SAFEL - A SITUATION-AWARE FEAR LEARNING MODEL

A THESIS SUBMITTED TO
THE UNIVERSITY OF KENT
IN THE SUBJECT OF COMPUTER SCIENCE
FOR THE DEGREE
OF PHD.

By
Caroline Rizzi Raymundo
December 2017

Abstract

This thesis proposes a novel and robust online adaptation mechanism for threat prediction and prevention capable of taking into consideration complex contextual and temporal information in its internal learning processes. The proposed mechanism is a hybrid cognitive computational model named SAFEL (Situation-Aware FEar Learning), which integrates machine learning algorithms with concepts of situation-awareness from expert systems to simulate both the cued and contextual fear-conditioning phenomena. SAFEL is inspired by well-known neuroscience findings on the brain's mechanisms of fear learning and memory to provide autonomous robots with the ability to predict undesirable or threatening situations to themselves. SAFEL's ultimate goal is to allow autonomous robots to perceive intricate elements and relationships in their environment, learn with experience through autonomous environmental exploration, and adapt at execution time to environmental changes and threats.

SAFEL consists of a hybrid architecture composed of three modules, each based on a different approach and inspired by a different region (or function) of the brain involved in fear learning. These modules are: the *Amygdala Module* (AM), the *Hippocampus Module* (HM) and the *Working Memory Module* (WMM). The AM learns and detects environmental threats while the HM makes sense of the robot's context. The WMM is responsible for combining and associating the two types of information processed by the AM and HM.

More specifically, the AM simulates the cued conditioning phenomenon by creating associations between co-occurring aversive and neutral environmental stimuli. The AM represents the kernel of emotional appraisal and threat detection in SAFEL's architecture. The HM, in turn, handles environmental information at a higher level of abstraction and complexity than the AM, which depicts the robot's situation as a whole. The information managed by the HM embeds in a unified representation the temporal interactions of multiple stimuli in the environment. Finally, the WMM simulates the contextual conditioning phenomenon by creating associations between the contextual memory formed in the HM and the emotional

memory formed in the AM, thus giving emotional meaning to the contextual information acquired in past experiences. Ultimately, any previously experienced pattern of contextual information triggers the retrieval of that stored contextual memory and its emotional meaning from the WMM, warning the robot that an undesirable situation is likely to happen in the near future.

The main contribution of this work as compared to the state of the art is a domain-independent mechanism for online learning and adaptation that combines a fear-learning model with the concept of temporal context and is focused on real-world applications for autonomous robotics. SAFEL successfully integrates a symbolic rule-based paradigm for situation management with machine learning algorithms for memorizing and predicting environmental threats to the robot based on complex temporal context.

SAFEL has been evaluated in several experiments, which analysed the performance of each module separately. Ultimately, we conducted a comprehensive case study in the robot soccer scenario to evaluate the collective work of all modules as a whole. This case study also analyses to which extent the emotional feedback of SAFEL can improve the intelligent behaviour of a robot in a practical real-world situation, where adaptive skills and fast/flexible decision-making are crucial.

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Colin G. Johnson, for his continuous guidance and commitment to my Ph.D. I am also immensely grateful to my co-supervisor, Dr. Patricia A. Vargas, for providing me countless academic opportunities that have made my work possible and visible in many ways. The endless support of Colin and Patricia, as well as their confidence in my work, gave me the courage to endure the challenges of my research topic and persist in the pursuit of my aspirations.

The support of my family was essential and gave me the strength to persist even in the most difficult and stressful times. I thank my father, Hercules F. Raymundo, my mother, Eliene M. Rizzi, and my stepfather, Paulo R. F. dos Santos, for listening, advising, encouraging and always being there for me, even from afar. I thank especially my beloved husband, Fabio Fabris, without whom this work would not be possible. His unconditional support, patience and kindness were crucial during my journey to this accomplishment. He also contributed with numerous technical conversations that gave me brilliant insights and different perspectives over my research. Thank you for never ceasing to believe in me.

I also thank all the 'Brasucas' (they will get it) that I have befriended during my time in the UK, for adding so many great and funny moments to this endeavour and also for bringing a Brazilian colouration back to my life. I thank especially Izabela Barata and Guilherme B. Ferreira, with whom I am immensely thrilled to befriend.

I thank CAPES, the Brazilian research agency that sponsored my Ph.D., for making my research possible. I also thank my supervisory panel at the School of Computing of the University of Kent, composed by Dr. Fernando Otero, Dr. Howard Bowman and Dr. Sally Fincher, for their constructive feedback that contributed to improving my work. I also appreciate the valuable feedback to my seminar talks from the members of the Computational Intelligence Group of the School of Computing. Finally, I thank the members of my examination committee, composed by Dr. Jon Timmis and Dr. Marek Grzes, for accepting the invitation and for their valued assessment and advice.

Contents

List of Tables

List of Figures

Acronyms

LTD Long-Term Depression . . . 45

LTP Long-Term Potentiation . . . 44

NS Neutral Stimulus . . . 43

RBWM Rule-Based Working Memory . . . 77

RHS Right-Hand Side . . . 77

RoboCup Robot World Cup . . . 34

SAFEL Situation-Aware FEar Learning . . . 13

SDD Situation Detection Delay . . . 95

SPL Standard Platform League . . . 132

UA Unconditioned Aversive . . . 92

UR Unconditioned Response . . . 42

US Unconditioned Stimulus . . . 42

WMM Working Memory Module . . . 38

List of Publications Related to this Research

Rizzi, C., Johnson, C. G. and Vargas, P. A. (2016). Improving the predictive performance of SAFEL: A Situation-Aware FEar Learning model. In *25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 736–742.

Rizzi, C., Johnson, C. G. and Vargas, P. A. (2017). Fear Learning for Flexible Decision Making in RoboCup: A Discussion. In *RoboCup 2017: Robot World Cup XXI*, Springer, Lecture Notes in Artificial Intelligence, in press.

Rizzi, C., Johnson, C. G. and Vargas, P. A. (2018). Online Adaptation and Situation Awareness Using SAFEL with Applications to Robot Soccer. *Artificial Intelligence*, under review.

Rizzi, C., Johnson, C. G., Fabris, F. and Vargas, P. A. (2017). A Situation-Aware Fear Learning (SAFEL) model for robots. *Neurocomputing*, 221, pp. 32–47.

Rizzi Raymundo, C. and Johnson, C. G. (2014). An Artificial Synaptic Plasticity Mechanism for Classical Conditioning with Neural Networks. In Z. Zeng, Y. Li and I. King, eds., *Advances in Neural Networks - ISNN 2014*, *Lecture Notes in Computer Science*, vol. 8866, Springer, pp. 213–221.

Rizzi Raymundo, C., Johnson, C. G. and Vargas, P. A. (2015). An Architecture for Emotional and Context-Aware Associative Learning for Robot Companions. In *24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 31–36.

Rizzi Raymundo, C., Dockhorn Costa, P., Almeida, J. and Pereira, I. (2014). An Infrastructure for Distributed Rule-Based Situation Management. In *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, pp. 202–208.

Chapter 1

Introduction

Autonomous robots are gradually becoming part of our every-day life. A few years ago, autonomous robots would be found in industry, military and research applications only. In recent years, however, a variety of domestic autonomous robots has come to light, such as vacuum cleaner robots, toy robots and personal assistants. Smart homes, for instance, would be considered science-fiction a couple of decades ago. By contrast, smart homes are a reality nowadays; these technologies are becoming increasingly common and accessible.

Autonomous robots are ‘intelligent machines capable of performing tasks in the world by themselves, without explicit human control’ (Bekey 2005). These machines may assume different forms and perform a variety of tasks. Regardless of their profile and aims, all autonomous robots are expected to show a certain level of autonomy so as to operate in the real world without any form of external control for a period of time.

This proposition poses an exceptional challenge to robotics, especially for robots operating in dynamic, uncertain or unstructured environments. In such cases, adaptive skills are required for autonomous behaviour to be successfully accomplished because the environment is constantly changing. In addition, robots dealing with uncertain situations, such as socially interacting with humans, may also struggle at successfully exhibiting fully autonomous behaviour. The outcomes of an interaction with a person can hardly be anticipated because humans’ ways to interact can greatly vary from individual to individual, and even within the same individual at different times. Therefore, a robot that is fully dependent on predefined rules of interaction is likely to respond, at some point, in a manner that humans consider to be either inconsistent, incoherent or annoying.

To solve this issue, robot controllers must integrate computational mechanisms that provide adaptive skills. We argue that an efficient way to implement adaptive behaviours is by computationally simulating the brain’s mechanisms of emotion.

