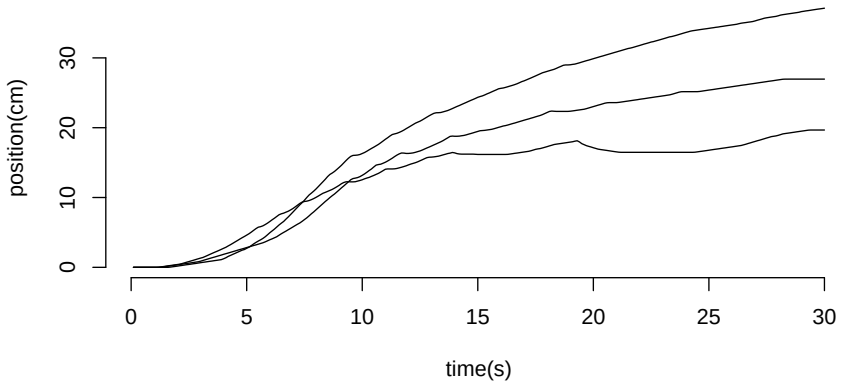


Figure 5.3: Successive positions of the robot during exploration for three different values of q (0, 0.5 and 1) for curves from top to bottom.

5.2 Exploitation and Interruption-Related Behaviors

We have shown that the robot can autonomously modulate two kinds of behaviors—seeking stability (previous chapter) and exploration—depending on its internal states and on external events. The first behavior makes the robot try to reach sensations known as familiar or pleasant, the second makes it explore new sensations when it is already in a familiar or desired situation. However, if during an action (executed either in order to reach a desired sensation or in order to explore), the well-being increases or decreases suddenly, the robot should interrupt its current behavior and increase or respectively decrease the effect of the perceived ongoing actions. In fact, if the robot accidentally moves close to a reward (stimulus increasing the well-being), it should continue its movement, showing *opportunism*. On the contrary, it should cancel or resist to this movement if it discovers a danger (stimulus decreasing the

well-being), therefore showing *avoidance* behavior. Affect is able to interrupt ongoing behavior, as could be done in animals or human using a motivational and emotional control (Simon 1967). The difficulty for the robot is to know if the variation of well-being is due to the recent changes or to long-term changes. Therefore, we use the past well-being ($\overline{Wb^\gamma}$) at different time scales to compare it with the actual well-being (Wb). In fact, we use the same learning rates as those used for the computation of desired sensations in order to estimate the well-being associated to each desired sensation:

$$\overline{Wb_t^\gamma} = \overline{Wb_{t-1}^\gamma} + \eta_t^\gamma (Wb_t - \overline{Wb_{t-1}^\gamma}) \tag{5.5}$$

We call *pleasure* (Pl^γ) the measure of the variation (between -1 and 1) of well-being at different time scales:

$$Pl_t^\gamma = Wb_t - \overline{Wb_t^\gamma} \tag{5.6}$$

We then use this notion of pleasure to compute the motivation to continue (Mc^γ), which now not only depends on the openness to the world (Op^γ) but also on the variation of well-being:

$$Mc_t^\gamma = Op_t^\gamma + Pl_t^\gamma \tag{5.7}$$

The new way of computing the motivation to continue generating opportunism and avoidance behaviors is presented in Figure 5.4.

To test the ability of the robot to take advantage of opportunities, we use the same experimental setup as in previous sections but this time we put boxes on the side of the path of the robot as rewards, as shown in Fig. 5.5: when the robot moves forward, the boxes touching the contact sensor on the side of the robot increase the

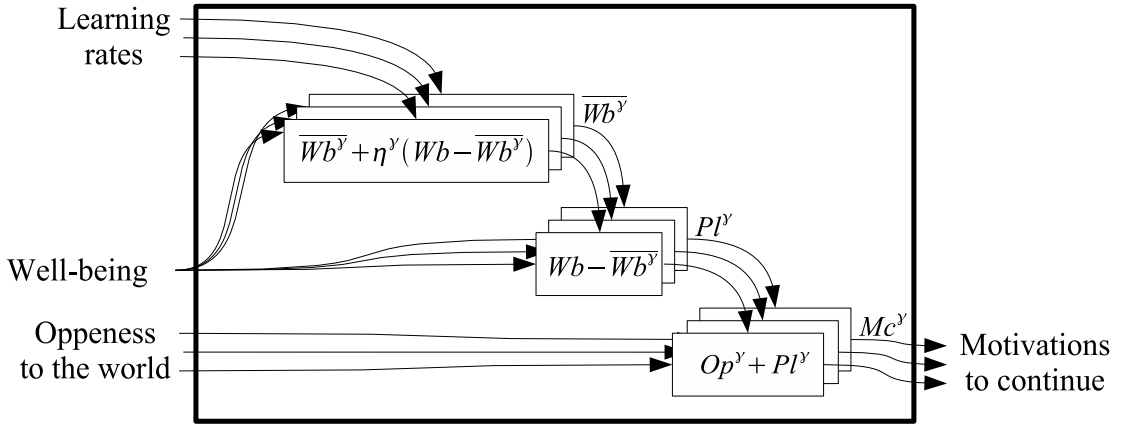


Figure 5.4: Computation of motivation to continue either to express opportunism or avoidance behaviors. The layers correspond to the time scales.

well-being. The values of the static parameters are similar to the previous experiment but s is set to 0.001 (a small s decreases the influence of the initial situation) and q to 0.75 (a large q decreases the influence of the exploratory behavior). We run a dozen experiments in three different contexts. In the first experiment (exp 1) where there is no external reward—the well-being is constant, like in Sec. 5.1—only s and q have changed. In the second experiment (exp 2), there is a reward at the front of the robot. In the third experiment (exp 3), the robot is already next to a reward and by exploring will lose it.

Figure 5.6 presents the successive positions and well-being of the robot during one typical trial in each context (exp 1 in solid line, exp 2 in dashed line and exp 3 in dotted line). We observe that the robot always starts with an exploratory behavior similar to that in Sec. 5.1; however, if it encounters a reward it accelerates (exp 2), and if the reward is lost it tries to come back and stop exploring for a while (exp 2) or forever (exp 3). The robot is able to produce the behaviors shown before (seeking stability and exploration) but can also interrupt these behaviors to take advantage

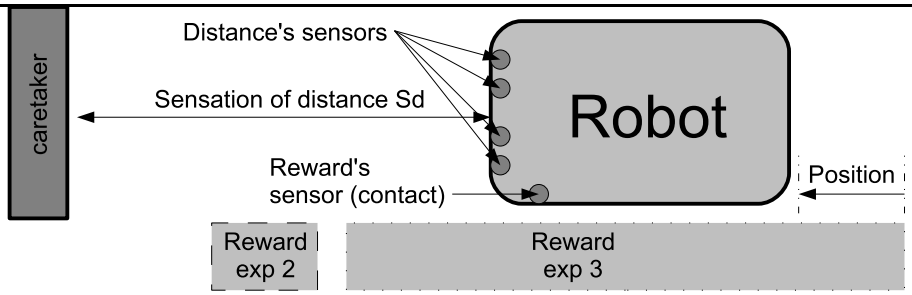


Figure 5.5: Setups of the experiments with rewards: in the first experiment there is no reward, in the second experiment (exp 2), the reward is on the way of the robot and in the third experiment (exp 3), the reward is on the side of the robot.

of opportunities and avoid dangers.

Figure 5.7 depicts the entire architecture allowing the robot to explore when it feels safe, seek stability if it feels uncomfortable, and interrupt these behaviors if it discovers an opportunity or a sudden danger.

5.3 Low-level Imitation

We have presented our final architecture which allows an agent to:

1. learn and remember familiar and pleasant sensations;
2. act in order to explore its environment;
3. interrupt the previous behaviors in order to take advantage of an opportunity, or avoid a danger.

All these behaviors are smoothly balanced depending on the context and the internal states of the agent. However, our goal was to produce imitative behavior (even if at low level) depending on affect and we have presented our final architecture without mentioning yet imitation.

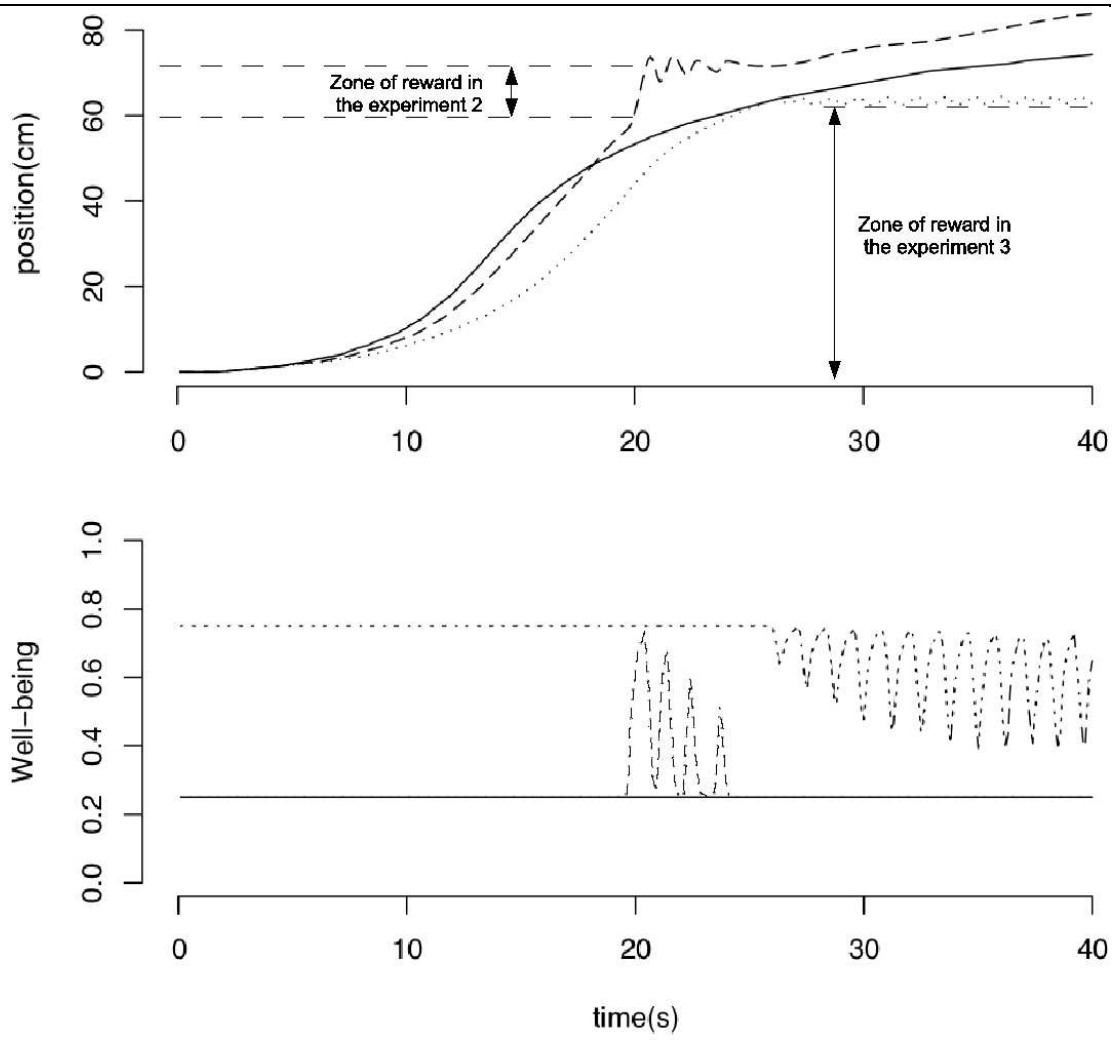


Figure 5.6: Successive positions of the robot (top) and its corresponding well-being (bottom) for the three experiments—exp 1 without reward in solid line, exp 2 with a local reward in dashed line, and exp 3 with a disappearing reward in dotted line.

Interestingly, our current architecture also allows a robot to perform low-level imitation as a side effect of the exploratory process depending on affect and pleasure. If the caretaker moves at the front of the robot, the robot receives unexpected sensations and inhibits its exploratory behavior. Moreover, if the robot is not too far from

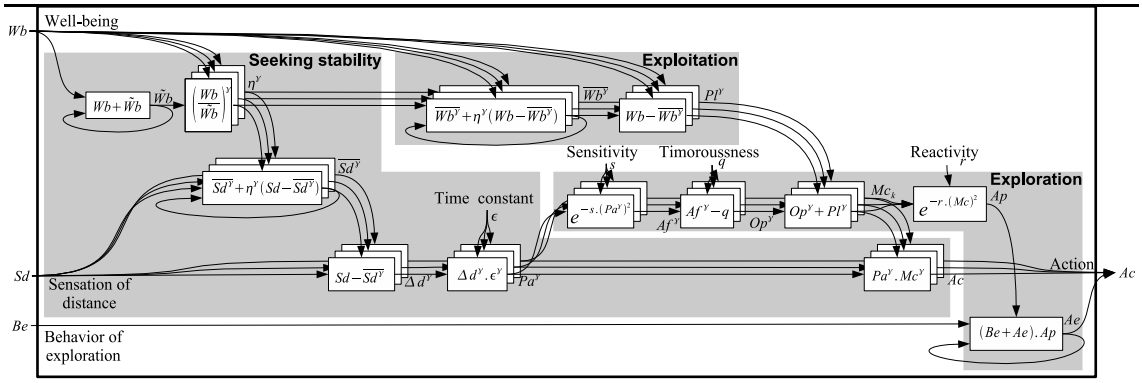


Figure 5.7: Final architecture allowing to produce seeking stability, opportunism, avoidance and exploration.

its desired sensation (familiar zone) or its well-being increases (positive pleasure), the motivation to continue (Mc^γ) is positive and therefore the robot tries to amplify the variation of sensation (the distance between the current new sensation and its desired sensations). The result is that the robot moves toward the caretaker when the caretaker moves toward the robot and the robot moves away from the caretaker when the caretaker moves away from the robot (see video: 'exploration and imitation with positive affect.mov'). This is not the case if the robot is in an unfamiliar zone or if the well-being decreases; we observe low-level imitation depending on affect and pleasure. This view of low-level imitation differs from other approaches of low-level imitation like (Andry et al. 2003, Demiris and Johnson 2003) because we do not consider imitation as the reduction of error between what is expected and what is actually sensed but, on the contrary, as the process of amplifying—only in the appropriate context—an unexpected or unfamiliar sensation.

Figure 5.8 presents a typical result of a dynamical interaction with the robot. In this case, we use the same setup as in the experiment about exploration (see Sec. 5.1), but this time the caretaker moves to observe the reaction of the robot.

In the top graph we can see the successive positions of the robot in solid line, and the estimated[1] position of the caretaker in dashed line. In the bottom graph we can see the values (Sd) of the distance sensor of the robot. We observe that the robot tries to amplify the relative movement of the caretaker (represented by the arrows in the figure) when its sensation is close to its initial sensation (imitative behavior), but this amplification becomes null or even negative when its sensation is far from its initial, therefore familiar, sensation (avoidance zone). Our robot can therefore imitate another agent to discover new sensations while remaining able to interrupt its behavior to avoid new dangers or take advantage of new opportunities like we have seen in the previous section.

5.4 Conclusion

We have presented a way to make our robot explore its world when it is in a familiar situation. Increasing autonomy and exploring the environment are essential activities for autonomous agents to learn new things and to consolidate past experiences and apply them to improve behavior. However, exploration is also risky as it exposes the agent to unknown, potentially overwhelming or dangerous situations, and therefore a trade-off must exist between activities such as seeking stability, exploring, imitating another agent (and, in so doing, discovering new sensations) and taking advantage of opportunities offered by new situations and events while at the same time avoiding danger. Our architecture achieves an adapted execution of different behaviors on the grounds of modulatory mechanisms based on notions of "well-being" and "affect." This includes production and modulation of imitative

[1]Estimation done using the absolute position of the robot and the detected distance of the caretaker to the robot

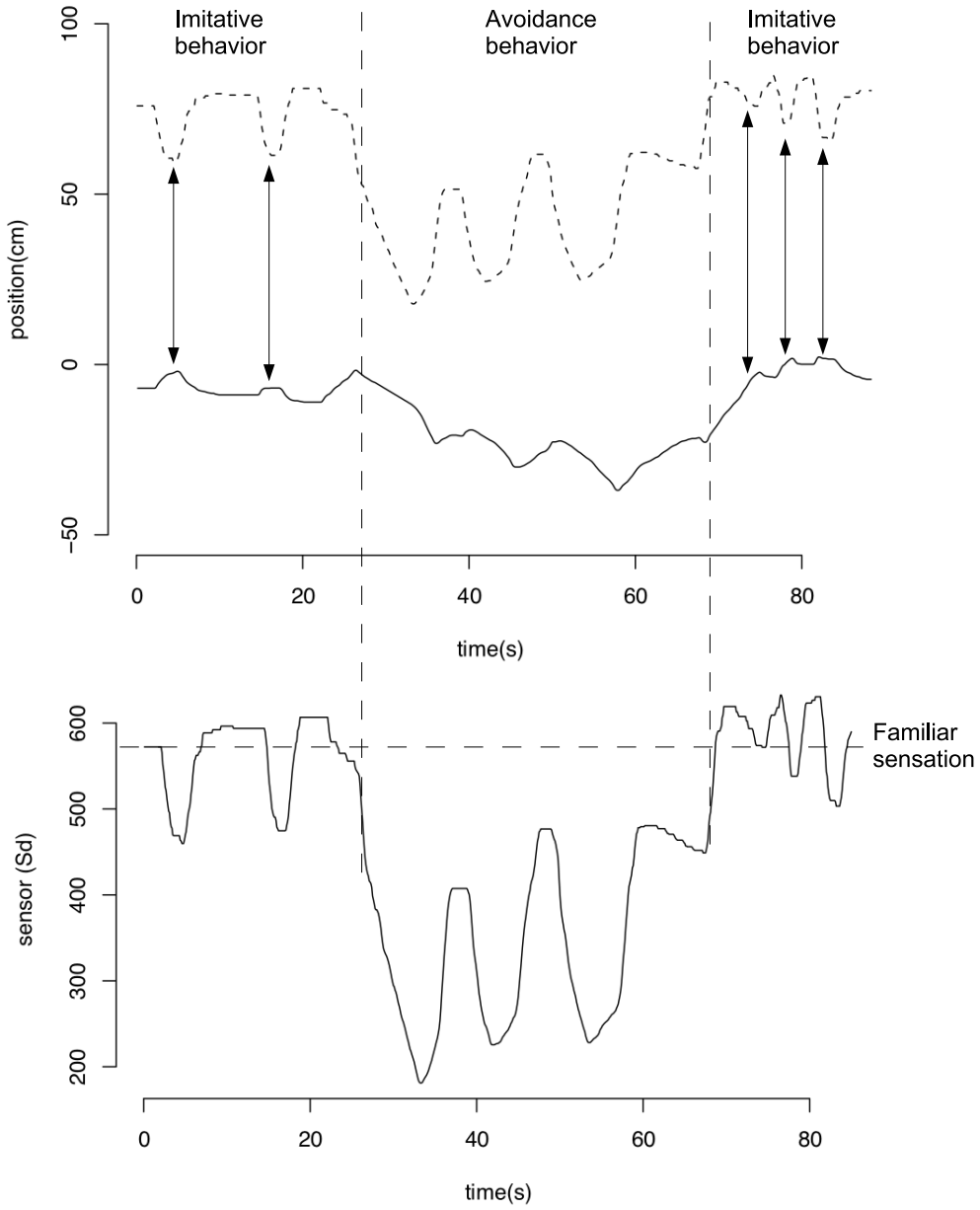


Figure 5.8: Top: successive positions of the robot (solid line) and of the caretaker (dashed line). Bottom: sensation from the sensors to the robot. When the sensation is close to the familiar sensation (the one the robot starts with), the robot moves like the caretaker (imitative behavior) but when the sensation is far from the familiar sensation, the robot avoids the movements of the caretaker (avoidance behavior).

behavior, which is used instead of the exploratory behavior when a "teacher" proposes new stimulations. The modulation of exploration depending on familiarity is not only a way to make the robot safely explore its environment, but it can be also a great advantage during human-robot interaction. Actually, if the robot explores (therefore take initiatives) when it is familiar with it environment, it means reciprocally that it will take initiatives only once the caretaker (or a human partner in the case of a companion robot) is familiar with the robot.