Evidence from well-known neuroscience findings (Damasio 1994; LeDoux 1995, 2003; Lewis, Haviland-Jones and Barrett 2010) indicate that emotions are crucial for learning and manifesting survival skills in the great majority of animal species, including humans, by underlying associative mechanisms that bypass regions of the brain responsible for conscious and rational thinking. Emotions play an essential role in brain functions that are vital for the expression of intelligent behaviours, such as adaptation, fast and flexible decision-making, learning, perception and memory.

Nonetheless, the effectiveness of emotional mechanisms for adaptive and flexible decision-making is highly dependent on one's competence to perceive one's situation. For instance, the emotional mechanisms that support survival skills are commonly triggered when the individual perceives danger. If that individual is unable to effectively perceive and detect potential sources of threat in the environment then such emotional mechanisms will not be activated. Therefore, in addition to mechanisms of emotion, a robust mechanism that allows a robot to efficiently and thoroughly perceive its state of affairs is essential for the robot to express adaptive and flexible behaviours.

In the following sections of this chapter, we explore the relevance of emotions for human intelligence, especially in regards to fear learning and memory (Section 1.1). Section 1.2 presents scenarios in the field of autonomous robotics that represent strong sources of motivation to the proposal of this thesis. Section 1.3 elaborates on the requirements of an emotionally and situation-aware based system that meets the needs of autonomous robotics for adaptive skills. Finally, Section 1.4 discusses our research questions, hypothesis and contributions. The outline of the thesis is presented in Section 1.5.

1.1 Emotions

Emotions are mostly taken for granted. We can feel them, observe their onset and experience their effects in our lives and interaction with others. Despite being part of a tacit and universal knowledge among people, the scientific community has not yet reached a consensus when it comes to an ultimate definition of emotions. Notwithstanding the lack of definition, emotions are widely recognized by neuroscientists and psychologists as essential for the proper functioning of several brain functions. In this section, we discuss the relevance of emotions for human intelligence, as well as how they can contribute and serve as an inspiration to improve artificial intelligence.

1.1.1 What are emotions?

What emotions are, what their ultimate purpose is and how they manifest in the brain is a polemical discussion that divides the opinion of many researchers. Emotions are interpreted in a number of ways by neuroscientists and psychologists. Among other interpretations, emotions are commonly viewed as either:

- bodily responses that evolved as part of adaptation and survival skills,
- mental states that result when bodily responses are 'sensed' by the brain,
- ways for an individual to act or express itself,
- unconscious impulses,
- thoughts about the situations in which people find themselves, or
- the building block of a social system, thus happening between rather than within individuals.

Regardless of how emotions are interpreted, their importance for the expression of intelligent behaviour is a consensus among scientists nowadays. However, despite the body of scientific evidence pointing in another direction, emotions are still widely seen as obstacles to logical thinking, as if reason and emotion were part of two distinct systems working separately and unable to collaborate. More than that, unable to coexist. It is a commonly held opinion that reasonable decisions come from a rational and clear mind where emotions are not allowed to interfere.

Hence, it is not surprising that the dictionary (Oxford Dictionaries 2017b) defines 'emotion' as an 'instinctive or intuitive feeling as distinguished from reasoning or knowledge'. Ironically, the latest neuroscience findings indicate that emotions play an essential role in the very mechanisms of rational thinking, such as decision making, perception, learning and memory (LeDoux 1999). This is not to deny that intense emotional states can disturb logical reasoning, but instead to advocate that the lack of emotions can be as harmful to logical thinking as is their exacerbation.

Damasio (1994) discusses cases of patients who partially lost their emotional capabilities, among which stands out the story of an American railroad-construction foreman called Phineas Gage. Gage was involved in a rock-blasting accident in 1848, in which an iron bar completely crossed the left frontal part of his head. Gage suffered a serious brain injury, which partially destroyed his frontal lobe. Against medical expectations, Gage survived without any intellectual or physical damage besides the loss of vision in his left eye. He was able to make calculations, talk naturally and move all parts of his body as before.

However, Gage suffered a behavioural change so intense that his friends and acquaintances could hardly recognize him. Gage became impatient, disrespectful, incapable of sticking to plans and lacking any sense of embarrassment, many times making large use of bad language regardless of the presence of others. Gage's new personality led him to lose most of his social network, including friends, colleagues and jobs.

Gage's behaviour was clearly disadvantageous to himself; yet, he could neither avoid it nor adapt his behaviour by interpreting people's reaction to his attitudes. Damasio (1994) observes that:

> Gage's example indicated that something in the brain was concerned specifically with unique human properties, among them the ability to anticipate the future and plan accordingly within a complex social environment; the sense of responsibility toward the self and others; and the ability to orchestrate one's survival deliberately, at the command of one's free will.

Gage's brain selective damage did not alter any of his motor, language or logical abilities. However, he was now incapable of living according to the social rules he once learned, mostly taking decisions unfavourable to his survival. Picard (2000) observes that patients like Gage with emotional deficiency are 'similar to today's computers – particularly in how they malfunction'. Despite its clear implications, such fact has only recently captured the attention of researchers in the area of artificial intelligence.

All emotions may be relevant in some way as inspirations for improving particular skills of a robot or system. For instance, among social animals like humans, emotions such as embarrassment, empathy, guilt, disgust, compassion and gratitude are essential for living in accordance with the social rules of the group. Similarly, robots that incorporate models of such emotions in their artificial intelligence may also be more successful in applications that heavily depend on social interaction with humans, such as companion robots.

Each emotion certainly plays an essential and special role in an individual's behaviour and chances of survival. Nonetheless, it is interesting to observe that among all emotions, fear is the most ubiquitous in nature, influencing individuals' behaviour and decision-making in the great majority of animal species (LeDoux 1999). It is also notably the most extensively studied emotion in the neuroscience, psychology and cognitive computing fields.

1.1.2 Fear Learning

In this thesis, we are specifically interested in the simulation of one particular emotion: fear. More precisely, we are interested in brain mechanisms that provide humans and other animals with the ability to learn to fear environmental stimuli and situations that were previously emotionally neutral, as well as their ability to memorise and retrieve such acquired knowledge with exceptional rapidness.

Learning to detect, predict and quickly react to new threats in the environment is a pervasive skill in nature, as most known animals have been noted to manifest it in some way. Known as *fear learning*, this skill evolved as the brain's mechanism for automatic learning and memorization of potential threats to one's survival.

The success of this mechanism, evidenced by its ubiquitousness in nature, is partially a consequence of its implicit and involuntary activation. It is implicit because the feeling of fear commonly manifests itself even before the individual is aware of it, and it is involuntary because it happens regardless of the individual's will. These two characteristics make of the fear-learning mechanism a powerful and critical ally in the face of immediate danger. Unlike conscious/rational thinking, which is slow, the implicit and automatic responses triggered by fear provide the individual with a balance between the speed and gains of a decision (Damasio 1994; LeDoux 1999), while automatically increasing focus and attention (Fragopanagos and Taylor 2006). In addition, fear learning is also essential for environmental adaptation because the brain constantly associates the feeling of fear with newly experienced dangers. In unfamiliar environments, these fear associations allow the individual to learn to predict and quickly react to new threats.

Fear learning supports not only survival and environmental adaptation, but also social adaptation (i.e., one's ability to adjust its behaviour to the rules of one's own society) (Twining et al. 2017). The concept of society applies to many animal species, where individuals feel an instinctive need to be accepted by others of its kind. As belonging to a community can highly increase one's chances of survival, the brain of many animal species evolved to process social rejection as an aversive environmental stimulus. Consequently, the brain triggers fear learning when an individual observes disapproval from others towards its actions.

Studies on the cerebral interactions of the *amygdala* (area of the brain essential for the acquisition and expression of fear (LeDoux 2003)) indicate that emotions have a strong relation with cognitive processing, being able to change an individual's environmental perception and reaction strategy (LeDoux 1995). According to LeDoux (1995), the amygdala is capable of allowing the appraisal of danger to modulate complex information-processing functions of the *hippocampus*, including spatial behaviour, contextual processing, and memory storage and retrieval. This means that brain circuitries involved in fear expression are capable of influencing the outcome of areas of the brain responsible for logical processing.

By being real agents that inhabit the physical world and interact with human beings, autonomous robots are also susceptible to environmental threats and to managing social adaptation. There is a variety of areas in robotics which could take great advantage of emotion-based models of artificial intelligence, especially those based on the fear-learning mechanisms of the brain. Among those areas, we highlight *Human-Robot Interaction* (HRI), healthcare and autonomous vehicles. Section 1.2 explores the potential uses of emotionally intelligent robots for each of these areas of robotics.

1.2 Emotionally Intelligent Robots

This section presents robotic fields and applications that have the potential to take great advantage of a situation-aware fear-learning model of artificial intelligence, thus constituting solid motivations to pursuing the goals contemplated in this thesis. In addition to the fields and applications introduced in the next sections, we also highlight the robot soccer world cup as an exemplary scenario for the application of a situation-aware fear-learning model of artificial intelligence. This is, in fact, the scenario of the case study that we have performed to evaluate our model. This scenario and its relevance for the advancement of robotics are discussed in Chapter 6.