We have shown how affect can modulate low-level imitation in a coherent global architecture. However, there are some limitations like the fact that the robot cannot learn more than one sensation desired for each time scale. It can be also useful to memorize the sensations that should be avoided. This is the subject of our next chapter, where we will show how we can make the robot memorize an unlimited number of desired sensations at each time scale and also how it can memorize the sensations it should avoid. We will discuss about the other issues like the problem of multidimensionality later in Chap. 7.

Chapter 6

Complex Desired Sensations

"It's not just learning things that's important. It's learning what to do with what you learn [...]"—Norton Juster, in *The Phantom Tollbooth* (1961)

6.1 Introduction

We have presented an architecture allowing an agent to learn and remember sensations it should try to reach in order to increase or stabilize its well-being. We call these sensations "desired sensations". Desired sensations correspond to sensations associated with three factors:

1. the well-being;
2. the familiarity, probability for a sensation to occur;
3. the recency of a sensation in time.

Each desired sensation at different time scales is a desired sensation with different weights regarding these factors. The agent is therefore capable of setting its own goals (i.e. sensation it should try to reach) on the basis of reinforcement (value of the well-being), probability, and recency.

However, there are some limitations; for example, the agent can only remember one desired sensation for each time scale, whereas different sensations can be associated with the same level of familiarity, well-being, or recency. Moreover, for an agent it is useful to not only memorize the sensations it should reach, but also the sensations it should avoid. We present in this chapter a method to improve the learning system in order to make the agent learn: a) many desired sensations for each time scale, and b) "avoided sensations", the sensations it should avoid.

Although the main emphasis of our work is not reinforcement learning, it presents some commonalities with this approach to learning. Rewards or positive reinforcements can be seen as high well-being, whereas punishment and negative reward can be seen as low well-being. Section 6.2 presents the main ideas of classical approaches of reinforcement learning and discusses the problems raised by them, like the need of arbitrary discretization of the environment and the large need of computational resources. Then in Sec. 6.3 we propose a solution to handle these problems in the continuity of our current architecture. Section. 6.4 presents the results of experiments we have carried out on our real robot. Finally Sec. 6.5 draws some conclusions and perspectives to continue this work.

6.2 Classical Reinforcement Learning

The temporal-difference model (Sutton and Barto 1987) is a very common and efficient reinforcement learning method (Sutton and Barto 1998). Its principle is to discretize the inputs (from the sensors and the internal states of an agent) in order to obtain a finite number of possible states (inputs). The expected reward (i.e. well-being) for each state is evaluated using the actual reward of the state in addition to the reward expected in states immediately accessible. The agent then acts in order to reach the states maximizing the expected reward; it needs a world model to know which action to execute to get in one precise state. Even if the convergence of the algorithm is proved, learning is very slow because the agent needs to try each state several times, and the speed strongly depends on the discretization used, which can lead to a huge number of different states. It is therefore also very demanding in terms of memory, in order to store all the expected reinforcements for each possible state, or it needs a good generalizer (Univ. of Michigan RL Group 2007).

Q-learning (Watkins 1989) uses similar principles but it works even if the agent does not know which action to execute in order to reach a given state (it does not require a world model). The agent learns the expected reinforcement for each possible state-action couple. This decreases again the speed of learning because there are many more possibilities to explore, and the quantity of memory needed is multiplied by the number of different possible actions.

6.2.1 The Issue of Discretization

In artificial intelligence, numerous powerful algorithms have been designed to learn, anticipate and decide. However, they are often inappropriate when applied to robots

in the real world, particularly if the robots are not pre-programmed to detect specific stimuli. For example, many models of classical or instrumental conditioning need to predefine the set of possible stimuli to consider. Information theory (Cover and Thomas 1991) provides powerful tools to statistically measure the temporal correlation between events and anticipate them (Capdepuy, Polani and Nehaniv 2006). However, this approach also relies on discretization, and the problem is again to define the set of events by discretizing the world.

Discretization can be adaptive, for example by grouping together events that carry the same predictive information. To do this, we can use classification algorithms like k-mean, Kohonen's maps, Estimation-Maximization algorithm—see e.g. (Butz, Sigaud and Gérard 2003). Many of these algorithms need strong assumptions on the distribution of classes and the discretization needs to be arbitrarily or randomly initiated therefore the quality of the learning process depends on random initializations. When developing the Q-learning algorithm, Watkins was aware of the difficulty of coping with continuity: "To avoid the complications of systems which have continuous state-spaces, continuous action sets, or which operate in continuous time, I will consider only finite, discrete-time Markov decision processes" (Watkins 1989, page 38). Even after the discretization is done, the algorithm converges quite slowly because it needs to try several times the different possible state-action pairs in order to estimate statistically the reward that can be expected for each one. Once the reinforcement can be reliably anticipated for each state-action pair, the agent can act in order to reach the state with the highest expected reinforcement.

These approaches are very powerful when they are used in simulation, since the environment is often discrete (e.g. a grid where the agent is moving) and it is easy to make an agent try different situations a large number of times. They can be

well adapted to robotics when the elements of the environment are predefined, and there are obvious salient cues that the robot can consider as classes of events (e.g. a salient color or pattern).

However, in the case of robots in real environments without specific features, the robots have to find by themselves the cues predicting rewards. These cues are not necessarily salient, and can for example be a specific light intensity, a range of sound frequencies or a specific position, rather than binary signals associated with the presence or absence of light, sound, shape, etc., as it is usually the case in discretized environments. Humans and animals are very efficient at discriminating between similar stimuli if they have distinctive predictive values. In this case, using the salience of sensations can be misleading, since for example a light being turned on or off might not have any predictive value, whereas a small change in the intensity of a light at a specific level can be significant.

Most algorithms involving discretization are not able to cope efficiently with this kind of situation because they waste vast amounts of memory storing the predictions of expected rewards for many different values of the sensory input, even though most of them are not relevant or are redundant. Moreover, there is usually no difference between the effect of a small reward obtained immediately and the promise of an important reward to be obtained later. However, in some cases it is very important to make such difference: for example if a robot is about to "die" it should go where it is sure and quickly find at least a small reward (e.g. satisfy an urgent need in part by consuming a small resource), whereas it should try to maximize the long-term reward when it has more time (e.g. go to a farther location or explore for larger quantities of resources, where it could possibly better satisfy needs).

6.3 Our Approach to Learning

As there is no free lunch (Wolpert and Macready 1997), there is no general algorithm that, on average, performs better than another one without further and more suitable assumptions, we will make and use assumptions about the world in order to improve our architecture. In our work, we assume that the world is continuous since physical robots acting in the real world have to deal with a continuous, rather than discrete, environment: there are continuous variations of rewards with continuous variations of sensory inputs (sensations), and the relations between rewards and sensory inputs are consistent. Consequently, if the agent—a robot in our case—receives a high reward for a specific sensory input, it can anticipate a good reward for other close or similar sensations. Therefore, instead of estimating the expected reward for all the many possible states and trying to reach the state anticipating the maximum reward, we propose to make the robot remember only the sensation (desired sensation) associated with the best reward (see Fig. 6.1). As previously mentioned, our robot can memorize and recall desired sensations at different time scales, depending on its affective state. However, for the sake of clarity, in this chapter we focus only on one time scale.

6.3.1 Desired Sensations

To illustrate the various possibilities, we consider a continuous environment—our usual environment in which a robot interacts with a human and objects in the real world—and as previously, we use S, the distance of obstacles at the front of the robot for sensory input, A for the speed of forward movement (backward movement if the speed is negative) and Wb to represent current well-being or the immediate

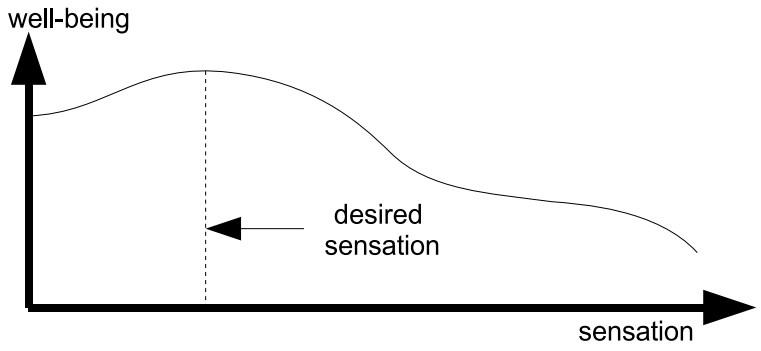


Figure 6.1: Desired sensation depending on the maximum of well-being associated with the sensation

reward in terms of reinforcement learning. We consider the problem depicted in Figure 6.2: the robot receives the distance to a landmark (or a caretaker) as sensory input (sensation$S$) and its well-being (Wb) increases when a reward (e.g. a source of energy symbolized by a box) is located at its side.

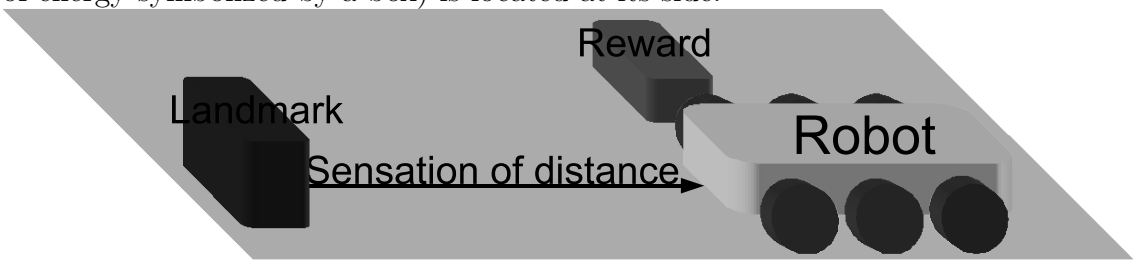


Figure 6.2: Using the distance to a landmark detected by its distance sensors, the robot must learn a desired sensation: which corresponds to a reward on its side.

In order to make the robot learn the sensation associated with the highest well-being, we could simply set the desired sensation ($\hat{S}$) to be equal to the current sensation (S), but only when the well-being (Wb) is higher than the highest remem-

bered well-being ($\hat{Wb}$):

$$\text{if } Wb > \hat{Wb} \text{ then } \begin{cases} \hat{Wb} = Wb \\ \hat{S} = S \end{cases} \tag{6.1}$$

The problem with this equation is that if the well-being is very high once and is never high again, or if the sensation is very hard to obtain, the desired sensation learned would be useless. Moreover, the robot would not be able to learn more than one sensation associated with a reward. Actually, even if it memorizes another desired sensation associated with a slightly smaller reward, the principle of continuity makes this desired sensation infinitely close to the previous one learned as we can see in Figure 6.3. Therefore, to be reliable and robust the robot should not only memorize the sensations associated with the highest reward, but also the sensations associated with a positive reward at a high probability. We have shown in Eq. (6.1)

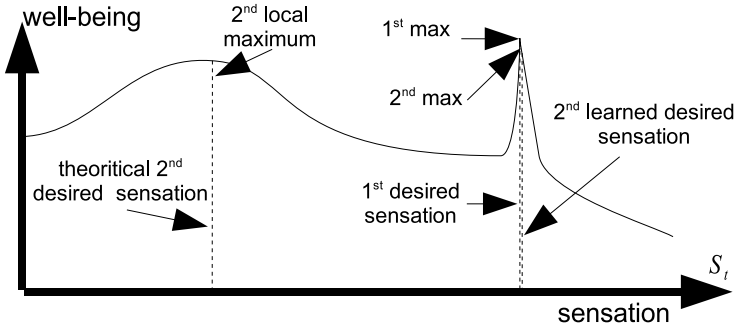


Figure 6.3: Impossibility to learn two local maxima.

how to memorize the sensation only associated with the maximum well-being.

On the contrary equation (6.2) shows how the robot can memorize the most frequent sensation ($\overline{S}$) (maximum of familiarity) as the average of all sensations at each point in time (t). We have already presented this equation similar to the

learning rule of Rescorla and Wagner (Rescorla and Wagner 1972) in Chap. 4, where we use it to make the robot become imprinted to familiar sensations.

$$\overline{S}_t = \overline{S}_{t-1} + \eta_t \left(S_t - \overline{S}_{t-1} \right) \tag{6.2}$$

The learning rate is $\eta_t = \frac{1}{\widetilde{T}_t}$, and in this case we only need a variable that increases with time ($\widetilde{T}_t = \widetilde{T}_{t-1} + 1; \widetilde{T}_0 = 1$) and another variable to memorize the current average sensation ($\overline{S}$). The complexity of the calculation is very low and biologically plausible. As we work with a discrete digital computer, the calculation are done step by step, however, the values of functions we use (e.g. average) do not depend on the size of these steps, only the precision can be affected.

Now the agent can learn two extreme cases: the sensation associated with the best reward ($\hat{S}$), and the average sensation ($\overline{S}$), regardless of what the reward is. It is nevertheless not very useful to learn only those extreme cases. The first one indicates the sensation associated with the best reward, but this memory might not be reliable as it may have happened only once. The second case indicates which are the sensations that happen more often, but this does not mean that they are good things for the robot, only that they appear often in its environment. However, all the intermediate cases are very important because in order to maximize the cumulative reward, the agent should balance the effect of the reward and the effect of the probability. If a robot urgently needs a reward (for example consuming a resource to avoid dying), it should focus on the sensations promising small rewards with high probabilities (easy to obtain), but if the situation is not urgent, it should focus on sensations promising higher rewards in order to maximize the cumulative reward and also to learn more about these high rewards. The robot must thus

be able to memorize a range of desired sensations, from those often obtained but predicting small rewards, to those rarely obtained but predicting high reward.

In Sec. 4.2 of the Chap. 4, we have shown how a robot can learn the average "best" sensation by weighting the learning rate with the well-being: η_t with $\eta_t = \frac{Wb_t}{\widetilde{Wb_t}}$ with $\widetilde{Wb_t} = \widetilde{Wb_{t-1}} + Wb_t; \widetilde{Wb_0} = Wb_0$. Using the time scales (γ), we were able to balance the effect of the recency, familiarity and well-being in the learning of desired sensations. However as we used only one parameter to balance three factors, they could not be independent: either recency and well-being (small γ) have a strong influence on the learning rate or past and familiarity have a strong influence (large γ). We were unable to make the robot memorize the absolute best sensation it had like in Eq. 6.1 because if γ is close to zero, it would only memorize the best sensation among the most recent sensations—for a null γ the memorized desired sensation is equal to the current sensation.

We propose in Eq. 6.3 a modification to make the agent learn different desired sensations ($\overline{S^k}$) where the balance between the weight of the well-being and the familiarity is controlled by the parameter k (we consider only one time scale $\gamma = 1$):

$$\begin{aligned} \overline{S_t^k} &= \frac{e^{k.Wb_0}.S_0 + \ldots + e^{k.Wb_t}.S_t}{e^{k.Wb_0} + \ldots + e^{k.Wb_t}} \\ &= \overline{S_{t-1}^k} + \frac{e^{k.Wb_t}}{e^{k.Wb_0} + \ldots + e^{k.Wb_t}} \left(S_t - \overline{S_{t-1}^k}\right) \end{aligned} \tag{6.3}$$

For extreme values of k (namely 0 and $+\infty$) we obtain, respectively, the same results as in Eq. 4.2 where only the familiarity is taken into account because $e^0 = 1$, and Eq. 6.1 where only the maximum of the well-being is taken into account because:

$$\lim_{k \to +\infty} \frac{e^{k.Wb_0}.S_0 + \ldots + e^{Wb.r_t}.S_t}{e^{k.Wb_0} + \ldots + e^{k.Wb_t}} = S_{argmax(Wb_0,\ldots,Wb_t)}. \tag{6.4}$$

We can have all the combinations of the weight of the well-being versus the familiarity by making the values of k vary between 0 and $+\infty$.

Like in our previous method, only the variation of the well-being and not its absolute value, influences learning; therefore, we do not need to define a priori which value of well-being has to be considered as a good well-being.We can actually add any constant (c) to the well-being and it does not change the learning rate:

$$\begin{aligned} \eta_t^k &= \frac{e^{k.(Wb_t+c)}}{e^{k.Wb_0+k.c}+\ldots+e^{k.Wb_t+k.c}} \\ &= \frac{e^{k.Wb_t}.e^{k.c}}{e^{k.Wb_0}.e^{k.c}+\ldots+e^{k.Wb_t}.e^{k.c}} \\ &= \frac{e^{k.Wb_t}}{e^{k.Wb_0}+\ldots+e^{k.Wb_t}} \\ &= \frac{e^{k.Wb_t}}{\widetilde{Wb_t^k}} \end{aligned} \tag{6.5}$$

with $\widetilde{Wb_t^k} = \widetilde{Wb_{t-1}^k} + e^{k.Wb_t}$.

6.3.2 Avoided Sensations

We have shown how a robot can learn desired sensations associated with high well-being, but it can also be useful to learn sensations predicting danger or low well-being (negative reward or punishment) in order to avoid them. With our model, they are easy to compute as they are equal to the sensations $\overline{S_t^k}$ for negative values of the parameter k. If k tends to $-\infty$, $\overline{S_t^k}$ corresponds to the "worst" sensation. We call them "avoided sensations".

The issue of computing the desired sensations[1] is that they can be between two

[1]We use "desired sensation" here as a generic notion, but it can refer either to a proper desired sensation ($k > 0$) or to the particular case where $k < 0$, which should rigorously be an avoided

local maxima and therefore predict (reward) high well-being where the well-being is in fact low (no reward), i.e. a "false positive", see Figure 6.4. This is a very general

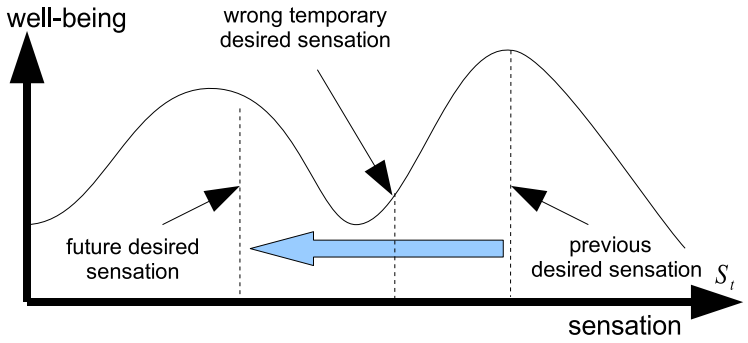


Figure 6.4: Wrong desired sensation, resulting from the average of multiple local maxima.

problem in optimization; while searching for the global maximum of a function we may temporarily find a local minimum. In fact, this happens when the desired sensation is moving from a local maximum to another local maximum, because it is higher, or more frequent. A solution is to make the robot quickly forget the past and consequently have its desired sensations quickly moving from local maxima to local maxima. We have therefore to focus learning on recent sensations, and focusing learning on past or recent sensations is exactly the role of the time scales that we have developed in Chap. 4.

Therefore we raised the learning rate η_{t^k} to the power of γ, with γ taking values between 0 and 1, in order to make the recent sensations have more weight than the old ones in the learning of the desired sensations:

$$\overline{S_t^{\gamma,k}} = \overline{S_{t-1}^{\gamma,k}} + \left(\frac{e^{k.Wb_t}}{\widetilde{Wb_t^k}}\right)^{\gamma} \left(S_t - \overline{S_{t-1}^{\gamma,k}}\right). \tag{6.6}$$

sensation.

The smaller γ, the larger the learning rate is, and the faster the desired sensations change; therefore, the desired sensations oscillate between local maxima depending on the current sensations (exploration) of the agent, as depicted in Figure 6.5. The

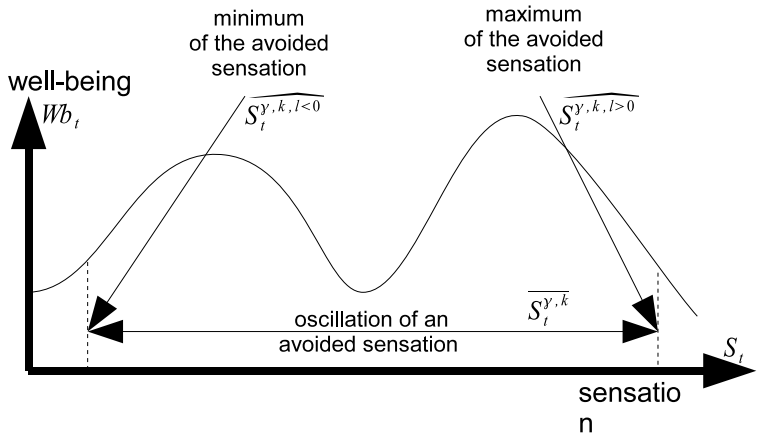


Figure 6.5: Oscillation of a desired sensation between local maxima. While an agent explores the sensory-space, its desired sensations move from local maxima to local maxima crossing local minima.

main problem with partly forgetting the past is that the robot will not be able to remember a sensation associated with a high well-being if it did not experience it recently, it is especially a problem for the imprinted sensation. However, desired sensations oscillate between local maxima, and avoided sensations oscillate between local minima; therefore, if the robot memorizes the extreme values ($\widehat{S}$) of the successive desired and avoided sensations (see Figure 6.6 desired sensation), it can remember two sensations—minimum and maximum values—anticipating a high well-being and two sensations anticipating a low well-being. We will see in the next chapter how we could make the agent memorize an unlimited number of sensations of each kind.

Using the Eq. (6.3) the agent was able to memorize the sensations ($\overline{S_t^k}$) associated

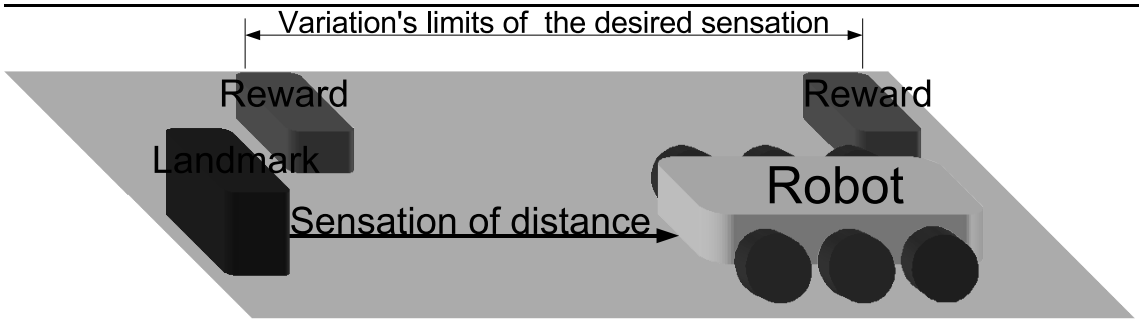


Figure 6.6: The desired sensations strictly oscillate between the two extreme rewards. Learning the variation's limits of the oscillations makes the agent learn the sensations associated with the extreme rewards.

with extreme values of the well-being (low well-being with $k < 0$ and high well-being with $k > 0$); with the same principle, we will make the agent learn the extreme values of the desired sensations. Since there is a parameter k to control which extremum (high or low) of the well-being is considered, we use a new parameter l to control which extremum values of the desired sensations to consider (minimum values with $l < 0$ and maximum values with $l > 0$, respectively left and right extremities in Fig. 6.5). Equation (6.7) presents how the extreme desired and avoided sensations ($\widehat{S_t^{\gamma,k,l}}$) are computed:

$$\widehat{S_t^{\gamma,k,l}} = \widehat{S_{t-1}^{\gamma,k,l}} + \frac{e^{l.\overline{S_t^{\gamma,k}}}}{\widetilde{S_t^{\gamma,k,l}}} \left(\overline{S_t^{\gamma,k}} - \widehat{S_{t-1}^{\gamma,k,l}} \right) \tag{6.7}$$

with $\widetilde{S_t^{\gamma,k,l}} = \widetilde{S_{t-1}^{\gamma,k,l}} + e^{l.\overline{S_t^{\gamma,k}}}$; $\widetilde{S_0^{\gamma,k,l}} = e^{l.\overline{S_0^{\gamma,k}}}$.

Figure 6.7 illustrates learning of extreme desired sensations ($k > 0$) and Fig. 6.8 illustrates learning of extreme avoided sensations ($k < 0$).

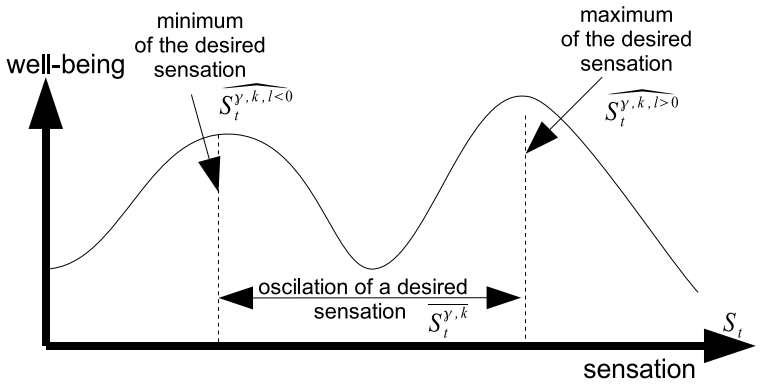


Figure 6.7: Extreme values of a desired sensation ($k > 0$). Values for l are $l < 0$ at the leftmost extreme, and $l > 0$ at the rightmost extreme.

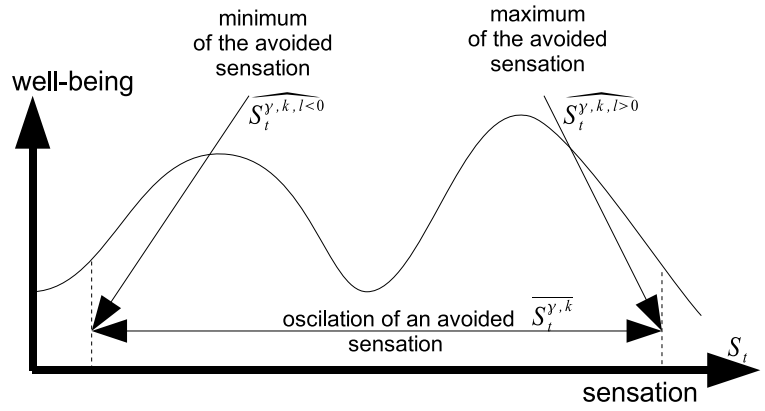


Figure 6.8: Extreme values of an avoided sensation ($k < 0$). Values for l are $l < 0$ at the leftmost extreme, and $l > 0$ at the rightmost extreme.

6.4 Experiments and Results

We have tested this algorithm on our Koala robot that had to remember sensations associated with reward or punishment (high or low well-being). We make[2] the robot alternatively move forward and backward, and we observe how it creates its desired

[2]It is hardcoded.

and avoided sensations depending on variations of its well-being (depending on the rewards it reaches). The experimental setup was the one depicted in Fig. 6.6; the sensory input (S) used was its frontal distance sensor measuring its distance to a caretaker[3] at the front used as a landmark. The right distance sensor was used to detect rewards—an object located in close proximity within the range of its right sensor provides high value of the well-being of the robot. Figure 6.9 shows the well-being of the robot as a function of the sensation of distance to the landmark (e.g. the caretaker).

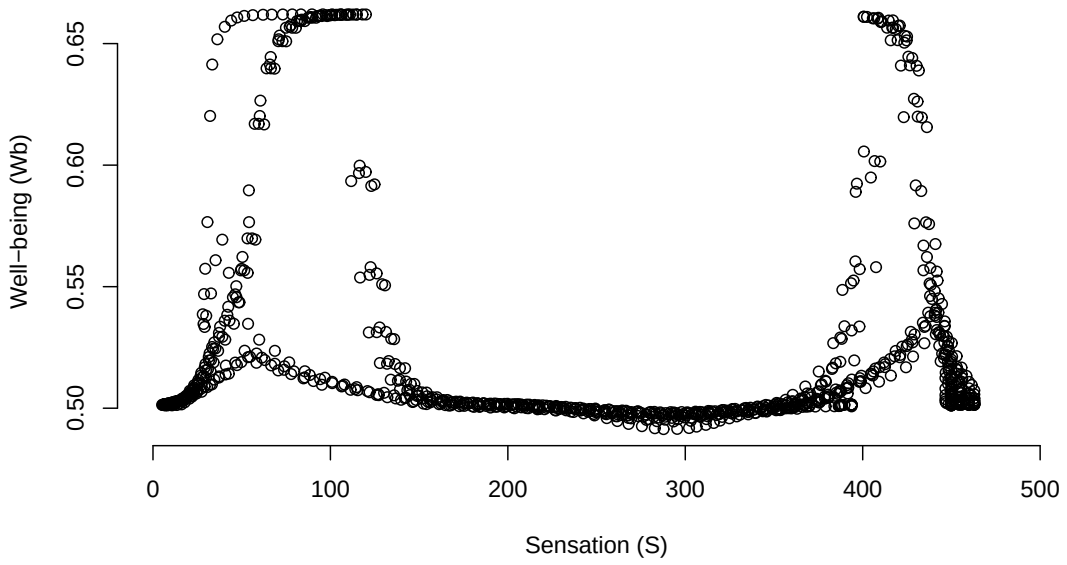


Figure 6.9: Value of the well-being (Wb) as a function of the sensation (S) for different pass of the robot. We see that the maximum of well-being is for sensations of distances of about 75 and 425 (not in spatial units, but values given by the sensors), which correspond to the presence of the two objects (rewards) on the right of the robot.

[3]We used different types of stimuli as "caretaker", notably humans and cardboard boxes.

In Fig. 6.10 we can observe how the desired and avoided sensations of the robot evolve with time and experiences of the robot. We compute the desired sensations $\overline{S^{\gamma,k}}$ with $k = +400; \gamma = 0.9$, and the avoided sensation $\overline{S^{\gamma,k}}$ with $k = -400; \gamma = 0.9$. If k or γ differ, the curves are more or less smooth but qualitatively similar. The desired sensation (in solid line) oscillates between sensations 75 and 425, which correspond to the presence of the reward (object on the right of the robot). The avoided sensation (in dash line) strictly oscillates between the two rewards at the beginning and then around them, which indicates that the robot should avoid to be either between the objects providing reward or behind them.

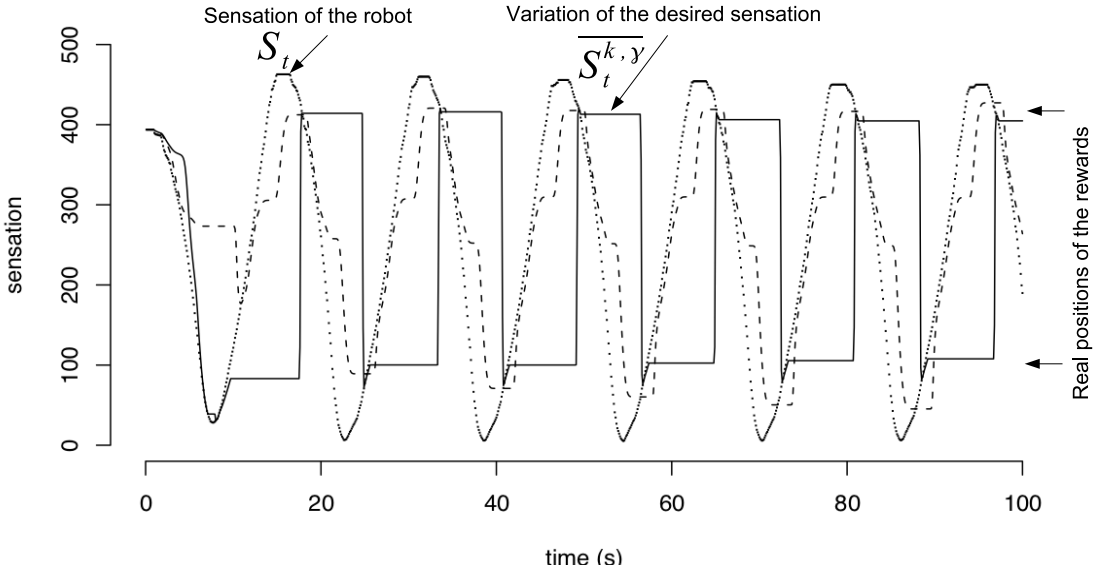


Figure 6.10: Evolution of the current sensation (S_t) of the robot in dotted line, the desired sensation ($\overline{S^{\gamma,k}}$ with $k = +400; \gamma = 0.9$) in solid line and the avoided sensation in dashed line ($\overline{S^{\gamma,k}}$ with $k = -400; \gamma = 0.9$). The desired sensation oscillates between sensations 75 and 425, which correspond to the presence of the reward. The avoided sensation oscillates strictly between the two rewards at the beginning and then around (farther) them, indicating that the robot should avoid to be either between or around the objects providing the rewards.