1.2.1 Human-Robot Interaction

Robot companions (Dautenhahn 1998; Ho et al. 2009; Vargas et al. 2011; Enz et al. 2011) are gaining more space in our society as social entities and have shown a great potential for applications in many areas (e.g., health and mental care (Kim, Gu and Heo 2016; Riek 2015)). However, a common issue with long-term robot companions is the rapid loss of interest from their users, who get frustrated and lose motivation over time as companions continue to perform predefined and repetitive behaviours (Ho et al. 2009). This poses a challenge to the broad development and practical use of robot companions.

From the HRI (Human-Robot Interaction) point of view, robots' social interaction becomes more believable and natural as they become more adaptable and responsive to environmental cues (Lazzeri et al. 2013; Dautenhahn 1998; Vargas et al. 2011). As humans, we expect others to be able to identify environmental factors that can represent unpleasantness or danger to themselves and act accordingly. Therefore, being able to properly express fear responses could highly increase the believability of a long-term robot companion (Lazzeri et al. 2013).

Additionally, fear learning also supports social adaptation, as discussed in Section 1.1.2. In natural organisms, fear may also be triggered when an individual perceives that its actions have been condemned by other individuals of its social environment. Similarly, simulated fear could also work as a motivation for the robot to seek social acceptance among humans. For instance, a robot's fear mechanisms could be activated whenever users express discontentment towards its actions (e.g. through verbal warning or facial expression), allowing the robot to learn to identify and inhibit behaviours that may be socially awkward, annoying or unacceptable. Over time, the robot's behaviour would be gradually adjusted to its user's custom preferences and social habits.

1.2.2 Healthcare

Robotics has been increasingly used in a variety of ways for both health and mental care applications (Riek 2015). The most widely known use of robotics in medicine is, perhaps, to increase the precision of surgical procedures or to conduct remote surgeries. In recent years, however, other areas of robotics such as HRI have also received growing attention in healthcare. Section 1.2.1 discussed how artificial mechanisms of fear learning can aid the field of HRI by improving robots' believability and social behaviour. In this section, we explore how fear learning can be indirectly used through HRI to improve social robots in healthcare areas such as rehabilitation therapies, treatment of *autistic spectrum disorder* (ASD) and elderly care.

Physical therapies usually involve constant intervention from therapists to perform repetitive limb movements. In many cases, the full attention of more than one therapist is necessary for one single patient. The development of rehabilitation robots emerged from the increasing need to support therapists with laborious and repetitive training. In addition, such robots also provide better means to asses the motor recovery of patients by measuring changes in their limb movements (Kim, Gu and Heo 2016). For instance, Garmsiri, Najafi and Saadat (2013) control rehabilitation robots using an emotion-based model that considers and adapts to patients' physical reactions rather than performing repetitive preprogrammed movements. HRI approaches are also valuable for rehabilitation therapies, as has been demonstrated by Matarić et al. (2009). They conducted a study involving an autonomous assistive mobile robot that provides social and cognitive support to stroke patients in rehabilitation. Their study concluded that improving HRI aspects leads to an increase in users' task performance in daily and rehabilitation activities.

Social robots have also been used for studying and treating ASD (Kozima, Nakagawa and Yasuda 2007; Pioggia et al. 2008). These robots help to encourage proactive interactive responses in children with ASD. According to Pioggia et al. (2008), studies with robotic dolls, mobile robots and humanoids acting as social mediators have provided important insights into the study and treatment of children with ADS. For instance, using a minimalistic robot capable to express attentional and emotional responses, Kozima, Nakagawa and Yasuda (2007) demonstrated that, against the commonly held opinion, autistic children do exhibit motivation to share mental states with others.

Finally, elderly care is a specially critical subject for robotics in health care. By 2050, the population of Europe aged 65 or older is expected to reach nearly 173 million people (Bemelmans et al. 2012). Such population accounted for only 101 million people in 1995. Consequently, the ratio of younger and older people will be severely unbalanced in comparison to the current status, requiring more caregivers and likely reducing the quality and support of health care systems. The need for robotic technologies capable to aid and alleviate the pressure on elderly care systems is evident, which led to a large body of new studies and developments in assistive robotics. While assistive robots that aid with mobility and other physical related needs are also necessary, an increasing need for socially assistive robotics has led to the fusion of HRI with elderly care. The goal of such robots is to develop close and effective interactions with the person while providing support to further needs, such as rehabilitation and mobility (Bemelmans et al. 2012).

1.2.3 Autonomous Vehicles

Highly dynamic and ever-changing environments require from individuals an exceptional ability to take rapid and flexible decisions while continuously adapting to the frequent variations of that environment. Therefore, robots dealing with highly dynamic environments are the ones in most need of human-like flexibility in decision making.

A very popular example of such robots nowadays is autonomous vehicles, especially autonomous cars. The environment of an autonomous car in a motorway is highly dynamic, involving not only the usual path-finding task of a robot but also traffic regulations, vehicle dynamics and other drivers' intentions. A human driver takes advantage of emotional reasoning to take all these factors into account when processing information and making decisions. Research has demonstrated that autonomous vehicles can also take advantage of emotional models that simulate a human-like driving behaviour for situation and risk assessment, as well as driver assistance (Kraus et al. 2009; Reichardt 2008). In addition to autonomous cars, other types of autonomous vehicles have been shown to also take advantage of cognitive models of emotion, such as unmanned aerial vehicles (UAVs) (Jafari, Shahri and Shouraki 2013) and aerospace launch vehicles (ALV's) (Mehrabian, Lucas and Roshanian 2006).

In addition to dynamic environments, uncertain environments also pose an exceptional challenge to many exploration and research robots such as autonomous or unmanned surface, underwater and aerial vehicles (Paula and Acosta 2015). A number of research areas depend on robots to explore and study inhospitable environments, where the human presence is impractical, if not impossible. A strong example of such scenario is planetary exploration. These robots require dynamic and highly adaptive decision-making mechanisms to increase their chances of successful mission completion. Therefore, such robots could greatly benefit from control mechanisms inspired by emotional models of human-like decision making (Ippolito, Pisanich and Young 2005).

1.3 Requisites of a Robust Situation-Aware Fear-Learning System

Fear learning has been a strong source of inspiration for developing more flexible and adaptive artificial intelligence (Morén and Balkenius 2001; Neal and Timmis 2003; Timmis, Neal and Thorniley 2009; Lucas, Shahmirzadi and Sheikholeslami 2004; Lotfi and Akbarzadeh-T. 2014b; Salichs and Malfaz 2012). The potential of artificial intelligence based on fear-learning models is demonstrated by its successful contribution to a variety of industrial, engineering and robotic applications, such as:

- Aerospace Launch Vehicle (ALV) control (Mehrabian, Lucas and Roshanian 2006);
- Washing machine control (Lucas, Milasi and Araabi 2006);
- Embedded systems (Jamali et al. 2010);
- Speed and flux control of induction motors (Markadeh et al. 2011; Daryabeigi, Abjadi and Arab Markadeh 2014);

- Online prediction of geomagnetic activity indices (Lotfi and Akbarzadeh-T. 2014a)
- Path tracking and collision problem in automated highway systems (Jafarzadeh et al. 2008);
- Patient-cooperative control of rehabilitation robots (Garmsiri, Najafi and Saadat 2013);
- Target tracking control of a mobile robot (Kim and Langari 2009);
- Motion control of omni-directional three-wheel robots (Sharbafi, Lucas and Daneshvar 2010);

Despite its advances, research on artificial fear-learning is still in its infancy and has several aspects with a margin for improvement, among which we can highlight *situation appraisal*.

In Rizzi et al. (2017), we define *situation appraisal* as one's ability to 'make emotional evaluations and associations over perceived situations, where *situations* are collections of contextual information linked to relevant entities. In other words, a situation captures the fluctuations in an individual's context during a particular period of time. Situation appraisal is, therefore, an individual's ability to attach emotional meanings to perceived situations and react accordingly.

To the best of our knowledge, artificial fear-learning models proposed to date do not substantially address situation appraisal, which is a significant part of the brain's fear-learning system, and essential for an organism to predict outcomes and adapt to threats and environmental changes (Harrison, Duggins and Friston 2006). To model and implement situation appraisal, a number of requirements must be taken into consideration, which can be divided into two groups: requirements of a situation-aware intelligence and requirements of an emotional intelligence.