Desired and avoided sensations are constantly changing; therefore, the robot cannot remember anything for a long time. However, the next step for the robot is to memorize the extremes of these desired and avoided sensations. We present in Figure 6.11 the evolution of these extremes ($\widehat{S^{\gamma,k,l}}$) for the same values of k and γ and -0.1 and 0.1 for l (l is small because the amplitude of the sensation is large); however, this does not have effect on the qualitative results. The extremes of the avoided sensations quickly converge (almost at the first cycle) to the sensations corresponding to the rewards (sensations of 75 and 425). The extremes of the avoided sensations correspond at the beginning to the sensation obtained when the robot is located between the rewards, and at the end to the sensation obtained when it is located behind the rewards. This means that the robot should avoid staying between or behind the objects providing reward, since in those two cases no reward is obtained.

6.5 Conclusion and Perspectives

We have presented the first basic principles and implementation of what can be regarded as a new approach to reinforcement learning, where agents can learn to anticipate rewards and punishment using their sensory inputs especially in the real (continuous) world.

Doya, for example, proposed to approximate the reward function in order to process reinforcement learning in continuous time and space (Doya 2000), but we argue that it is enough to only memorize where the rewards are even if the robots cannot know what are the values of these rewards. The advantages of our approach are that the agent memorizes only the relevant information and does not need much

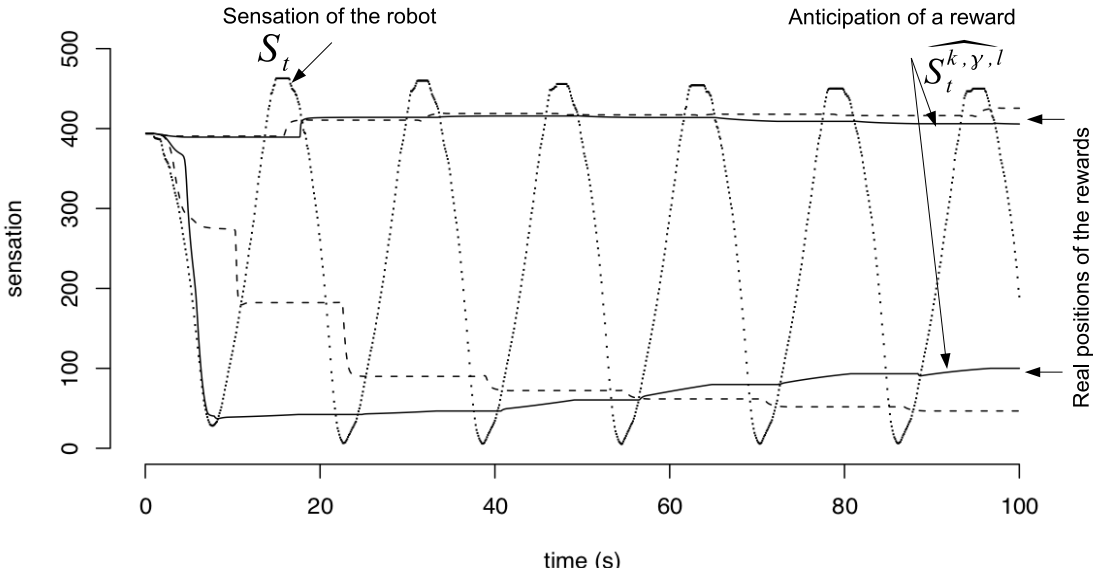


Figure 6.11: Evolution of the extreme values ($S^{\gamma,k,l}$) of the desired and avoided sensations using the same values of k and γ than in Fig. 6.10, but with a value of 0.1 for l in the curves at the top, and -0.1 in the curves at the bottom. The extremes of the desired sensations are in solid line and the extremes of the avoided sensations are in dashed line.

memory or computer time; it does not use notion of events or discretization, and this strongly reduces the effects of a priori choices and decreases learning time. The agent cannot memorize precise situations: the exact highest reward (k would need to be $+\infty$), or the exact first situation (γ would need to be $+\infty$), etc. However, when we work in the real world with robots we are not interested in exact particular situations, but in situations which last for a while or happen often. The difficulties are to define how long is a "while" and how often is "often" and actually, there are no definite right answers. We have addressed already this issue in Chap. 4 for the time scale, and we have made the agent memorize many desired sensations with many different time scales. We propose to apply the same process here and make the agent learn many desired sensations, some where the value of the reward

is important, some where the familiarity is important, some where the recency is important, etc. Therefore, to make the agent learn all these combinations of desired sensations we use different variables $S^{\gamma,k,l}$ with different combinations of the values of γ, k, and l.

The agent still needs to decide which of these desired sensations it should consider: should it reach a desired sensation associated with a high well-being ($k > 0$) or avoid a desired sensation associated with a low well-being ($k < 0$)? Should it give more importance to the value of a reward or to its probability ? We have already proposed a solution in Chap. 4 to select the time scale of a desired sensation using the internal state of the agent. In the next chapter, we will discuss how to make more complex selections among desired sensations driven by three parameters (γ, k, l). We will also discuss the treatment of multidimensional dependent sensations as we have already dealt with the case of the independent ones in Chap 4.

Chapter 7

Discussion

In this chapter, we first present our ideas and proposals to go further on different specific points of the thesis and in a second part we study the links of our propositions with other studies, and finally we present our new *Perceive, Appraise and Act (PAA)* architecture which structures and formalizes what we have done and proposed.

7.1 Going Further

7.1.1 Multiple Desired Sensations

We have shown in Chap. 6 how an agent can memorize the sensations associated to two extreme positive and negative rewards; however, we can extend our approach to make the robot learn many more rewards: looking for the two extreme desired sensations in between two extreme avoided sensations and so on. We can define several levels of desired sensations: the first one representing the limits of the avoided sensations on the total range of sensations the agent has experienced; then inside

these limits, the limits of the desired sensations, and inside these new limits, the limits of the avoided sensations at the second level etc. Figure 7.1 illustrates two levels of desired sensations in an example with three rewards.

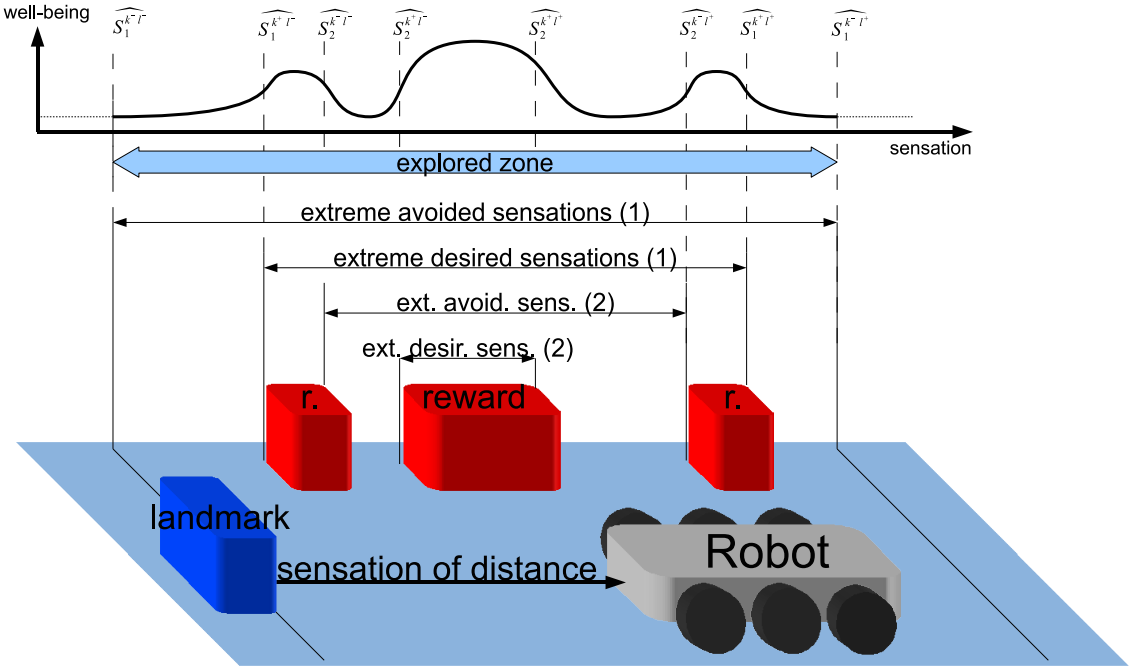


Figure 7.1: Using hierarchical intervals between extreme desired and avoided sensations, the robot can memorize many rewards or punishments. On the extreme sensations ($\widehat{S}$), the parameters γ and t have been omitted, the signs after the parameters k, and l precise if the values of the parameters are positive or negative. The subscript number indicates the level of the extreme sensation (here we represent only two levels).

7.1.2 Exploration, Opportunism and Danger Avoidance

Inside the limits of the explored zone, the agent has learned the sensations it should avoid or reach. However without external stimulation, the agent is not able to explore and the explored zone depicted in Fig. 7.1 is restrained to its initial sensation.

All the desired sensations ($\overline{S}$) and extreme desired sensations ($\widehat{S}$) are equal to the initial sensation.

We raised this issue in Chap. 5 where we showed that an agent does not only has to learn which sensations are associated with high well-being but has also to explore its environment in order to discover these sensations. We first proposed to use the notion of affect measuring the familiarity and "positiveness" of the current sensation to trigger an exploratory behavior. As soon as the agent moves too far from the familiar and positive sensations the affect becomes low again and stops the exploration until the current new sensation becomes familiar; this process prevents risky explorations.

Moreover, we have shown the exploration process should be modulated by an exploitation process inhibiting or amplifying the effect of an ongoing action if the well-being is respectively decreasing or increasing. The use of the notion of extreme desired sensations is therefore well adapted to directly modulating the exploration process depending on the opportunities or sudden danger. When the well-being is increasing, the extreme positive desired sensations (with a positive parameter k) will change faster than the extreme avoided sensations (with a negative parameter k). The difference between two kinds of extreme desired sensations can therefore directly modulate the current ongoing actions, which indirectly corresponds to using the variation of well-being as we did in Chap. 5 but in a more coherent way with the extreme desired sensations.

When the agent is in a safe and familiar situation (high affect) it should be eager to explore, its openness (Op) to the world will be high and will give motivation to continue (Mc) the exploration of sensations out of the bounds of its extreme desired sensations. However, as we saw in the Chap. 5, exploration should be modulated by

the variation of the well-being (pleasure Pl) which can be evaluated by measuring the difference between the extreme positive desired sensations and the extreme negative (avoided) desired sensations. If the well-being of the explored sensation increases, the positive extreme desired sensations becomes closer to the new sensation, the difference (Pl) between extreme positive and negative desired sensations is positive, and the motivation to continue (Mc) increases (see Fig. 7.2). On the contrary, if the well-being of the explored sensation decreases, the negative extreme desired sensation becomes closer to the new sensation, the difference (Pl) between extreme positive and negative desired sensations is negative, and the motivation to continue (Mc) decreases (see Fig. 7.3).

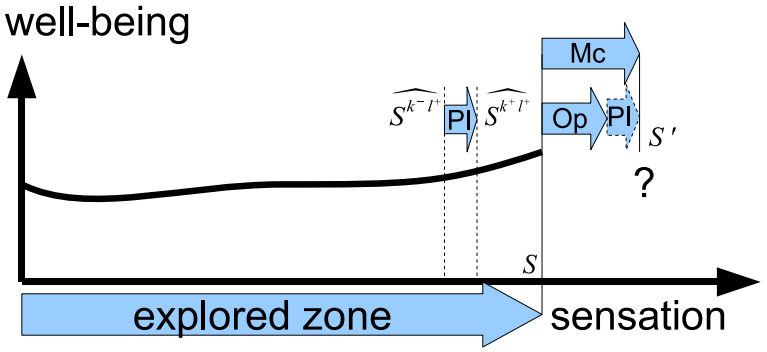


Figure 7.2: When the explored sensation S is associated to high well-being, the pleasure Pl is positive which increases the motivation (Mc) to continue the exploration (Op).

7.1.3 Multiple Dimensions

In this thesis, we have purposely limited our study to one dimension in order to simplify and clarify what we are studying. However, for most real problems, it is necessary to deal with multiple dimensions or modalities. Although this is not an issue for independent sensations and actions, this becomes complex when the

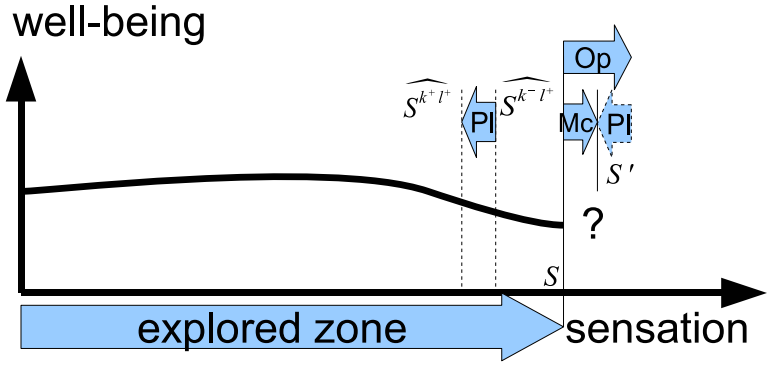


Figure 7.3: When the explored sensation S is associated to low well-being, the pleasure Pl is negative which decreases the motivation (Mc) to continue the exploration (Op). It may even stop the exploration.

different modalities of sensations and actions are correlated. In Chap. 4 we have shown an example with two independent dimensions. The desired sensation $\overline{S}$ was a vector of two components, the measure of the distance on the left side of the robot s_0 (the average of the left sensors) and the measure of distance on the right side of the robot s_1 (the average of the right sensors). The action A was also a vector of two components, the velocity of the left wheel a_0, and the velocity of the right wheel a_1. The robot was able to follow a care-taker in a two dimensional space because the two sides of the robot can be independent and work in parallel. For example, if the robot is too far from the person on the left side (s_0) it activates its left wheel (positive a_0) and better follows the care-taker. Independently the same process happens on the right side and the resulting behavior is coherent as the robot turns to follow or avoid the care-taker in a same way that a Braitenberg vehicle (Braitenberg 1984) follows or avoids a light—with the great advantage in our case that the robot decides itself to follow or not. Figure 7.4 illustrates the fact that the two components ($\overline{s_0}$ and $\overline{s_1}$) of the desired sensation ($\overline{S}$) have independent values.

However, in most cases, the dimensions are not independent, the ideal value of

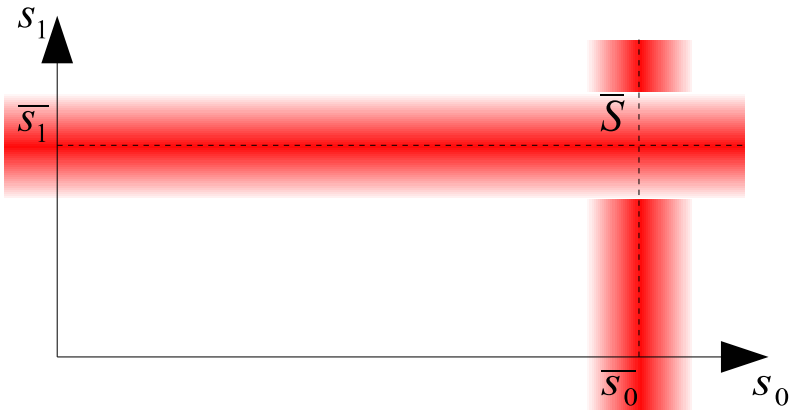


Figure 7.4: When the dimensions are independent, each component of a desired sensation has a value independent of the other component of the desired sensation.

a component of a desired sensation on one dimension depends on the components of other dimensions. Let us imagine an agent evolving in a 2D world with zones of reward and punishment. The components of the sensation of the agent are the coordinates of the agent in the world. The well-being increases when the agent is in zones of reward and decreases when it is in zones of punishment, see Fig. 7.5.

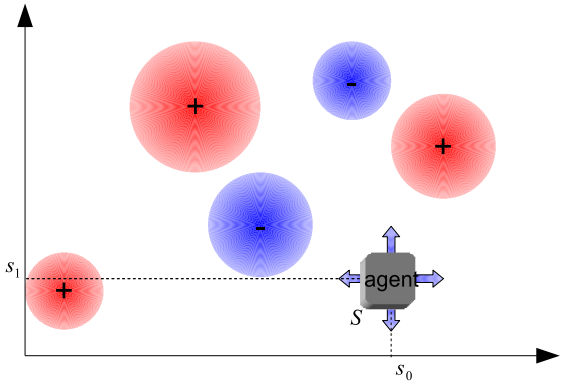


Figure 7.5: The agent has to learn where the zones of reward (in red with a plus sign) and punishment (in blue with a minus sign) are. The components of the sensation are the coordinates of the agent ($S = \{s_0, s_1\}$).

In this case the component of a desired sensation for example the abscissa de-

pends on the ordinate; for a same abscissa the agent can have a reward or a punishment depending on the ordinate. As we did in Sec. 7.1.1 we propose to bound the zones of reward and punishment using extreme desired sensations for each possible combination of extreme values of desired sensations on each dimension. In one dimension, we bounded each desired sensation with two extreme desired sensations defining the minimum and maximum values of the desired sensation using respectively negative and positive values of l. In multidimensional spaces, we bound each desired sensation with extreme desired sensations defined by all the combinations of minimum and maximum values for each component of the desired sensation.