1.3.1 Situation-Aware Intelligence

In the real world, people display emotional responses not only to individual environmental stimuli (e.g. pain, smells, noises, location, light levels, etc.), but also to context (i.e., the unified meaning of co-occurring stimuli) and to context's variation over time. In light of how fearful emotional reactions emerge in organisms, we argue that in order for a robot to have a detailed and extensive comprehension of its state of affairs, a few requirements must be addressed during the modelling phase of the robot controller. These are:

- The robot controller must take into consideration the current state of each particular stimulus sensed by the robot (either by means of sensors or direct user input), as well as how each particular state of each stimulus influences the robot's interaction with its environment.

- The robot controller must take into consideration the current combined state of all stimuli sensed by the robot, as well as how particular configurations of these stimuli influence the robot's interaction with its environment. This describes the robot's *context*.

- The robot controller must take into consideration the variation of the state of each particular stimulus sensed by the robot over time, as well as how the pattern wherein each particular stimulus varies over time influences the robot's interaction with its environment.

- The robot controller must take into consideration the variation of the combined state of all stimuli sensed by the robot over time, as well as how the pattern wherein particular configurations of these stimuli vary over time influence the robot's interaction with its environment. This describes the robot's *situation*.

1.3.2 Emotional Intelligence

The items listed in Section 1.3.1 address factors related to environmental perception from a neutral perspective only. In order for a robot to have an emotional reaction in response to the perceived environment, an emotional mechanism must be integrated with the contextual perceptions listed in Section 1.3.1. Because in this thesis we are particularly interested in the fear emotion, we will address only requirements for the integration of models that are inspired by the mechanisms of fear learning in the brain.

There are particular features which are essential and recurrently present in the mechanisms of fear learning in the brain. These are:

- *Neuroplasticity*, which regards to the capacity of neural circuitries in the brain to adapt to new environments and situations. This usually involves the ability of that neural circuitry to forget once learned information that is no longer valuable, allied with the ability to learn new information based on experiences from interacting with the environment.

- *Associative learning and memory*, which regards to the capacity of neural circuitries to create and store links between two or more distinct stimuli, based on coincidences in their patterns when they co-occur. Later, this association facilitates retrieving the memory of one stimulus when the individual experiences the presence of the other related stimulus.

- *Real-time learning and adaptation*, which regards to the ability to learn and adapt to new stimuli and threats while they occur in an automatic and rapid manner. This skill is particularly essential for the survival of animals in the wild, which need to quickly identify the presence of predators and decide a course of action (e.g. fleeing, fighting or freezing). In fact, research shows that brain circuitries involved in fear learning are usually activated before the activation of brain regions responsible for conscious reasoning (LeDoux 1999).

Ideally, a robust model for simulating situation-aware fear learning should meet all of the above-mentioned requirements, including those listed in Section 1.3.1. Nonetheless, as we understand, most models of fear learning applied to robotics address these requirements partially only, providing incomplete solutions based on an overly simplified environmental perception that leads to inconsistent emotional appraisal and responses. Chapter 2 discusses the main related work in the literature, analysing how and whether they meet the requirements discussed in this section.

1.4 Contributions of this Thesis

The central hypothesis of this thesis is as follows:

> It is possible to provide robots with online and domain-independent fear learning and memory capabilities at both stimulus and contextual abstraction levels through a robust mechanism for situation awareness that considers multi-stimulus temporal relationships. Such learning mechanism shall allow robots to perceive complex stimuli patterns by observing series of events in their environment, learn with experience through autonomous environmental exploration and adapt at execution time to environmental changes and threats.

In other words, the work presented in this thesis aims at allowing robots to learn complex temporal patterns of sensed environmental stimuli and create a representation of these patterns. This memory is later associated with a negative or positive 'emotion', analogous to fear and confidence. The learning and memorization processes are domain independent, so any robotic task that requires adaptive skills can take advantage of such model.

There is a number of fear learning architectures and models proposed to date that successfully address the requirements mentioned in Section 1.3.2. However, despite meeting essential requirements for implementing emotional intelligence, none of these models managed to simultaneously meet all the requirements listed in Section 1.3.1, which are crucial for generating situation-aware intelligence. The lack of models of fear learning in the literature capable to harmoniously meet the requirements for both situation-aware and emotional intelligence raises the first of our research questions, which is:

1. Can a cognitive computational model be designed so to fully meet the requirements of a robust situation-aware fear-learning model of artificial intelligence?

As we will later discuss in Chapter 3, Chapter 4 and Chapter 5, fear learning involves diverse brain systems performing considerably distinct tasks at different levels of abstraction by means of different mechanisms. Nevertheless, most models inspired by the brain's mechanisms of fear learning rely on one single technique to implement all these tasks and abstraction levels. This is, in fact, one of the reasons why many of these models provide overly simplified perceptions of the environment, both from an emotional and a contextual perspective. This fact leads us to our second research question, which is:

2. Can a hybrid cognitive computational model, depending on the contribution of different approaches and techniques, meet the requirements of a robust situation-aware fear-learning model of artificial intelligence?

Finally, a number of studies proposing fear-learning models provide an insufficient assessment of their models, especially in regards to their practical usage in real-world robotics applications. Other studies, in turn, openly declare their models as means to better study the underlying brain mechanisms involved in fear learning. These works usually focus on providing a close-to-real emulation of brain functions without addressing the practical usage of the model for artificial intelligence. The lack of fear learning models dedicated and thoroughly evaluated with respect to real-world robotics applications raises our third and final research question, which is:

3. Can a robust situation-aware fear-learning model of artificial intelligence be effective in real-world robotics applications?

This thesis proposes a novel hybrid cognitive computational model named *Situation-Aware FEar Learning* (SAFEL) (Rizzi et al. 2017; Rizzi, Johnson and Vargas 2016, 2017, 2018; Rizzi Raymundo, Johnson and Vargas 2015), which is inspired by the main brain regions involved in fear learning. SAFEL combines machine learning algorithms and concepts of situation-aware expert systems with well-known neuroscience findings on the brain's fear-learning mechanisms. SAFEL's ultimate goal is to provide autonomous robots with the ability to predict undesirable or threatening situations based on their past experiences and use this information for adapting to environmental changes and threats. SAFEL is designed to be domain independent and meet all the requirements discussed in Section 1.3. We evaluate the varied aspects of SAFEL's performance in a robot soccer scenario and demonstrate that SAFEL is capable of predicting undesirable outcomes based on complex contextual and temporal information and improve robots' adaptive behaviour and flexible decision making at execution time.

The main contributions of this work as compared to the state of the art are:

1. Integration of a fear learning model with the concept of temporal context. SAFEL performs threat predictions based on complex temporal and contextual information. Existing fear memory models either focus on the contextual or the temporal aspect, overlooking the need for both skills for an artificially intelligent agent to properly react to real-world threatening situations.

2. SAFEL is focused on real-world applications for artificial and autonomous intelligence in robotics. Many existing fear-learning models that are inspired by the real mechanisms of the brain focus on providing a close-to-real emulation of brain functions without addressing the practical usage of the model for artificial intelligence.

3. The successful integration of a symbolic rule-based platform for situation management with machine learning algorithms for memorizing and predicting threats based on complex temporal context.

1.5 Thesis Organisation

This thesis is organised as follows:

- *Chapter 2. Emotional and Adaptive Robots – A Prospectus:* This chapter discusses works in the literature related with the proposal of this thesis, particularly in the areas of affective computing, cognitive computational models and emotion simulation. It concludes by presenting an introductory overview of SAFEL's architecture.

- *Chapter 3. Amygdala Module:* This chapter presents the first module of SAFEL: the Amygdala Module (AM). This module is responsible for the emotional appraisal of SAFEL. Here, the biological background that inspired the model of the AM is discussed, as well as the underlying technology used in its implementation, the model of the AM and preliminary experiments.

- *Chapter 4. Hippocampus Module:* This chapter presents the Hippocampus Module (HM) of SAFEL, which is responsible for understanding and managing the contextual and temporal aspects of the robot's state of affairs. This chapter explores the biological background that inspired the design of this module. It also presents relevant conceptualizations and technologies used in its implementation, followed by the actual model of the HM.

- *Chapter 5. Working Memory Module:* This chapter presents the Working Memory Module (WMM) of SAFEL, which is responsible for associating the emotional memory generated in the AM with the contextual memory generated in the HM. Similarly to the other modules, this chapter briefly introduces the biological inspiration for the WMM, followed by its technological basis. It then presents the model of the WMM and preliminary experiments evaluating the HM and the WMM together.

- *Chapter 6. Case Study: Robot Soccer:* This chapter explores the application of SAFEL in the robot soccer scenario by means of a thorough case study. The analysis performed in this case study evaluates SAFEL in relation to three distinct perspectives: the predictive performance of SAFEL, improvements in the robot's playing performance and how learning evolves at runtime inside SAFEL's modules.

- *Chapter 7. Conclusion:* This chapter summarizes the work presented in this thesis, revisiting the hypothesis, contributions and research formulated in Section 1.4. Ultimately, our final considerations are expressed, followed by a perspective for future research.

Chapter 2

Emotional and Adaptive Robots – A Prospectus

The problem of adaptive robotics as addressed in this thesis concerns two domains of artificial intelligence and robotics: adaptive behaviour and emotion simulation. A number of methods can be used to approach each of these areas. In this chapter, we briefly introduce the varied approaches to simulating emotional and adaptive behaviour for robotics, as well as the main related works in the literature.

We start by introducing the field of affective computing in Section 2.1. Section 2.2 narrows the scope of the discussion to literature that is more closely related to the main goals of this thesis. Finally, Section 2.3 overviews the proposed Situation-Aware FEar Learning (SAFEL) model, outlining each of its modules along with their function in the overall architecture and their methodological basis.

2.1 Affective Computing

It is known from common sense that high levels of emotion may impair the ability to make rational decisions. However, as discussed in Chapter 1, neuroscience findings show that the absence of emotions may be even more harmful to rationality (Damasio 1994; LeDoux 1999). In the light of such discoveries, computer scientists started to consider the use of emotional models as a means to improve the intelligent and adaptive behaviour of artificial systems, leading to the creation of a new area of computer science: the *affective computing*.

Affective computing is one of the most recent branches of computer science, which originated from Picard's research in 1995 (Picard 1995, 2000). According to Picard (2000), affective computing tackles three aspects of artificial intelligence: (1) the ability of machines to recognise and express emotions, (2) the ability of machines to respond intelligently to human emotion, and (3) the capability of machines to regulate and utilize emotions in order to behave more intelligently and effectively.

The first and second branches of affective computing are concerned with the external aspects of emotions and develop techniques that allow artificial systems to recognize, evaluate, interpret and/or express human emotions, having as main target Human-Robot Interaction (HRI) applications. The third branch of affective computing is concerned with the internal aspects of emotions such as learning, perception and attention (Arbib and Fellous 2004). This is the branch of most interest to our work, though we may indirectly address aspects of the other branches as well. It studies techniques for simulating emotional behaviours in computer systems, inspired by the emotional model of humans and other animals. It derives from the idea that emotions give humans unique abilities, which make our decisions not only intelligent but also flexible, fast and efficient. The question in this branch of affective computing is whether artificial systems could also benefit from the positive aspects of emotions.

Next, we analyse how researchers in the area of affective computing describe and justify the need for emotional models in robotics. We also approach the main controversies and questions inevitably raised by the prospect of 'giving emotions to robots'. We conclude this section by exploring the main approaches for simulating emotions and emotion-derived behaviours.

2.1.1 The Role of Emotions in Robotics

We have discussed in Chapter 1 some applications for emotional robots that motivated and inspired our work. Here, we approach a similar discussion from a more structured perspective, aiming to evaluate the 'why and how' of applying emotional models in particular robotics applications.

Salichs and Malfaz (2012) consolidates the opinion of several researchers on this subject and delivers a democratic point of view on the benefits and roles that emotional models play in robotics. In summary, researchers' opinions generally converge to the conclusion that robots need emotions for the same reasons why humans and animals need and have emotions: because emotions help them with confronting their environment. Since robots share with us the same environment, it is reasonable to assume that they may benefit from emotions in the same way we benefit when dealing with our environment. Similar to humans and animals, emotions can be used in robotics for ends such as adapting to limitations, managing social behaviour and handling interpersonal communication. The ultimate goal is, therefore, to provide robots with improved autonomy and better social skills.

Breazeal and Brooks (2005) agree with this point of view by suggesting that the main reasons for developing emotionally capable robots converge to two design issues in robotics: (1) robust operation and behaviour in the human environment and (2) effective interaction and cooperation with humans. According to Breazeal and Brooks (2005), the list of advantages for giving robots emotional capacities include:

- Expressing intelligent behaviour in complex and unpredictable environments;
- Sensing and recognizing affect and emotion in others;
- Expressing affect and internal states in a manner that is familiar to humans; and
- Adapting to humans' social rules so to exhibit coherent and socially acceptable responses.

Breazeal and Brooks (2005) classify the applications of emotional robots into four paradigms:

- Robot as a tool: where the robot is used by humans as a device for performing a given task. As mentioned in Section 1.2, some robotics applications deal with dangerous and/or inhospitable environments, such as planetary and undersea exploration. In such scenarios, similar to an animal, the robot must apply limited resources to tackle multiple concerns while dealing with an uncertain environment and potentially dangerous situations. Breazeal and Brooks (2005) suggest that balancing emotion-inspired mechanisms such as interest and fear, for instance, could help the robot to maintain a focused state for performing its task while minding safety measures and surrounding dangers.
- Robot as a cyborg extension: where the robot is physically attached to the human body as an extension or replacement of a body part. Breazeal and Brooks (2005) argue that emotions play an essential role in the mind-body connection. Therefore, to effectively be part of or extend a human body, a robot must be able to recognize, adapt and match its features with the rest of the person's body according to his/her emotional state. For instance, when the person is in a calmer mood or situation, the robotic extension could enter into energy-conservation mode, as the need for power is mild at that moment.

- Robot as an avatar: where the person uses the robot to remotely communicate with others. Breazeal and Brooks (2005) argue that, in this scenario, the level of cognitive control and physical coordination required from the user to operate the robot in simple tasks (e.g., locomotion, object manipulation and facial expression) is overwhelming. Therefore, the robot needs to be equipped with mechanisms that allow it to understand high-level instructions from the user and autonomously perform these tasks. To do so, this mechanism must be able to effectively recognize the emotional and linguistic intent of the user.

- Robot as a partner: where the person interacts with the robot in a collaborative or social manner. This scenario depicts the classical HRI problem, in which a robot that interacts with people needs to display emotional and social intelligence so to respond coherently to peoples' expectation.

Examples of applications for models of emotions involving humans' interaction with robots (either socially or as a tool) is plentiful in the affective computing literature. Nonetheless, models of emotions can also be useful in applications that do not involve human interaction. For instance, Steunebrink, Dastani and Meyer (2006) discuss an application for emotion-inspired models involving the interaction and collaboration between artificial agents only. Steunebrink, Dastani and Meyer (2006) suggest creating a multi-agent model inspired by human emotions for solving two recurring problems of multi-agent systems: non-deterministic decision-making and lack of flexibility in agents' cooperation and coordination.

Multi-agent systems are composed of individual agents that are capable of autonomously deciding their own actions within the group in order to achieve a common goal. These agents are commonly modelled under standard statistical or rule-based techniques for decision-making. According to Steunebrink, Dastani and Meyer (2006), in many practical applications, such techniques give the agents multiple action possibilities with near-equal priority, leading the agent to behave nondeterministically.

To reduce the agents' nondeterministic behaviour and excessive deliberation, Steunebrink, Dastani and Meyer (2006) propose using the model of Ortony, Clore and Collins (1990). For instance, suppose a particular agent that creates a plan 'hoping' to achieve a particular goal. However, if the plan execution starts to fail, that agent will experience 'fear', which will increase over time as the plan continues to fail. Eventually, the 'fear' emotion will be more intense than the 'hope', leading the agent to abandon the initial plan and attempt another strategy.

Another issue discussed by Steunebrink, Dastani and Meyer (2006) involves the need for multiple agents to cooperate and coordinate their actions in order to achieve the system's global goal. The cooperation and coordination of agents are specified by interaction protocols, which usually impose restrictive constraints on the agents' behaviour and limit their autonomy. On the other hand, the absence of these protocols would lead to the nondeterminism issue because the agents would be left with several possibilities of interaction with each other. Steunebrink, Dastani and Meyer (2006) suggest solving this problem with a protocol that is based on the human interaction model, in which a person constructs a mental model of the affective state of other people for predicting their probable reaction. In such protocols, individual agents would be capable of anticipating other agents' reaction by becoming aware of their current emotional state. Suppose, for example, that a particular agent is having problems and is about to abandon its current task, which could compromise the system's global goal. If another agent is able to infer the emotional state of that agent, then it may be able to predict and avoid the interruption of the concerning task.

2.1.2 Debates and Deliberations

As discussed thus far, the benefits of emotions for natural life are numerous and self-evident. However, this is not a sufficient argument on its own to justify the inclusion of emotion-like mechanisms in robotics architectures. Cañamero (2005) questions whether the inclusion of emotional elements in robots' artificial intelligence makes them more valuable per se. In fact, if the inclusion of such emotional elements does not improve a robot's performance by any means, then what is the purpose of adding an extra resource-consuming processing step to the robot's architecture? Therefore, researchers must be capable to accurately demonstrate that the inclusion of such models of emotion does, in fact, improve the task performance or interaction capabilities of autonomous robots. Cañamero (2005) suggests, as a first obvious option, comparing the results of control experiments in which the robot performs a given task with and without the aid of emotional mechanisms.