In our 2D example, we use two parameters (l_0 and l_1), one for each dimension and for each desired sensation ($\overline{S_t^k}$) we define four extreme desired sensations with all the combination of negative and positive values of l_0 and l_1 ($S_t^{\widehat{kl_0^-l_1^-}}$, $S_t^{\widehat{kl_0^-l_1^+}}$, $S_t^{\widehat{kl_0^+l_1^-}}$, $S_t^{\widehat{kl_0^+l_1^+}}$). The learning rate of an extreme desired sensation is modulated by the contribution of the desired sensation on each dimension. Therefore, the final learning rate is the product of the learning rate for each dimension ($\eta_t^{k,l_0}, \eta_t^{k,l_1}$), see Eq. 7.1.

$$S_t^{\widehat{k,l_0,l_1}} = S_{t-1}^{\widehat{k,l_0,l_1}} + \eta_t^{k,l_0}.\eta_t^{k,l_1}(\overline{S_t^k} - S_{t-1}^{\widehat{k,l_0,l_1}}) \tag{7.1}$$

with

$$\eta_t^{k,l_i} = \frac{e^{l_i.\overline{s_{i,t}^k}}}{\widehat{S_t^{k,l_i}}} \tag{7.2}$$

and

$$\widetilde{S_t^{k,l_i}} = \widetilde{S_{t-1}^{k,l_i}} + e^{l_i.\overline{s_{i,t}^k}}; \widetilde{S_0^{k,l_i}} = e^{l_i.\overline{s_{i,0}^k}} \tag{7.3}$$

In our 2D example, each desired sensation is bounded by four extreme desired sensations which define a quadrilateral delimiting a zone of reward or punishment, see Fig. 7.6. We use eight extreme desired sensations defining a hexahedron in three

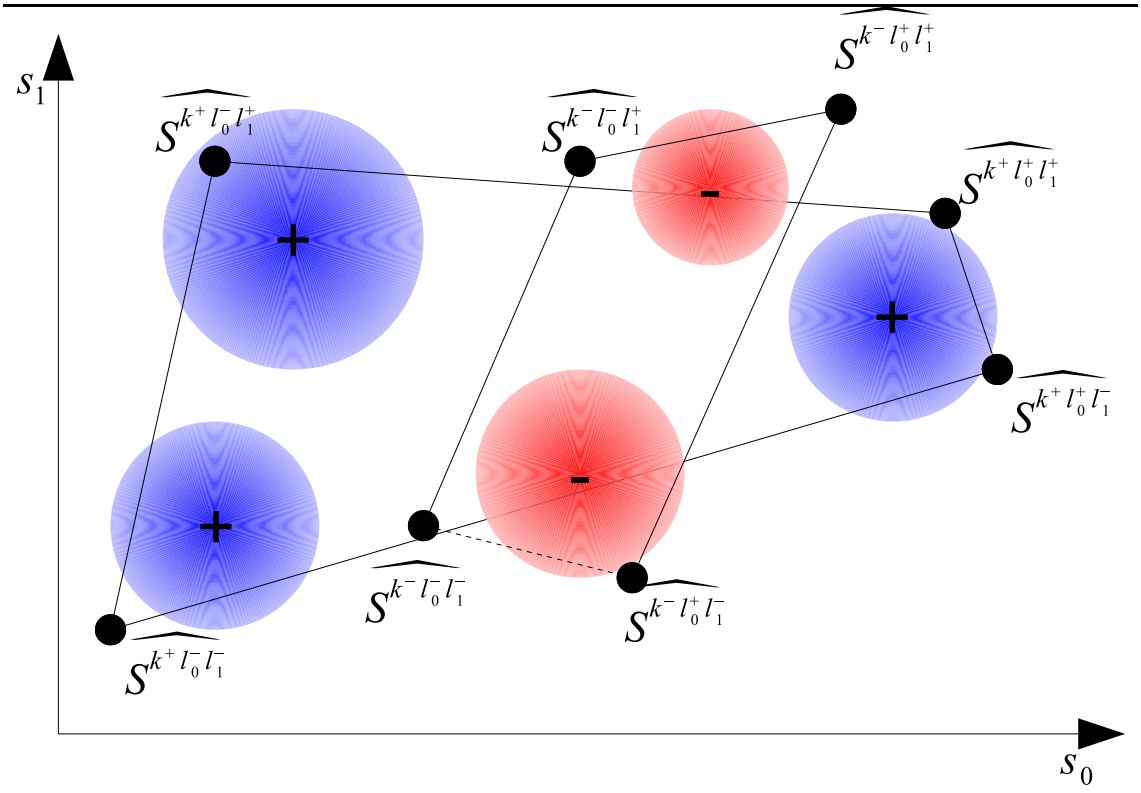


Figure 7.6: Quadrilaterals created by extreme desired sensations bounding zones of reward and punishment.

dimensions, 16 extreme desired sensations in four dimensions and so on. It may not be sufficient to describe all the possible distributions of zones of reward and punishment, therefore, as we proposed in Sec. 7.1.1, we can refine the intersections of the different zones using a second level of extreme desired sensations. However, once we have defined these zones using extreme desired sensations, the agent needs to evaluate in which zone it is. In two dimensions, we can compute the surface of the triangle defined by the current sensation and two extreme sensations using the half of a mathematical function: the *determinant* see Fig. 7.7.

The determinant is a function that can be applied in any number of dimensions

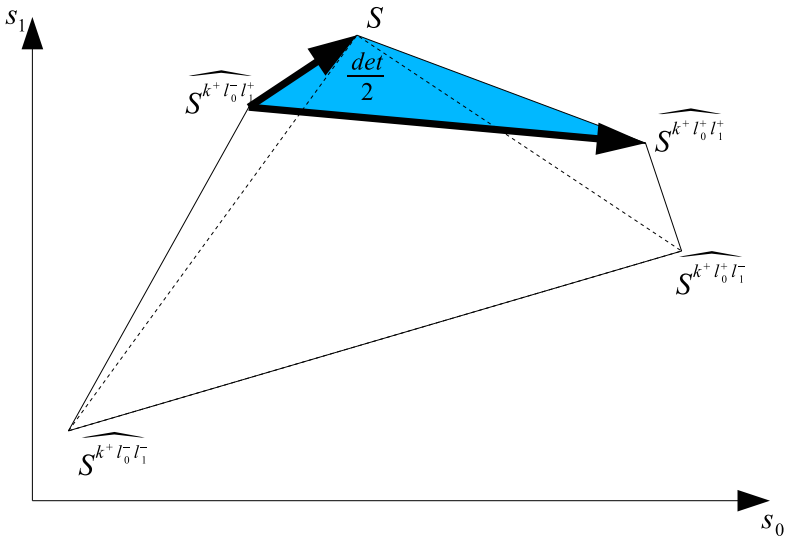


Figure 7.7: Surface between the current sensation (S) and two extreme desired sensations ($\hat{S}$) evaluated using the determinant.

and the sign of the determinant can be used to specify if a sensation is inside or outside a zone. Figure 7.8 illustrates a one dimensional function evaluating the distance of a sensation to a zone defined by extreme desired sensations. Figure 7.9 illustrates a two dimensional case, the 2D function is the sum of the exponential of the determinant computed for each side of the quadrilateral delimiting the zone.

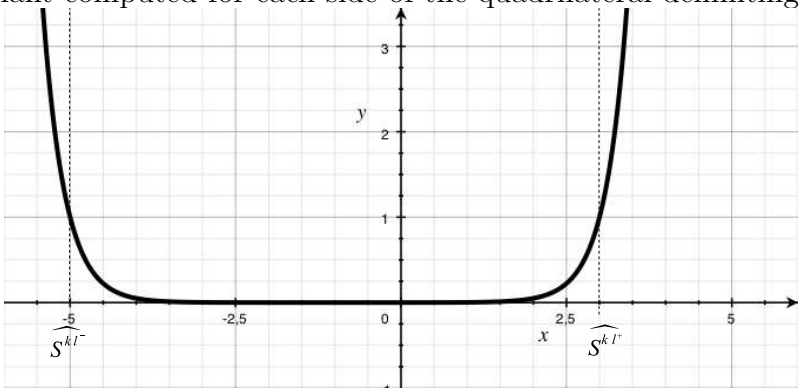


Figure 7.8: One dimensional function determining if a sensation is within a interval of extreme desired sensations ($\widehat{S}$).

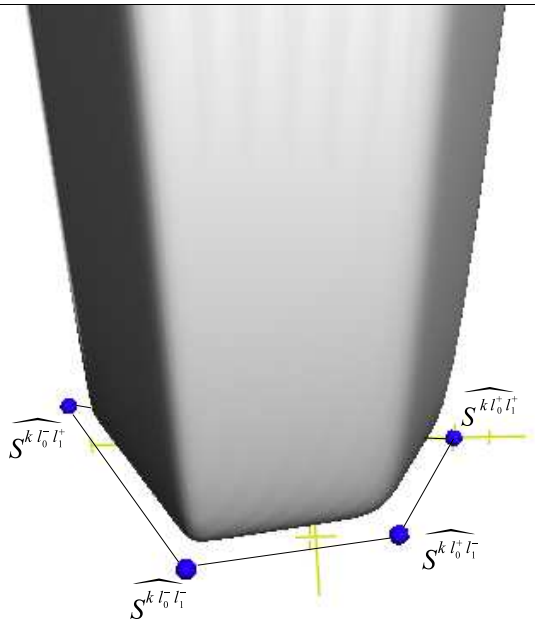


Figure 7.9: Two dimensional function determining if a sensation is within a quadrilateral of extreme desired sensations ($\widehat{S}$).

7.1.4 Complexity

The first limits we could see with this architecture is the increasing complexity with the number of dimensions. Actually, we need to compute a large number of desired sensations and extreme desired sensations with multiple parameters. However, many problems in artificial intelligence concern navigation in two dimensions or movements of effectors usually arms or legs with about five degrees of freedom. The number of extreme desired sensations needed to bound the desired sensations is 2^n where n is the number of dimensions, and the calculation of the determinant needs n^3 operations. Therefore, we need $2^n n^3$ operations in function of n the number of

dimensions. For two dimensions, this represents 24 operations[1] , which is not a problem even for small processors; for 5 dimensions, this represents 4000 operations which it still accessible. For 10 dimensions, this represents about a million operations which is still possible with modern computer, as it is about the number of operations needed to treat classical images of one megapixel.

We guess that the practical limits are around 10 dimensions, however, we saw we may not need a very high number of dimensions, and there are possibilities to reduce them. For example, by alternatively freezing degrees of freedom (Lungarella and Berthouze 2002), using only the main significant dimensions with principal or independent component analysis (PCA or ICA) (Calinon and Billard 2005), or setting a priori the dimensions which can be considered independently.

An other issue is to store and manage the multiple desired sensations and extreme desired sensations varying on different axes $(\gamma, k, l, level, ...)$. However, even if the notation we used is quite complex, the multiple axes are easy to handle using the notion of hypercubes that we have defined in Appendix C

7.1.5 Imitation

In Chap. 5 we have proposed a new vision of low-level imitation, where imitation is interpreted as an amplification of novelty and not a reduction of difference between the actual sensations and what is expected (Andry et al. 2003, Demiris and Johnson 2003). By novelty we mean, the difference between the actual sensation and familiar sensations. We have illustrated it with a simple behavior where the robot had to move forward and backward in symmetry with the experimenter. However, this

[1] An operation can correspond to many basic instructions, but this number of instructions does not depend on the number of dimensions.

behavior may not be interpreted as low-level imitation, and a behavior of following someone can be interpreted as low-level imitation (Hayes and Demiris 1994) whereas we interpreted it as a behavior of seeking for stability.

The ambiguity comes with the fact that it depends on which sensation we are referring to. If we consider the absolute position of the robot, in the room, following the experimenter can be seen as imitation, as it performs the actions changing its sensation of position in the room. If we consider the sensation coming from the frontal sensors of the robot (vision or sensation of distance) following the experimenter makes it have the same sensation and therefore it is seen as a behavior of seeking for stability. Learning and focusing on the relevant sensation may be an issue but at the same time, it can explain why we do not imitate the same thing all the time, and be a good way to move between nests of learning (Kaplan and Oudeyer 2004). Once something is familiar the agent will stop amplifying it, therefore the agent will stop imitating, and try to explore by focussing on other sensations.

In another context, imitation by amplifying novelty can be more explicit. Let us imagine an arm acting on two dimensions, along two axes, if an experimenter creates unexpected activity on an area of it visual field, and the agent tries to amplify this activity, using its sensory-motor association it will move in order to create activity in the same area, therefore it will seem to imitate the experimenter. The result would be similar to the experiment of (Andry et al. 2003) or (Demiris and Johnson 2003) in one dimension, however, in our case the interpretation is different and we do not need to move the camera to switch between explorative or imitative behaviors. In all our architectures, we have hardcoded the sensory-motor association because in animals and even humans, many low-level behaviors are innate. Moreover, we assumed that during the explorative behavior classical learning methods allow the

agent to learn the associations between actions and sensations. Actually, in recent work done with Antoine Hiolle (Hiolle, Cañamero and Blanchard 2007) our Koala robot was able to learn which actions to execute in order to follow an experimenter. At the beginning the robot may be wrong (e.g. move backward instead of forward when the experimenter is moving away) but after some trials and errors, it executes the appropriate actions.

We have exposed our main ideas to go further and to fix some of the technical issues our architecture may raise. Now to resume our chapter and to address the points we have just raised, we propose a new global architecture called the *Perceive, Appraise and Act* architecture improving the Per-Ac (Gaussier and Zrehen 1995) model.

7.2 Perceive, Appraise and Act Architecture (PAA)

7.2.1 "Perceive"

We have proposed to not define events in advance (i.e. we do not cluster sensations) therefore the agents have to categorize events by themselves. This process is similar to self organizing maps and it is in fact a particular case but here all the points (desired sensations and extreme desired sensations) are initialized with the same values (the first sensation) and the differences in their evolution are only due to the differences in their parameters $\gamma, k, l, level, ...$, not in their initialization.

Therefore, one of the main part of the architecture is the one which categorizes sensations. Using many extreme desired sensations this structure defines many zones of sensations associated with high or low well-being, high or low familiarity, more or less recency ... This structure called the "Perceive" module which defines zones

of sensations, allows the agent to interpret its current sensation and then perceiving its environment. The generated perception can then be sent to the "Act" module to generate an appropriate action using sensory-motor associations.

7.2.2 "Act"

The "Act" module uses the sensory-motor associations to generate actions corresponding to the perception sent by the perceive module. The association of the perceive and act modules (see Fig. 7.10) is equivalent to a Per-Ac architecture as it is defined in (Gaussier and Zrehen 1995). With this approach, the perception of an

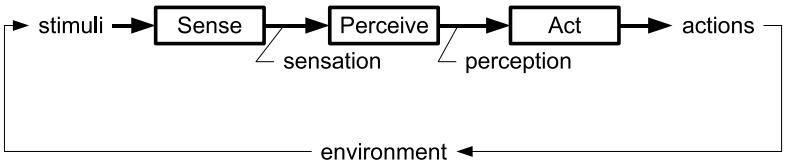


Figure 7.10: Scheme of a Per-Ac architecture.

object through the sensations is characterized by the set of actions an agent does in presence of the object. For example, the perception of an object by an agent which has to move to the object corresponds to the set of associations between sensations produced by the presence of the object and the actions the agent has to execute. "Perception [is seen] as a dynamical sensory-motor attraction basin" (Maillard et al. 2005).

7.2.3 "Appraise"

The Per-Ac architecture allows agents to react to stimuli as if they recognized objects even if they do not have high level processes. Chapter 3 illustrates how the

behavior of a robot following the movement of a rotating drum can be implemented with a Per-Ac architecture. The Perceive module generates the perception of the unwanted movement of the drum by evaluating the difference between the current visual sensation of movement and the goal sensation, which is in this case to not have sensation of movement. Therefore the Act module receiving the undesired perception of movement sends commands to the wheels in order to follow the movement of the drum and therefore inhibits the visual sensation of movement.

However, we have shown in Chap. 4 and 5 that the goal sensation should not be constant, but should depend on the history of the agent. We have defined the notion of desired sensations allowing an agent to learn which sensations generate high well-being, or at the opposite low well-being, which ones are familiar and which ones are recent, etc. Therefore, depending on which desired sensation the agent uses as goal sensation, the agent can generate many different perceptions, but as we have already seen, in order to act, the agent has to select a desired sensation to reach or avoid (in the particular case of avoided sensations predicting low well-being).

We have already proposed several ways of selecting a desired sensation, especially using the notion of comfort composed of well-being as endogenous factor and affect as exogenous factor. Even once a desired sensation has been selected, the actions executed to reach this sensation can be more or less activated (as we did in Chap. 4) or even be inverted (as we did in Chap. 5 for exploration or imitation). Therefore to complete the Per-Ac architecture, with the elements we have developed we need to add a new module which will evaluate or appraise the situation in order to select an appropriate desired sensation and to modulate the action. We therefore present the global Perceive, Appraise and Act (PAA) architecture Fig. 7.11.

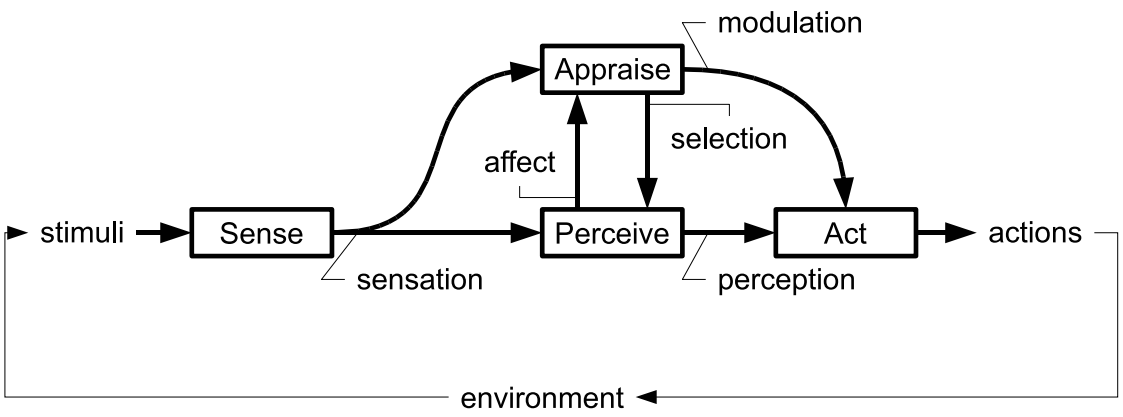


Figure 7.11: Scheme representing the global Perceive, Appraise and Act (PAA) architecture we have developed.