We share the opinion of Cañamero (2005) and believe that if models of emotions are to be included in robotics architectures, they must first and foremost fulfil the requirement of improving the robots' autonomy and believability competencies. On this account, we present a number of preliminary experiments along this thesis which evaluate our emotional model and compare the outcomes of the robot with and without its influence. Ultimately, we provide the reader with a comprehensive case study in Chapter 6, which evaluates the robot's performance in a practical and highly dynamic scenario, subsequently demonstrating that the robot's task performance is substantially improved when SAFEL is incorporated into its architecture.

Finally, despite the extensive use of the word 'emotion' within the affective computing literature, it is important to keep in mind that such emotion-inspired models do not aim at fully and realistically mimicking the actual mechanisms of emotions in natural organisms. As Breazeal and Brooks (2005) observe, these models simulate emotions from a functional perspective only, aiming to give robots the adaptive benefits that emotions provide to natural beings and help them to more successfully accomplish their tasks.

On the other hand, one may raise ethical questions on whether robots should or could have human emotions (Lin, Abney and Bekey 2011; Coeckelbergh 2012; Bringsjord and Clark 2012). The prospect of 'giving emotions to machines' raises delicate ethical concerns and polemic discussions. In this direction, one of the most discussed matters is the deceptive behaviour of emotive robots, under the claim that they 'intend to deceive' by expressing non-authentic emotions and 'pretending' to be a kind of entity that they are not (Coeckelbergh 2012; Bringsjord and Clark 2012). We contend that such questions are beside the purposes of our research at this point, as artificial emotions serve solely as a means to improve robots autonomy and adaptation in our work. Yet, we hold the idea that such computational mechanisms can be justifiably acknowledged as the artificial counterparts of the real emotional phenomena. We share the point of view of Breazeal and Brooks (2005), who argue that artificial emotions 'are not "fake" because they serve a pragmatic purpose for the robot that mirrors their natural analogs in living creature'.

2.1.3 Approaches to Simulating Emotion

The work of (Ortony, Clore and Collins 1990), widely known as the *OCC theory*, is among the most cited models of emotion in the literature, being a source of inspiration for the development of several other models of emotions. According to Ortony, Clore and Collins (1990), an individuals' emotions result from its cognitive capacity of appraising an event or situation in regards to its own goals, standards and actions. Therefore, the emotional appraisal is unique to each individual because it depends on many elements that are particular to each individual. This explains why the same event may lead to different emotional reactions from different individuals, or even from the same individual at different times.

The idea of establishing affective appraisal in terms of the agent's goals is particularly well endorsed among the affective-computing community and is commonly implemented as the agent's 'drives' and 'motivations'. According to Ahn and Picard (2005), motivation plays a fundamental role in the processes of learning and decision making in humans and other animals. They proposed an affective-cognitive learning framework that models the motivation of artificial agents by means of rewards, which may be internal (cognition and emotion) or external (interaction with the external world). Once the agents are motivated, they are able to autonomously learn and make decisions. For example, happiness when finishing a given task is an example of internal reward. On the other hand, an example of external reward would be positive reactions of users when interacting with the system. If the user demonstrates happiness or satisfaction in response to the agent's action, then it is recognized as a reward by the agent, which learns that this action is correct in the current context.

While identifying the circumstances and components of the environment that influence the affective state of an individual is essential for simulating emotions, it is also necessary to adopt a methodology for modelling emotional appraisal, i.e., how the agent affectively assess the situation. According to Salichs and Malfaz (2012), two main approaches define the methodological foundation for modelling emotional appraisal: the *affective space model* and the *discrete model*. Both methodologies have as main concern defining how agents' situations and emotions are interrelated.

The affective space model associates situations and emotions by means of a set of intermediate variables working as dimensions of a Cartesian plane, namely, the affective space. Each emotion is associated with a different zone of that space. For instance, Hollinger et al. (2006) define emotions based on three intensity variables: pleasure, arousal and dominance. These variables represent the three axes of the affective space and vary with different intensities depending on the situation that the agent observes. The resulting point of these three coordinates in the affective space defines the agent's current emotion.

The approach of Breazeal and Brooks (2005) to social robots is another example of affective space model. In their approach, the robot's emotional state is defined according to a three-dimensional affective space, whose axes correspond to arousal (high or low), valence (good or bad) and stance (advance or withdraw). The interaction of these affective variables generates nine artificial emotions at varying intensities and is dependent on the robot's drives, goals and accomplishments.

By contrast, the discrete model is based on the discrete categorization of emotions for defining the agent's behaviour in a particular situation. This approach may also consider dimensions of emotional intensity, but only within the emotional categories and with descriptive purposes. A widely cited example of a discrete model is the approach proposed by Cañamero (1997). She models a set of six independent emotions (fear, anger, happiness, sadness, boredom and interest), each of which is characterized by an incentive stimulus, an intensity, hormonal discharges and the activation of physiological responses. The intensity of these emotions is influenced by factors such as external events, stimuli patterns and particular combinations of physiological states.

Salichs and Malfaz (2012) also propose a discrete-model–based approach for generating synthetic emotions, aiming at improving the decision-making process of autonomous virtual agents. They argue that, generally, 'appraisal occurs in a holistic fashion, and it is based on theme evaluation rather than on analytical processing using evaluation dimensions'. According to Salichs and Malfaz (2012), the same situation may be appraised in a different manner for each particular emotion, which could have distinct repercussions in the individual's decision-making. Therefore, the manner with which a particular situation is affectively appraised must be independently studied for each individual emotion.

Additionally, Salichs and Malfaz (2012) believe that virtual agents should be capable of learning by themselves to select the correct action in a particular situation based on their drives, motivations and emotions. Different people may respond differently to the same situations. Analogously, artificial agents should be able to 'discover' their own emotions in response to particular situations, as well as reactive tendencies in the face of particular emotions.

The emotional model proposed in this thesis is largely in agreement with the principles and arguments expressed by Salichs and Malfaz (2012). SAFEL's embodies a discrete-model approach for modelling and simulating the fear emotion in autonomous robots. Fear responses induced by SAFEL are integrally dependent on the outcomes of the robot's interaction with its environment and are shaped by the robot's own perception of its situation. In comparison to the above-mentioned approaches, especially those categorized as discrete models, SAFEL stands out by providing the robot with much richer and complex perception of its environment. This contributes to better precision when detecting fear triggers in the environment and, consequently, generates more coherent and believable fearful responses in the robot.

2.1.4 Unconventional Emotional Behaviours in Robotics

The wide range of different emotions studied by neuroscience and psychology has, over the years, provided a large body of evidence and theories that helped to formulate computational models for a number of distinct emotions. In previous sections, we introduced models for modelling conventional emotions and emotional behaviours such as happiness, sadness, fear and anger. In this section, we explore the computational modelling and application of more peculiar emotional behaviours, such as comfort, deception and guilty. In Section 7.3 we revisit the idea of potentially simulating these behaviours as an expression of fear by means of SAFEL.

Comfort Zones

The notion of comfort is well known between humans and other animals. It determines how safe or satisfied we feel in relation to our environment. If we are in a safe and familiar place, surrounded by friends, we feel comfortable and act confidently. On the other hand, when we feel under risk or without support in an unknown and hostile place, we feel uncomfortable and become resistant to explore the surroundings. The level of comfort, therefore, is a powerful danger-alert mechanism.

Likhachev and Arkin (2000) propose a comfort function for controlling robotic systems, which is based on a psychological research performed on infants. According to Likhachev and Arkin (2000), objects of attachment have a strong participation in determining an individual's level of comfort. For example, a mother may be an example of an object of attachment for a child, who associates the mother's presence with safety. When a person is close to an object of attachment, his or her comfort level is higher than it would normally be in the same environment, under the same circumstances.

The definition of a robot's object of attachment may be either hardwired or learned through environmental exploration. The attachment force between the robot and an object of attachment would depend on how much that object fulfils the robot's needs. In the work of Likhachev and Arkin (2000), the robot's 'desire' to return to an object of attachment increases when it's overall comfort level decreases. This 'desire' force is a vector directed toward the object of attachment, which depends on (1) the attachment bond to that object, (2) the robot's current comfort level and (3) the distance between the robot and the object. As a consequence, the attachment behaviour regulates the robot's exploration of the world, i.e., as the environment becomes more familiar and safe, the robot becomes more confident in getting far from the attachment object, therefore increasing its exploration range.

The advantage of this approach for robot exploration is clear: it prevents the robot from entering unknown areas without proper caution. As the robot gets familiar with that new area, it starts to act more confidently and stops spending resources with rigorous environmental evaluation. Thus, the use of comfort zones may help to reduce the chances of physical damage to the robot.

However, despite the comprehensive biological background provided in their work, Likhachev and Arkin (2000) do not explore its full potential in their implementation. For example, the authors extensively discuss the influence of endogenous (internal factors, e.g., hunger, body temperature, pain, etc.) and exogenous (external factors, e.g., the individual's degree of familiarity and past experiences in the current environment) variables in an individual's level of comfort. However, in their mathematical equations, endogenous variables are ignored, and only objects of attachment are taken into account from the exogenous variables, which, nonetheless, the robot is allowed to have only one.

Vazquez and Malcolm (2004) also used the concept of comfort zone in their work as a strategy for avoiding collisions and loss of connection in multi–autonomous-robot exploration tasks. They determine two important comfort zones, one for safety (where there is no risk of collision with other robots) and one for maintaining connectivity with nearby robots. To be considered safe and interconnected with another robot, a robot should lie inside its own safe region and inside another robot's connectivity region.

However, they do not explore the idea of comfort zones with the same depth as Likhachev and Arkin (2000). For example, Likhachev and Arkin (2000) provide an extensive discussion on the psychological factors of comfort zones and how they can be mapped to computer systems, whereas Vazquez and Malcolm (2004) limit comfort zones to threshold areas where the robot should lie for keeping the team operational. This is likely a consequence of the diverging goals of these two works: Likhachev and Arkin (2000) focus mainly on the idea of comfort zones while Vazquez and Malcolm (2004) are concerned with solving the team-collaboration problem, for which they use other techniques in addition to comfort zones.

Deception

Despite not being an emotion, the act of deceiving is usually triggered by emotions, such as fear, trust, shame, guilt, envy and even love or surprise. In our society, it is usual to think of deception as a misconduct, which may lead others to judge an individual's character. In nature, however, such behaviour is common and essential for the survival of certain animal species, mainly for small animals that cannot rely on the use of force to defend themselves. Analogously, there is a number of computing and robotics applications that can benefit from simulating deceptive behaviours, such as military technology, sportive training, games, environment simulation and healthcare (Nijholt et al. 2012).

Davis and Arkin (2012) propose a mathematical model for robotic deception based on the mobbing behaviour of a species of birds called Arabian Babbler. These birds act in groups to defend themselves against predators' attacks. A sentinel bird is responsible for observing the surroundings and alerting the others in case of danger, which in turn mob the predator if it persists with the attack. The mobbing may evolve to a harassment that, if sufficiently strong, can lead the predator to abandon its attack.

The computational model of Davis and Arkin (2012) consists in a mathematical equation that evaluates an individual's risk of participating in mobbing considering that individual's fitness, the predator's fitness, the price of bluffing and a *relatedness coefficient*. The relatedness coefficient expresses the level of prey-predator cooperation. In other words, the prey should avoid being chased and the predator should avoid wasting energy in hard hunts. As the relatedness coefficient increases, the agents are more likely to cooperate and, consequently, the chance of mobbing increases.

Another example of robotic deception based on animals' behaviour is the work of Shim and Arkin (2012). They simulate in a mobile robot the deceptive behaviour of tree squirrels when protecting stored food. Tree squirrels have the habit of periodically checking their food locations. However, when there are competitors nearby, they visit several empty locations in an attempt to confuse the competitors. Shim and Arkin (2012) have created a mathematical function for generating artificial deceptive-behaviour that calculates the probability of a robotic squirrel to visit a given food location (empty or not). This calculation takes into consideration the amount of food in each location and the presence of competitors around.

Both above-mentioned works, Davis and Arkin (2012) and Shim and Arkin (2012), are inspired by the deceptive behaviour of wild animals in a survival situation. The work of Wagner and Arkin (2011), by contrast, proposes a different approach for generating deceptive behaviour. They make use of game theory and interdependence theory to develop an algorithm that determines whether deception is warranted in a particular social situation.

They assume that the deceiver must have specific knowledge about the individual that will be deceived (called by the authors as *the mark*). For instance, camouflaging would not deceive a robot that relies on infrared vision. Therefore, the robot attempting to camouflage should be aware of the vision capacities of the mark in order to successfully deceive it.

In interdependence and game theories, social interactions are represented by an outcome matrix that stores information about the individuals interacting (in this case, the deceiver and the mark), as well as the gain/loss for each individual in relation to their potential combination of actions. The deceiver's task is to use its knowledge of the true outcome matrix (i.e., the outcome when there is no deception in the communication) to convince the mark that a particular decision will benefit it when it actually benefits the deceiver.

Guilt

Arkin, Ulam and Wagner (2012) address a delicate subject in affective computing: ethics. The ethical aspects of 'giving emotions to machines' have raised several discussions in the area of affective computing, and it is even more polemical when regarding military issues. Arkin, Ulam and Wagner (2012) propose a model of ethical behaviour for robotics in the military context, where robotic soldiers are induced to follow international agreements of war conduct.

In their approach, the ethical part of the robot's controller consists of 3 components. The first component, called as the *ethical governor*, is responsible for evaluating the ethical appropriateness of any lethal action intended by the robot, intervening when necessary to prevent unethical actions. The function that evaluates ethics is based on two international agreements, known as *Laws of War* (LOW) and *Rules of Engagement* (ROE) (Arkin, Ulam and Wagner 2012).

The second component, called as the *ethical adaptor*, allows the system to express the guilt emotion by adapting its behaviour to the consequence of its actions. The adaptor recognizes the need of guilt expression by means of the governor's judgement, which has autonomy to determine whether the actual collateral damage of the robot's action significantly exceeds the estimated damage. The level of guilt is calculated according to the number and importance of causalities caused by the robot. Guilt is expressed by deactivating the robot's weapons; if the level of guilt exceeds a specified threshold, all lethal weapons are deactivated for that robot until the end of the mission.

Finally, the third component of the controller allows the robot to strategically deceive its opponents if needed. The decision-making process for the deceptive action is determined by the previously discussed framework developed by Wagner and Arkin (2011). Even when developing a system with the sole purpose of deceiving, ethical issues should be considered. In the work of Arkin, Ulam and

Wagner (2012), the robot's decision to deceive takes into account that a false communication implies in a conflict with the mark. Therefore, it should deceive only when there is no other option. In other words, a true communication should be preferred if the deceiver can benefit from it, even when it aids the mark.

2.2 Fear, Context and Adaptation

Fear is one of the most discussed emotions in psychology and neuroscience when regarding survival and adaptation (LeDoux 1999; Damasio 1994). It allows us to take advantageous decisions to our well-being and interests before we even become conscious of it and faster than our 'rational brain' could process. The clear impact of fear learning in the expression of adaptive and intelligent behaviour captured the attention of many researchers in the field of artificial intelligence, who saw in the brain's mechanisms of fear an opportunity for leveraging the adaptive behaviour of artificial agents.

In this section, we revisit previously proposed computational models of fear learning and expression with main application to adaptive autonomous robotics. We also compare these models with the requirements specified in Section 1.3, especially those regarding situation awareness.

2.2.1 Artificial Homeostatic Systems

Artificial homeostatic systems (AHS) are adaptive systems inspired by the biological mechanisms of homeostasis, which is related to the property of an organism to regulate and maintain stable its internal state (Vargas et al. 2005). A homeostatic system is, therefore, the integration of all other systems in an organism that primarily supports its internal balance. In biological organisms, these are believed to be the immune, neural and endocrine systems (Vargas et al. 2005). Inspired by these biological systems, researchers in the area of artificial intelligence started to investigate and propose models of *artificial endocrine system* (AES) (Neal and Timmis 2003; Vargas et al. 2005; Timmis, Neal and Thorniley 2009; Thenius, Zahadat and Schmickl 2013) and *artificial immune system* (AIS) (Read, Andrews and Timmis 2012) to work in integration with and modulate *artificial neural networks* (ANNs).