Chapter 8

Summary and Perspectives

'It's hard to explain,' said Mr Anderson , 'but it will be easy to see. The dog will really love you. Robutt [a companion robot] *is just adjusted to act as though it loves you.'*

'But, dad, we don't know what's inside the dog, or what his feelings are. Maybe it's just acting too.' — Isaac Asimov in "A Boy's Best Friend" (1974).

8.1 Summary

In this thesis, we have explored ways of increasing the autonomy of agents, and we have oriented our research towards imitation as experiences suggest that it is a great way of increasing autonomy. In animals, humans or robotics, imitative behaviors are useful for acting, learning and communicating. We have explored in Chap. 2 different studies carried out on imitation and correlated developed theories. Many

architectures have been developed to better explain imitation processes in nature and have been implemented to increase quality (essentially in terms of reproducing relevant actions with accuracy) of imitation in robotics. Nevertheless, for agents to be autonomous in their choices of actions for survival, learning, communication and accomplishment of specific tasks, we need architectures where imitative behaviors are one of the components, which must be well integrated with the other necessary behaviors like seeking for stability, exploration or exploitation. Moreover, the use of imitative behavior (i.e. whether to imitate) should also depend on the interactive partner as some "models" may be "noxious", or bad teachers.

We have adopted the *animat* approach to artificial intelligence (Guillot and Meyer 2001), a bottom-up approach which seeks to understand animal or human intelligence by simulating and understanding animal-like behavior at a simple level from which functionality and complexity are added incrementally (Wilson 1991). We took inspiration on what is observed in nature for two reasons, a) it gives us good hints to solve our problems, and b) attempting to reproduce a natural phenomenon helps us to understand it better. In nature, emotion and affect have strong influences on imitative behaviors and reciprocally, imitation and synchronization influence affect and emotion (Heyes 2001, Trevarthen et al. 1999, Hatfield et al. 1994). Therefore our idea has been to study how the notion of affect can be used to modulate imitative behavior in robotics as seems to be the case in nature.

We first developed a programming framework appendix. C in order to be able to more easily design, monitor and adapt architectures, even during ongoing experiments. This was essential as we use a "bottom-up" approach. We want to be able to take advantage of opportunities coming from unexpected properties that otherwise would be considered "bugs" without a close and dynamic interaction between the

physical environment and the internal architecture of the robot.

Then in Chap. 3 we proposed our first architecture, using our "bottom-up" approach, to allow a robot to increase its synchronization during a task of following a moving target. However, in addition to synchronization being important for emotional contagion (Hatfield et al. 1994) we still had the issue of making a robot autonomously decide who and when to imitate.

We wanted to use affect to modulate imitative behaviors, and a possibility could have been to create affective bonds between agents and to modulate imitation depending on the value of these affective bonds. However, it would have not fit with a "bottom-up" approach which precludes high-level or abstract cognitive functions and does not have notion of "others". We have therefore proposed in Chap. 4 a solution, based on biologically plausible rules, to model affective bonds without cognitive function of high level. This is based on learning what we have defined as "desired sensations". These represent the raw inputs from the sensors (sensation) that an agent should try to reach. In trying to reach these desired sensations, our robot has reproduced a behavior observed in nature known as "imprinting phenomenon" where newborn birds follow the first thing (usually the mother) they see (Lorenz 1935). We then improved the process to allow the desired sensations to be modified depending on stimuli considered as rewards. We introduced the use of multiple simultaneous time scales in the learning process of the desired sensations allowing an agent to continuously pass from an "imprinting" behavior to a more classical learning behavior. This follows the observations of (Bateson and Barron 2000) who noticed that even though imprinting and learning are different processes they are still correlated. However, we have explained that without external stimulation the agents will not explore the environment and therefore will not learn.

In the following chapter, Chap. 5 we proposed to generate explorative behaviors using ideas based on works such as (Kaplan and Oudeyer 2004, Steels 2004, Oudeyer and Kaplan 2004). We have started with the issue of exploration rather than imitation for two reasons: a) we wanted to implement imitative behavior within the framework of other behaviors, and for an autonomous agent, learning and thus exploring is probably more important than imitating; b) we decided that within our approach, explorative behavior is a prerequisite for imitative behavior. We have also shown that it can be useful that an agent can act in order to amplify the variation of sensation if the agent receives a reward and inversely, act in order to reduce this variation if the agent receives a punishment. Similarly, acting in order to amplify the variation of sensation during exploration is a way of increasing the explorative process, whereas acting in order to decrease the variation decreases the explorative process. Therefore we used affect and the need of exploration to modulate the explorative process by acting in order to amplify or reduce the variation of sensation. The important point, is that if an agent induces a variation of sensation to another agent, depending on its affect, the second agent amplifies this variation and produces low-level imitative behavior or inhibits this variation and avoids interaction—this may be a step toward turn-taking. We therefore proposed that low-level imitation can be seen as an emergent property resulting from the balance using affect of behaviors such as seeking for stability, exploration and exploitation. Moreover, with this method imitative behaviors can be integrated within architectures providing other useful behaviors for autonomous agents and imitative behaviors are modulated by affect, as it is the case in nature (Heyes 2001, Hatfield et al. 1994). It is useful as agents should only imitate other agents they had good experiences with—they should not copy "bad" teachers.

One limitation is that this architecture can only handle simple cases, for example an agent can memorize only one kind of sensation. However in Chap. 6 we have extended our work to show some possible similarities with reinforcement learning, and we have proposed to use this architecture to make agents learn to distinguish between sensations associated with rewards and punishments.

Finally, in Chap. 7 we have proposed many solutions non-implemented yet to improve our architecture, especially in dealing with several rewards and punishments in several dimensions.

8.2 Perspectives

The idea that imprinting should be used in order "to learn new goals and ideals" has also been proposed by Marvin Minsky in 2006 (Minsky 2006, pp42). He came to this idea by using a philosophical approach whereas we came to the same idea in order to solve technical problem in robotics. Therefore, we will study how far, and useful parallel we can make between philosophical works and our robotics work. A first point is that "caretaker" does not always define exactly what we refer to; an agent can be imprinted with other agents which do not necessarily take care of it—even though it is generally the case in our epigenetic approach . Minsky encountered the same difficulty, and thus proposed to define a new word corresponding to this property: "imprimer". His definition is that "an *imprimer* is one of those persons to whom a child has become attached".

We would also like to develop the process of selecting the desired sensations—"Appraise" module of the general architecture presented in the discussion—as it a keystone of our architecture. Moreover, we can imagine more desired sensations

with more parameters. One of these parameters should be the potentiality of the agent to reach a desired sensation. Agents have different empowerment in different situations, therefore, they should also take account of it. We list the four main notions the Appraisal module should take into account to select a desired sensation or to modulate actions:

1. The valence associated with the desired sensations (corresponding to positive or negative values of the parameter k),

2. The potentiality or empowerment to reach the desired sensation (our architecture does not handle yet this parameter),

3. The arousal which modulateq the amplitude of actions (corresponding to Mc),

4. The familiarity or predictability of the sensation (corresponding to small values of the parameter k).

Psychological notions of emotion seem well adapted to help us understand and develop more efficient Appraisal modules. This is especially the case of the appraisal theory of emotion developed by Scherer (Scherer 2001, Scherer 1984). Moreover, although the emotional space is usually represented using two components, valence and arousal, a recent study (Roesch, Fontaine and Scherer 2006) shows that we can in fact extract four main components to better describe the emotional space (valence, potency, activation, and unpredictability) see Fig. 8.1). This is very important because they correspond to the main notions we need to select desired sensations. Therefore more studies on these works may help us improve our architectures and maybe we could contribute to better understand and model psychological emotional processes.

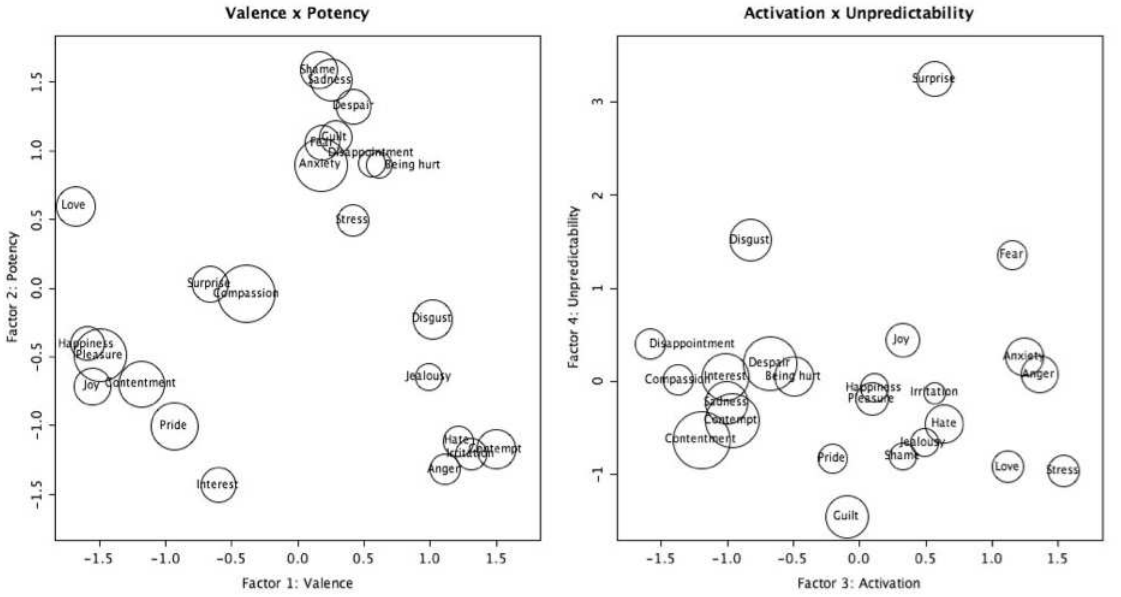


Figure 8.1: Four main components describing the emotional space (Roesh, Fontaine and Scherer 2006) reproduced with permission. Interestingly, these components correspond to those we need to select desired sensations.

Another domain we would like to develop is human-robot interaction. Although the behaviors that we have implemented are simple, it would be interesting to see if our architecture can be useful to develop companion robots (Dautenhahn et al. 2005) or therapeutic robots (Billard et al. 1998) as the possibility of adaptation to the interacting human should be a great advantage.

In the discussion we have stressed the fact that our architecture has some interesting properties to solve problems related to reinforcement learning, therefore we should do more work to improve these capabilities and do quantitative comparisons with classical algorithms used in reinforcement learning problems.

Finally, we would like to further develop our software, especially the user interface in order to make it more usable.

The main scientific points to remember are:

- the possibility to make agents simultaneously learn at different time scales,
- the new vision of low-level imitation—where low-level imitation is seen as amplification of novelty—,
- and the use of affect in order to generate and balance behaviors of seeking for stability, exploration, exploitation and imitation.

Appendix A

Glossary

affect (Af): immediate subjective (based on exogenous factors, external stimuli) evaluation of a situation (expectancy of well-being) without direct or logical explanation.

agent: human, animal, or actor of a simulation, which needs to act in its physical or virtual environment to keep its internal states close to ideal values.

apathy (Ap): lack of motivation to stop or continue an ongoing action.

avoided sensation ($\overline{S^{k^-}}$): sensation associated with low well-being as opposition to a positive desired sensation; agents should avoid them.

behavior of seeking for stability : behaviors where agents act in order to minimize variations of their sensation.

comfort: global evaluation of the "goodness" of a situation based on the evaluation of the exogenous factors (affect) and endogenous factors (well-being).

desired sensation ($\overline{S}$): sensation associated with high or low well-being, by extension it can design a positive desired sensation.

extreme desired sensation ($\widehat{S}$): bounding limit of the value of a desired sensation.

goal sensation: selected desired sensation to reach .

homeostatic control: actions are driven in order to keep some state in a range of values around a ideal value.

hypercube: generic representation of variables, vectors, matrix, cubes etc. with as many dimensions as we need.

motivation to continue (Mc): motivation to continue and amplify an ongoing action, or inversely (when negative) to stop an ongoing action.

openness to the world (Op): motivation to interact with the environment.

perception: interpretation of the ongoing action or affordance raised by a sensation.

pleasure (Pl): value representing the variation of comfort.

positive desired sensation ($\overline{S^{k^+}}$): sensation associated with high well-being as opposition to a avoided sensation; agents should try to reach them.

positive reward: stimulus increasing well-being.

punishment: stimulus decreasing well-being, as opposition to a positive reward.

reward: stimulus increasing or decreasing well-being, by extension it can design a positive reward.

sensation (S): set of raw data coming from the sensors.

well-being (Wb): value representing the proximity of internal states of an agent to their ideal values.

Appendix B

Publications and Dissemination

Several contributions developed in this thesis have published in, or submitted to, relevant conferences and journals. We have also disseminate our works on TV, newspaper and radio.

B.1 Publications

Chapter 3 .

- Blanchard, A.J. and Cañamero, L. (2005). **Using visual velocity detection to achieve synchronization.** *In Y. Demiris, K. Dautenhahn, and C.L. Nehaniv (Ed.), Proceedings of the AISB'05 Third International Symposium on Imitation in Animals and Artifacts. Hatfield, UK.*

Chapter 4 .

- Blanchard A.J. and Cañamero L. (2005). **From Imprinting to Adaptation: Building a History of Affective Interaction.** *Fifth Interna-*

tional Workshop on Epigenetic Robotics (EPIROB05), Nara Japa, Lund University Cognitive Studies, vol. 123, pp. 23–30.

- Blanchard A.J. , Cañamero L. and Nadel, J. (2006). **Attachment Bonds for Human-Like Robots.** *International Journal of Humanoïd Robotics. Vol. 3, Issue 3 (September 2006) pp. 301–320*

Chapter 5 .

- Blanchard, A.J. and Cañamero, L. (2006). **Modulation of Exploratory Behavior for Adaptation to the Context.** *Biologically Inspired Robotics (Biro-net) in AISB'06: Adaptation in Artificial and Biological Systems, Bristol, UK, Vol. 2, pp. 131–139*
- Blanchard, A.J. and Cañamero, L. (2006). **Developing Affect-Modulated Behaviors: Stability, Exploration, Exploitation or Imitation ?** *Proc. of the 6th Intl. Wksp. on Epigenetic Robotics (EPIROB06), Paris, France.*
- Blanchard, A. and Cañamero, L. (2007). Développement de Liens Affectifs Basés sur le Phénomène d'Empreinte pour Moduler l'Exploration et l'Imitation d'un Robot. Review Enfance, N. 1/2007, pp. 35–45.

Chapter 6 .

- Blanchard, A.J. and Cañamero, L. (2006). **Anticipating Rewards in Continuous Time and Space Without Discretisation.** *Third Workshop of SAB on Anticipatory Behavior in Adaptive Learning Systems (ABiALS06), Roma, Italy.*

Chapter 7 .

- Hiolle, A, Cañamero, L. and Blanchard, A.J. (2007). **Discovering the Caretaker: A Developmental Approach.** *Affective Computing and Intelligent Interaction, (ACII07), submitted.*

B.2 Dissemination

TV .

- **Programando emociones.** (January 2007). *Redes, TVE.* National Spanish TV.
- **British Satellite News** (March 2007) English Satellite TV.
- **Discovery Channel** (to be diffused) International TV.

Newspapers .

- **Emotion robots learn from people.** (2007). *BBC News 24.*
- **Audio Slideshows.** (March 2007). *St Alban Observer.* Local newspaper.

Radio .

- **Leading edge.** (1st March 2007) *BBC radio 4* National English Radio.

Appendix C

Programming framework

C.1 Introduction

Working with real robots using a bottom up approach makes us solve problems or take advantage of properties of the physical world. We need to see exactly which are the properties of the behavior coming from the physical parts of a setup and the properties due to software and architectures. All architectures are therefore implemented using software that we have developed to interface with different kinds of robots and to provide useful features to analyze and modify the architectures in real time.

We can summarize our different needs as follows. The software has to:

1. be fast and light to be able to compile in small processors, for embedded robotics;

2. be very dynamic in order to quickly change parts or parameters of an architecture—setup in robotics is time consuming;

3. allow us to easily check internal states of an architecture. Wrong behaviors do not always come from wrong architectures; in robotics many problems occur for technical reasons. For example, low battery, light changing conditions, interferences are the kind of problems I met during my experiments;

4. easily interface with robots, sensors, effectors and simulators;

5. analyze behaviors and store events from experiments.

We then studied different software or languages we can use to implement our architectures noticing the advantage and inconvenience of each one. Some were not stable or finalized when I started my PhD work but it is interesting to know and understand the possibilities, and prepare future work.

C.2 Existing Software and Languages

C.2.1 Statistical Software (R and Matlab)

R (R Development Core Team 2005) and Matlab (www.mathworks.com) are very interesting because they are easy to code and offer many interesting libraries for analyzing data (generation of graphs and statistical analysis) or controlling agents. However, they are slow, since they are interpreted languages, and not adapted at all for real-time control. For example, they are slow to draw a graph and according to our third request in Sec. C.1, we need to be able to check the internal values of an architecture in real time.

C.2.2 Eyes-web

Eyes-web (www.eyesweb.org) is a recent project developed by the InfoMus Lab in Genova (Italy). Its main interest is to graphically display architectures by applying different filters and functions on data coming from different inputs like microphones, cameras, keyboards, etc. These architectures can be described in a text file in the universal xml format or using a graphical user interface. If the functions or filters we need do not exist we can program them in C++ and include them in our architectures. The problem is that there is no interface to control robots, and eyes-web only works on Microsoft Windows. Moreover, the resulting architecture cannot be embedded in a robot.

C.2.3 Leto and Promethe

Leto and Promethe (perso-etis.ensea.fr/~andry/promethe/index_en.html) are programming frameworks written in C for linux by the neuro-cybernetic group of ETIS in Cergy (France) allowing users to respectively design and simulate neural networks. They provide several neural groups that we can reuse and we can also program our own neural groups in C. Moreover several devices such as a video-cards, serial ports, Koala robots (www.k-team.com), etc. can be controlled. The main problem of this software is that it is still in development, it is complex and difficult to tune to our own needs. Moreover, it only works on linux even if it is quite easy to adapt to other platforms because it is written in plain C and uses a multi-platform graphical library GTK (www.gtk.org). The code cannot be embedded in a robot and the separation between design and simulation make it impossible to dynamically modify an architecture during an experiment.

C.2.4 URBI

URBI (Universal Real-time Behavior Interface, www.urbiforge.com) is a recent interesting solution to interface robots. It is not adapted to design complex architectures and cannot be used alone; however it is an interesting option that complements any robot controller in order to provide a good generic interface with different kinds of robot. It can treat commands sequentially but it is especially well adapted to work asynchronously with a good management of events. Although its functions are limited it is very suitable to link functions with other languages like C, java or matlab ...

C.3 Our software

C.3.1 Language

All software presented above have interesting features; however, none of them really suits our needs enumerated in section C.1. To implement, develop and test our architectures, we will therefore develop our own program. In order to retain the pre-existing interesting features and not spend too much time in reprogramming them we tried to make our program as generic as possible for three main reasons.

1. Reusing already existing code
2. Integrating our code in other existing programs
3. Making our programs work on various platforms and systems (robot environments are usually specific).

For almost all processors and systems, there exists a C compiler which is very well optimized. This is the case of the robots like Hemisson and Koala (www.k-team.com), Lego, AIBO, etc. Moreover all the previously mentioned software is easy to interface with C programs and many libraries exist in C. Therefore, we have programmed the basic modules of our application in plain ANSI C, which makes it compile everywhere without needing virtual-machine or exotic libraries. These basic modules allow us to run any architecture everywhere but not to modify or monitor them easily and dynamically; these modules can also access specific modules which provide the functions for particular hardware or links to other software.

In order to modify architectures without needing to recompile the program, which is quite a long operation, or even modify the architectures during an experiment, we have developed a simple script language. A script can be stored in a text file, accessed and modified through a text interface working with any simple net such as telnet or ssh. We have also developed a more sophisticated, user friendly and powerful graphical user interface which needs to be executed on a computer. The overall structure is presented in Figure C.1 and in the next section we will present our script, our basic generic data structure called *hypercube*.

C.3.2 Script

The role of a script interpreter is to make it possible to dynamically design an architecture without needing to modify the source code of the program. Reading and executing a script consume more memory and processor time. This is not a problem with modern computers; however for complex and demanding architectures or for small embedded systems, a fast and light program can be necessary. That is why we use the exact syntax of the C language in the script. On top of avoiding the need

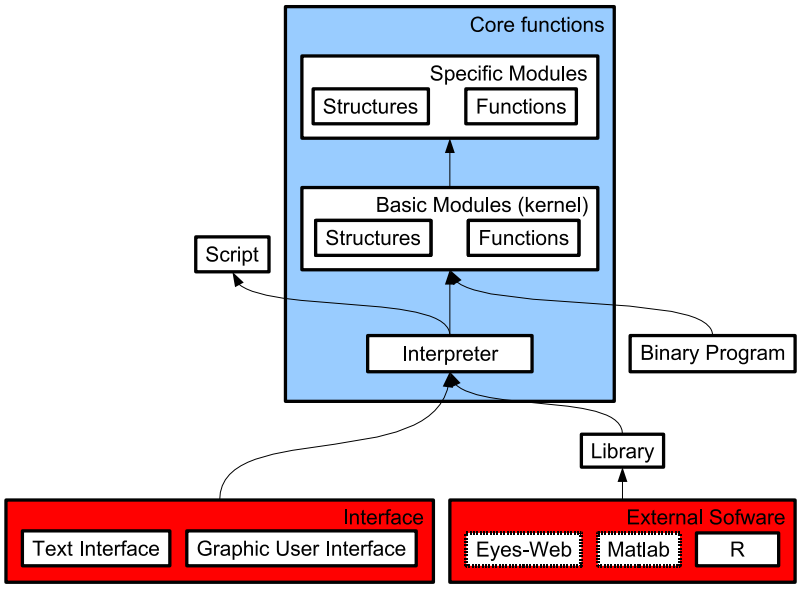


Figure C.1: Relations between the different parts of our application

to define a new syntax which is long to design and long to learn, we can use on the scripts the pre-compiler of the C. The main point is that it makes the architectures easy to embed in a C program. Actually, once we have developed a script with all the opportunities to modify it easily, we can copy it inside the source of a program, including the basic modules, and compile it. We obtain a dense optimized program running the previously defined architecture in the script; the drawback is that we cannot modify it anymore without recompiling all the program.

The structure of the script is a list of functions either creating and initiating new structures (allocating memory) or modifying the values of already created structures. We present an example of a script reading the value of the sixteen sensors of a Koala robot plugged into the serial port COM1 of the computer. We will explain what a "hypercube" is in the next subsection. This script can either be loaded and executed by the interpreter or be included in the source of a program to be compiled and

executed on a specific system.

```
koala_robot = createKoala(‘‘COM1’’);
sensors = createHypercube(1, ‘‘position’’, 0, 16);
getSensor = getKoalaSensor(koala_robot, sensors);
```

The architectures we build are based on perception-action (Gaussier and Zrehen 1995) structures; the general principle is to receive data from sensors—values that we call “sensations”—and process them in order to send commands to the effectors—values we call actions. However, the more possible functions we have to describe an architecture, the more complex it is to use and understand. It is more complex to program because it needs more time to test and debug and the interpreter will be more complex as well. In order to unify the processed data and limit the number of functions to program we have defined a universal data structure called *hypercube*.

C.3.3 Hypercubes

What we call “hypercube” is a generic representation of variables, vectors, matrices, cubes and so on with as many dimensions as we need. They are different from C multidimensional arrays in many ways. Firstly, all the data of a hypercube are stored in only one row of memory, which can save a lot of time and memory since it does not need intermediate pointers to access other dimensions, and secondly it reduces memory allocation. We can create multiple virtual hypercubes accessing the same memory, but with a different access order of the components, without reorganizing the memory structure. Moreover, virtual hypercubes can be a subset of other hypercubes of higher dimension without copying or reorganizing the memory structure. All the values of the hypercubes at each time step can be stored in

hypercubes of higher dimension, then we can analyze all the history of an experiment by printing in a file and exporting the values of these hypercubes. We can also use this history to virtually reproduce an experiment simulating the input from the data acquired during a real experiment. In this case interaction is lost (actions do not affect input) but we can still study evolutions of internal states or memories with different parameters or architectures.

In the same way that we have defined and used a generic unique data structure, we have defined a unique generic computational function. Apart from specific functions dedicated to acquire or export data respectively from and to specific hardware, the computation is based on a unique main function recursively applying basic operators on different dimensions of the hypercubes. This allows us to easily execute statistical functions, simulate neural networks or realize visual processing (filter, convolution). We detail this function *applyUpdate* because it is the main component of the architectures. Depending on the size, the position, the number and order of dimension of the hypercubes, different operations will be executed (scalar product, integration, convolution, ...). The first two parameters are input hypercubes, the third one is the output hypercube, the fourth one defines the operator to apply and the last one is a number allowing us to modify the consideration of the different hypercubes. Each operator has to be defined in the source code which is a constraint but at the same time, it makes architectures more homogenous, and forces the designer to reuse the same generic biologically plausible functions. The designer can apply almost any continuous functions without needing to define a new operator by combining the preexisting operators.

Other languages like Haskell (www.haskell.org) or Sac (www.sac-home.org) use also multidimensional arrays to simplify programing and make more generic pro-

grams but they are usually functional languages, and need a compiler other than than C compiler. They can be used only for strict mathematical functions, whereas we use less conventional functions to better suit needs for neural networks, filtering or image processing. As we will see in the next section, using a unique data structure will greatly simplify the way we can analyze or even modify the data during an experiment without interrupting it.

C.3.4 Graphical user interface (GUI)

The main goal of our software was to enable us to test, control and modify easily an architecture even during an experiment. It is not possible to display and analyze a large amount of data in text mode. That is why we have developed a graphical user interface to use during tests, debugging, demonstrations or programming. Moreover, we take the opportunity to have graphical capabilities to simplify and improve the usability of the software. The drawback is that the graphical interface cannot be used in embedded systems as it needs graphical possibilities and specific graphical libraries.

However, we restrict the number of functions of the graphical interface to the minimum in order to be able to fully use the core functions. The advantage of this is to test, modify and improve the core functions: moreover, each time we do a modification in the core functions, the consequences are immediately effective within the text or graphical interface. As we do not want to spend too much time to realize the graphical interface and we want to keep the possibility of using the software on different platforms, we use a powerful and generic graphical library wxWidget (www.wxwidgets.org) working on Windows, Linux and Mac OS.

The interface works as follows. Each component of an architecture is displayed

in a list representing the current architecture. We can select a component of the architecture in order to execute it once, a precise number of times or indefinitely; we can also know how long it takes to execute it, which is very useful when optimizing. All data (hypercubes) can be visualized and controlled in different manners: tables of values, chars or cursors see Fig. C.2. We can decide to modify them by writing in the tables, or by moving the cursors with the mouse. In the case of a hypercube with only one value, the frequency of a sound can represent the value of the variable. This is useful to interact with the robot without watching the screen and is more appealing during demonstrations.

C.4 Conclusion

We have developed our own software to meet our specific needs: however, we have been careful to a) not develop what already exists and b) always think about the possibility to embed our work in different platforms and software. All the work we will present in this thesis has been designed, tested and executed with the present software. Among the different capabilities potentially implementable in the software, only those relevant to this research have been actually implemented, well developed and tested. The graphical interface has been used on Linux and Windows, and the text interface has been used on Linux, Windows, and Mac. Moreover, we have already built a library to interface our software with the statistical software "R", and this library has been used to simulate and test behaviors of several sub-architectures.

One of the possibilities brought about by the capacity to modify an architecture during its execution time is to make the software modify the architecture itself depend on the events happening during an experiment. We have not exploited this

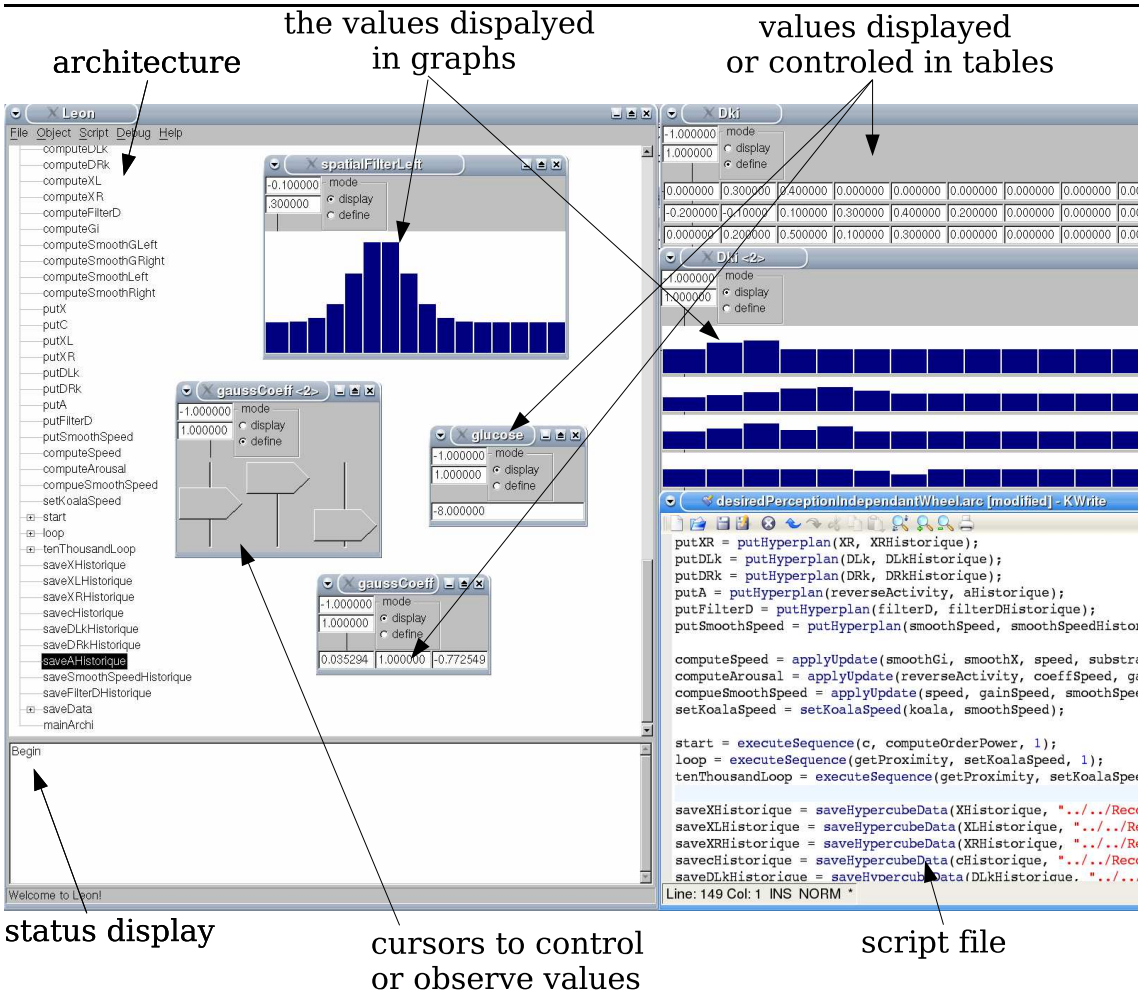


Figure C.2: Screenshot of the software running on Linux. The editor on the bottom right corner is a basic external text editor used to write and display the script in text mode.

possibility yet but we think it could be useful for example to automatically adapt the number of neurons to the complexity of the input data or to simulate the growing of a neural system. The way we thought about this software makes it very generic and depending on further needs, we can interface the software to other projects bringing powerful new functionalities like acquisition functions (e.g. Eyes-web) or

brain-like computing functions (like Promethe). This can also open different ways of cooperation: people developing architectures for robots could find it useful to use this software and we could spread our code in the form of an open project for example. Only the part of the software we really needed and heavily used are robust, efficient and easy to use; few corrections, tests and completions should make it robust and more broadly usable.

Bibliography

Ainsworth, M. D. S.: 1969, Object relations, dependency, and attachment: A theoretical review of the infant-mother relationship, *Child Development* **40**(4), 969–1025.

Alissandrakis, A., Nehaniv, C. and Dautenhahn, K.: 2007, Correspondence mapping induced state and action metrics for robotic imitation, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Special issue on Robot Learning by Observation, Demonstration and Imitation* **37**(2), 299–307.

Alissandrakis, A., Nehaniv, C., Dautenhahn, K. and Saunders, J.: 2005, An approach for programming robots by demonstration: Generalization across different initial configurations of manipulated object, *Computational Intelligence in Robotics and Automation (CIRA2005)*, IEEE, pp. 61–66.

Amari, S.: 1977, Dynamics of pattern formation in lateral-inhibition type neural fields, *Biological Cybernetics* **27**, 77–87.

Anderson, K.: 2003, *A Real-time Affective Computing System for Recognising Facial Expression*, Phd. thesis, University of London.

Andry, P., Gaussier, P. and Nadel, J.: 2003, From sensori-motor development to low-level imitation, *2nd Intl. Wksp. on Epigenetic Robotics*.

Arbib, M. and Fellous, J.-M.: 2004, Emotions: From brain to robot, *Trends in Cognitive Sciences* **8**(12), 554–561.

Arkin, R.: 1998, *Behavior-Based Robotics*, The MIT Press.

Ashby, W.: 1952, *Design for a Brain. The Origin of Adaptive Behaviour*, London: Chapman and Hall, Ltd.

Avila-García, O.: 2006, *Towards Emotional Modulation of Action Selection in Motivated Autonomous Robots*, PhD thesis, University of Hertfordshire (UK).

Avila-García, O. and Cañamero, L.: 2004, Using hormonal feedback to modulate action selection in a competitive scenario, *in* S. Schaal, J. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam and J. A. Meyer (eds), *From Animals to Animats 8: Proceedings of the 8th International Conference on Simulation of Adaptive Behavior*, Cambridge, MA: The MIT Press., pp. 243–252.

Avila-García, O. and Cañamero, L.: 2002, A comparison of behavior selection architectures using viability indicators, *Proc. of the EPSRC/BBSRC International Workshop Biologically-Inspired Robotics*, HP Labs Bristol, UK, pp. 86–93.

Bakker, P. and Kuniyoshi, Y.: 1996, Robot see, robot do: An overview of robot imitation, *AISB96 Wkshp. on Learning in Robots and Animals*, pp. 3–11.

Barron, J., Fleet, D. and Beauchemin, S.: 1994, Performance of optical flow techniques, *Intl. Journal of Computer Vision* **12(1)**, 43–77.

Bateson, P. and Barron, P.: 2000, What must be know in order to understand imprinting?, *in* C. Heyes and L. Huber (eds), *The Evolution of Cognition*, The MIT Press, pp. 85–102.

Bateson, P. B. and Martin, P.: 2000, Sensitive periods, *Design for a Life : How Behavior and Personality Develop*, Simon and Schuster, chapter 8.

Billard, A.: 2000, Learning motor skills by imitation: a biologically inspired robotic model, *Cybernetics and Systems* **32(1-2)**, 155–193.

Billard, A. and Mataric, M. J.: 2001, Learning human arm movements by imitation:: Evaluation of a biologically inspired connectionist architecture, *Robotics and Autonomous Systems* **37**(2-3), 145–160.

Billard, A., Dautenhahn, K. and Hayes, G.: 1998, Experiments on human-robot communication with robot, an imitative learning and communicating doll robot, *Proceedings of the Socially Situated Inteligence Workshop, Zurich*, Technical Report of Centre for Policy Modelling, Manchester Metropolitan University.

Blanchard, A. and Cañamero, L.: 2006, Modulation of exploratory behavior for adaptation to the context, *in* T. Kovacs and M. J. (eds), *Biologically Inspired Robotics (Biro-net) in AISB'06: Adaptation in Artificial and Biological Systems.*, Vol. II, pp. 131–139.

Blaney, P. H.: 1986, Affect and memory: A review, *Psychological Bulletin* **99**(2), 229–246.

Braitenberg, V.: 1984, *Vehicles–Experiments in Synthetic Psychology*, Braitenberg, V.

Breazeal, C. and Scassellati, B.: 2002a, Challenges in building robots that imitate people, *in* K. Dautenhahn and C. Nehaniv (eds), *Imitation in Animals and Artifacts*, The MIT Press, chapter 14, pp. 363–390.

Breazeal, C. and Scassellati, B.: 2002b, Robots that imitate humans, *Trends in Cognitive Sciences* **6**(11), 481–487.

Brooks, R. A.: 1991, Intelligence without reason, *in* J. Mylopoulos and R. Reiter (eds), *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, Sydney, Australia, pp. 569–595.

Butz, M. V., Sigaud, O. and Gérard, P.: 2003, Internal models and anticipations in adaptive learning systems, *in* M. V. Butz, O. Sigaud and P. Gérard (eds), *LNCS 2684 : Anticipatory Behavior in Adaptive Learning Systems*, Springer-Verlag.

Cañamero, L.: 1997, A hormonal model of emotions for behavioral control, *Technical Report 97-06. VUB AI-Lab Vrije Universiteit Brussel*.

Cañamero, L.: 2001, Emotions and adaptation in autonomous agents: A design perspective., *Cybernetics and Systems* **32**(5), 507–529.

Cañamero, L., Blanchard, A. and Nadel, J.: 2006, Attachment bonds for human-like robots, *International Journal of Humanoïd Robotics* **3**(3), 301–320.

Calinon, S. and Billard, A.: 2005, Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM, *Proceedings of the International Conference on Machine Learning (ICML)*, ACM Press, pp. 105–112.

Calinon, S. and Billard, A.: 2007, Learning of gestures by imitation in a humanoid robot, *in* C. Nehaniv and K. Dautenhahn (eds), *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, Cambridge University Press, pp. 153–177.

Capdepuy, P., Polani, D. and Nehaniv, C. L.: 2006, Construction of an internal predictive model by event anticipation, *in* M. Butz, O. Sigaud, G. Pezzulo and G. Baldassarre (eds), *Proceedings of the Third Workshop on Anticipatory Behavior in Adaptive Learning Systems*, LNCS/LNAI, Springer. (in press).

Chesters, W. and Hayes, G.: 1994, Connectionist environment modelling in a real robot, *in* D. Cliff, P. Husbands, J.-A. Meyer and S. Wilson (eds), *SAB94: Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, MIT press, pp. 189–197.

Cos-Aguilera, I., Cañamero, L. and Hayes, G.: 2003, Learning object functionalisites in the context of action selection, *in* U. Nehmzow and C. Melhuish (eds), *Towards Intelligent Mobile Robots: 4th British Conference on Mobile Robotics*, pp. 28–29.

Cover, T. M. and Thomas, J. A.: 1991, *Elements of Information Theory*, Wiley-Interscience.

Dautenhahn, K. and Nehaniv, C.: 2002, *Imitation in Animals and Artifacts*, MIT Press, chapter The Agent-Based Perspective on Imitation, pp. 1–40.

Dautenhahn, K., Woods, S., Kaouri, C., Walters, M. L., Koay, K. L. and Werry, I.: 2005, What is a robot companion - friend, assistant or butler?, *Intelligent Robots and Systems. IROS '05 Proceedings.*, pp. 1192–1197.

Dearden, A. and Demiris, Y.: 2005, Learning forward models for robotics, *International Joint Conferences on Artificial Intelligence*, Morgan Kaufmann Publishers, Inc., pp. 1440–1445.

Decety, J.: 1996, Do imagined and executed actions share the same neural substrate?, *Cognitive Brain Research* **3**, 87–93.

Demiris, Y. and Dearden, A.: 2005, From motor babbling to hierarchical learning by imitation: a robot developmental pathway, *Proceedings of the 5th International Workshop on Epigenetic Robotics*, Lund University Cognitives Studies 123, pp. 31–37.

Demiris, Y. and Johnson, M.: 2003, Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning, *Connection Science* **14**(4), 231–234.

Doya, K.: 2000, Reinforcement learning in continuous time and space, *Neural Computation* **12**(1), 219–245.

Drea, C.: 1998, Social context affects how rhesus monkeys explore their environment, *American Journal of Primatology* **44**(3), 205–214.

Dunn, J.: 1977, Distress and comfort, *in* J. Bruner, M. Cole and B. Lloyd (eds), *The Developing Child*, Fontana/Open Books, chapter Crying, Comfort and Attachment, pp. 67–75.

Ellis, R. and Tucker, M.: 2000, Micro-affordance: The potentiation of components of action by seen objects, *British Journal of Psychology* **91**, 451–471.

Fellous, J.-M. and Arbib, M. (eds): 2005, *Who Needs Emotions?*, Affective Science, Oxford University Press Inc.

Galef, B. G.: 1998, Recent progress in studies of imitation ans social learning in animals, *in* M. Saborin, F. Craik and M. Robert (eds), *Advances in Psychologic Science*, Vol. 2, UK: Psychology Press, pp. 275–300.

Gaussier, P. and Zrehen, S.: 1995, Per ac: A neural architecture to control artificial animals, *Robotics and Autonomous Systems* **16**, 291–320.

Gaussier, P., Moga, S., Banquet, J. and Quoy, M.: 1998, From perception-action loops to imitation processes: A bottom-up approach of learning by imitation, *Applied Artificial Intelligence* **1(7)**, 701–727.

Gibson, J.: 1979, *The Ecological Approach to Visual Perception*, Houghton Mifflin, Boston.

Guillot, A. and Meyer, J.-A.: 2001, The animat contribution to cognitive systems, *Journal of Cognitive Systems Research* **2(2)**, 157–165.

Harlow, H. F.: 1958, The nature of love, *American Psychologist* **13**, 573–685.

Hatfield, E., Cacioppo, J. and Rapson, R.: 1994, *Emotional Contagion*, Cambridge University Press.

Hayes, G. and Demiris, J.: 1994, A robot controller using learning by imitation, *Proceedings of the 2nd International Symposium on Intelligent Robotic Systems*, pp. 198–204.

Heyes, C.: 2001, Causes and consequences of imitation, *Trends in Cognitive Sciences* **5**(6), 253–261.

Hiolle, A., Cañamero, L. and Blanchard, A.: 2007, Discovering the caretaker: A developmental approach., *Affective Computing and Intelligent Interaction, (ACII07).*

Hofsten, C. v. and Rosander, K.: 1996, The development of gaze control and predictive tracking in young infants, *Vision Research* **36**, 81–96.

Holst, E. v. and Mittelstaedt, H.: 1950, Das reafferenzprinzip. wechselwirkungen zwischen zentralnervensystem und peripherie., *Naturwissenschaften* **37**, 464–476.

Johnston, A., Benton, C. and Morgan, M.: 1999, Concurrent measurement of perceived speed and speed discrimination threshold using the method of single stimuli, *Vision Research* **39**, 3849–3854.

Jones, B.: 1977, *Instrumentation, Measurement and Feedback*, McGraw-Hill Book Company.

K-Team: 2002-2006, Hemisson, designed for education, http://www.k-team.com/kteam. Date last consulted: December 2006.

Kahneman, D., Diener, E. and Schwarz, N. (eds): 1999, *Well-being: The Foundations of Hedonic Psychology*, Russell Sage Foundation.

Kaplan, F. and Oudeyer, P.-Y.: 2004, Maximizing learning progress: an internal reward system for development, *in* F. Iida, R. Pfeifer, L. Steels and Y. Kuniyoshi (eds), *Embodied Artificial Intelligence*, LNCS 3139, Springer-Verlag, London, UK, pp. 259–270.

Kaplan, F. and Oudeyer, P.-Y.: 2007, The progress-drive hypothesis: an interpretation of early imitation, *in* C. Nehaniv and K. Dautenhahn (eds), *Models and Mechanisms of Imitation and Social Learning: Behavioural, Social and Communication Dimensions*, Cambridge University Press. to appear.

Kaplan, K.: 2005, Everyday robotics: new visions, new challenges, *sOc-EUSAI'05: Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence*, ACM Press, New York, NY, USA, pp. 57–57.

Kozima, H. and Zlatev, J.: 2000, An epigenetic approach to human-robot communication, *IEEE International Workshop on Robot and Human Communication (ROMAN00).*, pp. 346–351.

Kugiumutzakis, G., Kokkinaki, T., Makrodimitraki, M. and Vitalaki, E.: 2005, Emotions in early mimesis, *in* J. Nadel and D. Muir (eds), *Emotional Development*, Oxford University press, pp. 161–182.

Kuniyoshi, Y.: 1994, Towards physically and socially grounded intelligence, *The Science of Imitation - Real World Computing Joint Symposium*.

Likhachev, M. and Arkin, R.: 2000, Robotic comfort zones, *in* M. G. and S. P. (eds), *SPIE Sensor Fusion and Decentralized Control in Robotic Systems III*, Vol. 4196, Society of Photo-Optical Instrumentation Engineers, pp. 27–41.

Lorenz, K.: 1935, Companions as factors in the bird's environment, *Studies in Animal and Human Behavior*, Vol. 1, London: Methuen & Co., and Cambridge, Mass.: Harvard University Press, pp. 101–258.

Lungarella, M. and Berthouze, L.: 2002, Adaptivity via alternate freeing and freezing of degrees of freedom., *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP '02).*, Vol. 1, pp. 482–487.

Maillard, M., Gapenne, O., Gaussier, P. and Hafemeister, L.: 2005, Perception as a dynamical sensori-motor attraction basin, *in* Capcarrere (ed.), *Advances in Artificial Life (8th European Conference, ECAL)*, Vol. LNAI 3630 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 37–46.

Matarić, M. J.: 2002, Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics, *in* K. Dautenhahn and C. L. Nehaniv (eds), *Imitation in Animals and Artifacts*, MIT press, pp. 392–422.

McOwan, P., Benton, C., Dale, J. and Johnston, A.: 1999, A multi-differential neuromorphic approach to motion detection, *Intl. Journal of Neural Systems* **9**, 429–434.

Minsky, M.: 2006, *The Emotion Machine*, Simon and Schuster.

Mitchell, R.: 1987, A comparative–developmental approach to understand imitation, *Perspectives in Ethology* **7**, 183–215.

Nadel, J., Guérini, C., Pezé, A. and Rivet, C.: 1999, The evolving nature of imitation as a format for communication, *in* J. Nadel and B. G. (eds), *Imitation in Infancy*, Cambridge University Press, pp. 209–234.

Nadel, J., Revel, A., Andry, P. and Gaussier, P.: 2004, Toward communication: First imitations in infants, low-functioning children with autism and robots, *Interaction Studies* **5(1)**, 45–74.

Nehaniv, C. and Dautenhahn, K.: 2002, The correspondence problem, *in* K. Dautenhahn and C. Nehaniv (eds), *Imitation in Animals and Artifacts*, MIT press, pp. 41–61.

Oudeyer, P.-Y. and Kaplan, F.: 2004, Intelligent adaptive curiosity: a source of self-development, *in* L. Berthouze, H. Kozima, C. G. Prince, G. Sandini, G. Stojanov, G. Metta and C. Balkenius (eds), *Proc. of the 4th Intl. Wks. on Epigenetic Robotics*, Vol. 117, Lund University Cognitive Studies, pp. 127–130.

Panksepp, J.: 1999, *Affective Neuroscience*, Oxford University Press, chapter SEEKING Systems and Anticipatory States of the Nervous System, pp. 144–162.

Power, T.: 2000, *Play and Exploration in Children and Animals*, Lawrence Erlbaum Associates, Publishers.

Prinz, W.: 1997, Perception and action planning, *European Journal of Cognitive Psychology* **9**(2), 129–154.

Prinz, W.: 2005, An ideomotor approach to imitation, *in* S. Hurley and N. Chater (eds), *Perspectives on Imitation*, Vol. 1, The MIT Press, chapter 5, pp. 141–156.

R Development Core Team: 2005, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Rescorla, R. and Wagner, A.: 1972, A theory of pavlovian conditioning: Variations in effectiveness of reinforcement and nonreinforcement., *in* A. Black and W. Prokasy (eds), *Classical Conditioning II*, New York: Appleton-Century-Crofts, pp. 64–99.

Richards, J. and Holley, F.: 1999, Infant attention and the development of smooth pursuit tracking., *Developmental Psychology* **35**, 856–867.

Rochat, P.: 2002, Ego function of early imitation, *in* A. Meltzoff and W. Prinz (eds), *The Imitative Mind*, Cambridge University Press, pp. 85–97.

Roesch, E., Fontaine, J. and Scherer, K.: 2006, The world of emotion is two-dimensional, ... or is it ?, *HUMAINE's summer school in Genova*.

Saunders, J.: 2006, *Observational Imitation, Self-Imitation and Environmental Scaffolding in Robotic Systems*, PhD thesis, University of Hertfordshire (UK).

Saunders, J., Nehaniv, C., Dautenhahn, K. and Alissandrakis, A.: 2007, Self-imitation and environmental scaffolding for robot teaching, *International Journal of Advanced Robotics Systemss, Special Issue on Human - Robot Interaction* **4**(1), 109–124.

Scherer, K.: 1984, On the nature and function of emotion: A component process approach, *in* K. Scherer and P. Ekman (eds), *Approaches to Emotion*, Hillsdale, NJ: Erlbaum, pp. 293–317.

Scherer, K.: 2001, Appraisal considered as a process of multilevel sequential checking, *in* K. Scherer, A. Schorr and T. Johnstone (eds), *Appraisal Processes in Emotion: Theory, Methods, Research*, Oxford University Press, New York, pp. 92–120.

Simon, H.: 1967, Motivational and emotional controls of cognition, *Psychological Review*, Vol. 74, pp. 29–39.

Steels, L.: 1994, Mathematical analysis of behavior systems, *in* I. C. S. Press (ed.), *From Perception to Action Conference*, pub-IEEE, Lausanne, Switzerland, pp. 88–95.

Steels, L.: 1995, When are robots intelligent autonomous agents?, *Robotics and Autonomous Systems* **15**(7), 3–9.

Steels, L.: 2004, The autotelic principle, *in* I. Fumiya, R. Pfeifer, L. Steels and K. Kunyoshi (eds), *Embodied Artificial Intelligence*, Vol. 3139 of *Lecture Notes in AI*, Springer Verlag, Berlin, pp. 231–242.

Sutton, R. and Barto, A.: 1987, A temporal-difference model of classical conditioning, *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, pp. 355–378.

Sutton, R. and Barto, A.: 1998, *Reinforcement Learning: An Introduction*, MIT Press.

Thorndike, E.: 1911, *Animal Intelligence: Experimental Studies*, The Macmillan company.

Thorndike, E. L.: 1898, Animal intelligence: an experimental study of the association process in animals, *Psychological Reviw Monographs* **2**(8), 551–553.

Thorpe, W.: 1963, *Learning and Instinct in Animals*, Cambridge, MA: Harvard University Press.

Trevarthen, C., T., K. and Fiamenghi, G.: 1999, What infants' imitations communicate: with mothers, with fathers and with peers, *in* J. Nadel and G. Butterworth (eds), *Imitation in Infancy*, Cambridge University Press, pp. 127–185.

University of Michigan RL Group, Common myths and misstatements about reinforcement learning, http://neuromancer.eecs.umich.edu/cgi-bin/twiki/view/Main/MythsofRL. Last visited date: April 2007.

Užgiris, I.: 1999, Imitation as activity: its developmental aspects, *in* J. Nadel and G. Butterworth (eds), *Imitation in Infancy*, Cambridge University Press, pp. 186–206.

Velasquez, J. D.: 1999, An emotion-based approach to robotics, *Intelligent Robots and Systems. IROS '99. Proceedings.*, Vol. 1, pp. 235–240.

Vogt, S.: 2002, Visuomotor couplings in object-oriented and imitative actions, *in* A. Meltzoff and W. Prinz (eds), *The Imitative Mind*, Cambridge University Press, pp. 206–220.

Watkins, C.: 1989, *Learning from Delayed Rewards*, PhD thesis, King's College.

Wilson, S.: 1991, The animat path to AI, *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, MIT press, pp. 15–21.

Wilson, S.: 1996, Explore/exploit strategies in autonomy., *in* P. Maes, M. Mataric, J. Pollack and J. Meyer (eds), *From Animals to Animats: Proc. of the 4th Intl. Conf. on Simulation of Adaptive Behavior*, MA: The MIT Press, pp. 325–332.

Wolpert, D. and Macready, W.: 1997, No free lunch theorems for optimisation, *IEEE Trans. on Evolutionary Computation*, Vol. 1 of *1*, pp. 67–82.

Zentall, T. and Akins, C.: 2001, Imitation in animals: Evidence, function, and mechanisms, *Cybernetics and Systems* **32**(1-2), 53–96.

Zlatev, J. and Balkenius, C.: 2001, Introduction: Why epigenetic robotics, *Proceedings of the First International Workshop on Epigenetic Robotics*, Vol. 85, Lund University Cognitive Studies, pp. 1–4.