An exemplary work in the area of AHS is the model proposed by Neal and Timmis (2003), which later served as a foundation for other AHS models in the literature (Vargas et al. 2005; Timmis, Neal and Thorniley 2009; Thenius, Zahadat

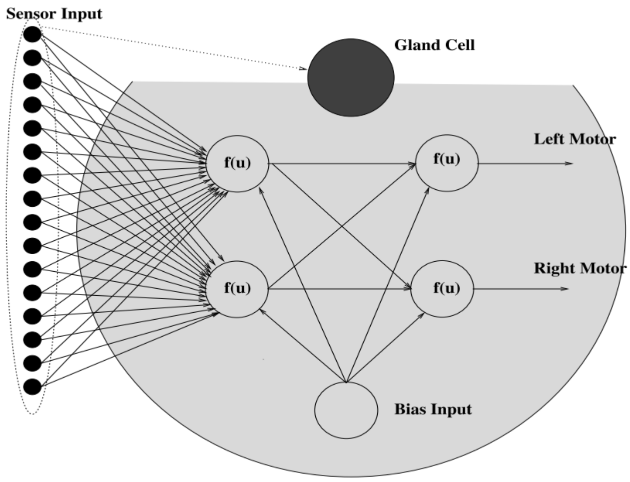


Figure 2.1: Integration of an artificial endocrine gland with the artificial neural network controlling the robot in the experiment of Neal and Timmis (2003). The grey area indicates the neurons influenced by the hormonal gland.

and Schmickl 2013). Neal and Timmis (2003) present an AES mechanism for generating emotive behaviour in artificial agents. Their AES mechanism consists in the modelling of an artificial endocrine gland that secretes 'hormones' in response to external stimuli and is responsible for varying the system's hormonal concentration over time. Hormone secretion of a particular gland g is given by r_g as seen in Eq. 2.1, where α_g is the rate of hormonal release for gland g. The hormonal concentration $c(t)_g$ induced by gland g at time t decays over time according to the formula of Eq. 2.2, where β is a decay constant.

$$r_g = \alpha_g \sum_{i=0}^{nx} x_i \tag{2.1}$$

$$c(t+1)_g = (c(t)_g \times \beta) + r_g \tag{2.2}$$

Neal and Timmis (2003) evaluate their model on a wheeled robot with 16 sonar distance-sensors. The proposed AES mechanism was integrated with the ANN controlling the robot's movements, as seen in Fig. 2.1. The goal is to use the system's hormonal concentration to modulate the output of the ANN and, consequently, the robot's behaviour. In other words, through the proposed AES mechanism, the system is expected to change its default behaviour depending on its perception of the environment. Neal and Timmis (2003) demonstrated in their experiment that the addition of the AES mechanism generated in the robot a behaviour analogous to the fear reflex, which was autonomously learned during environmental exploration. As the frequency of obstacle detection increased over time, the aggressiveness of the robot's movements in order to flee also increased as if it had 'fear' of colliding with obstacles.

The work of Neal and Timmis (2003) was later extended by Timmis, Neal and Thorniley (2009) in order to generate adaptive behaviour during the robot's environmental exploration. Timmis, Neal and Thorniley (2009) argue that even though the work of Neal and Timmis (2003) allows robots to change their behaviour according to environmental input, the hormonal concentrations are pre-configured rather than learned. Timmis, Neal and Thorniley (2009) propose an improved AES named as the *adaptive artificial neural-endocrine* (AANE) system. The AANE system includes a feedback mechanism that allows the autonomous modulation of the system's hormonal concentration during the robot's operational cycle.

Figure 2.2 depicts the system architecture that Timmis, Neal and Thorniley (2009) have implemented for their experiment. In this architecture, they use two neural networks, each associated with a hormonal-gland cell. Each ANN determines a particular behaviour of the robot, which in this case are 'wander' and 'avoid obstacle'. The gland cells regulate the intensity in which these behaviours dominate, so they can be seen as the robot's 'desire' to manifest a particular behaviour.

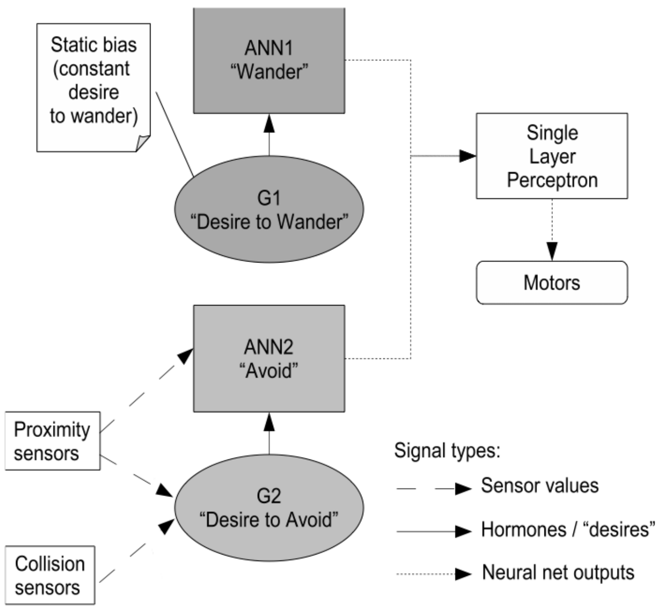


Figure 2.2: Overall system architecture for controlling the robot in the experiment of Timmis, Neal and Thorniley (2009).

The final output determines the speeds of the robot's wheels and is computed by a third neural network (the 'single layer perceptron' component in Figure 2.2), which combines the outputs of the other two ANNs. In addition to the components of Figure 2.2, Timmis, Neal and Thorniley (2009) add a *negative-feedback gland* in the second part of their experiment, which regulates the update of the ANN's weights in order to prevent their exponential increase/decrease.

The experimental results of Timmis, Neal and Thorniley (2009) demonstrated that the robot was capable to express adaptive behaviour over prolonged periods of time. In addition to the association between collision and proximity, similar to the association discussed in the experiment of Neal and Timmis (2003), the proximity signal was constantly adapting at runtime and self-reinforcing, so that no further collisions were necessary after some time for the collision-proximity association to be maintained.

Other works that improve or expand the functionalities of the model presented by Neal and Timmis (2003) include the models proposed by Vargas et al. (2005) and by Thenius, Zahadat and Schmickl (2013). Vargas et al. (2005) propose a homeostatic system that includes two other modules in addition to the artificial hormonal gland, which are the *hormone level repository* (HL) and the *hormone production controller* (HPC).

The HL records the level of hormone in the agent while the HPC controls hormone production according to the internal and external factors affecting the agent. The gland is responsible for the actual production and secretion of hormones, which is modulated by excitatory signals from the HPC. When excited by the HPC, the gland starts to produce and release hormones, which consequently increases the hormone level of the agent. The hormonal level, in turn, influences the output of the ANN and, as a consequence, the actions of the agent in the environment.

HPC excitatory signals depend on the internal state of the agent, which in turn depends on two factors: the agent's hormonal level and external state (i.e., how close the agent is to accomplish its goals). The HPC ceases excitatory signalling when these two factors rise above given thresholds, which in turn interrupts the hormonal production and secretion in the gland.

Besides the additional modules, an evident contrast between the models of Vargas et al. (2005) and Neal and Timmis (2003) is that the former also takes into consideration the internal state with regards to its drives and desires, as well as whether these have been accomplished. In this particular aspect, the works of Vargas et al. (2005) and Timmis, Neal and Thorniley (2009) are similar, as both seek to model the robot's desires and use it, along with the environmental feedback, to modulate the robot's actions in a manner that resembles an emotional reaction.

A more recent work proposed by Thenius, Zahadat and Schmickl (2013) follows in a similar direction, with the goal to improve the classical ANN by simulating hormone glands that influence the behaviour of individual neurons. Thenius, Zahadat and Schmickl (2013) argue that the approach of Neal and Timmis (2003) creates a strict separation between the hormonal and neural systems. On the other hand, their neural mechanism, called *EMANN* (EMotional Artificial Neural Network), allows the interaction between the neural and hormonal systems in both ways, generating a self-organizing feedback system.

Their work, however, lacks experimental evaluation on the system's performance regarding emotional aspects. Their performance analysis is based on a mathematical task and the system's performance is measured according to the increasing pace of fitness level. According to the evaluation criteria used by Thenius, Zahadat and Schmickl (2013), their work presented a significant improvement in relation to an ANN implementation without EMANN. However, the reader is left with no analysis on the emotional behaviour of the system.

Most of these models address all the requirements of an emotional intelligence (Section 1.3.2) by providing computational mechanisms that simulate the biological phenomena of neuroplasticity, associative learning and memory, as well as real-time learning and adaptation. However, none of the works above-mentioned fully address the all the requisites specified in Section 1.3.1, which concerns a situation-aware intelligence. Although these models manage to handle, in a sense, the unified meaning of multiple-stimuli in the environment, the temporal properties of these stimuli and how they interact with each other over time is not considered.

2.2.2 The Brain Emotional Learning Model

One of the most influential works in artificial fear conditioning is the *Brain Emotional Learning* (BEL) model, proposed by Morén and Balkenius (2001). Their model (Fig. 2.3) consists of interconnected modules of ANNs that simulate the role of neural circuitries involved in fear learning. It receives two types of inputs – environmental neutral stimuli and a reward signal – that are processed by four simulated neural regions: the thalamus, the sensory cortex, the amygdala and the orbitofrontal cortex.

The thalamus and sensory cortex simply relay input information to the orbitofrontal cortex and amygdala. This architecture is inspired by the concept of 'low and high roads to the amygdala' presented by LeDoux (1999). The sensory cortex receives information from the thalamus, which in turn receives information directly from the environment. Because the thalamic pathway is shorter (the 'low

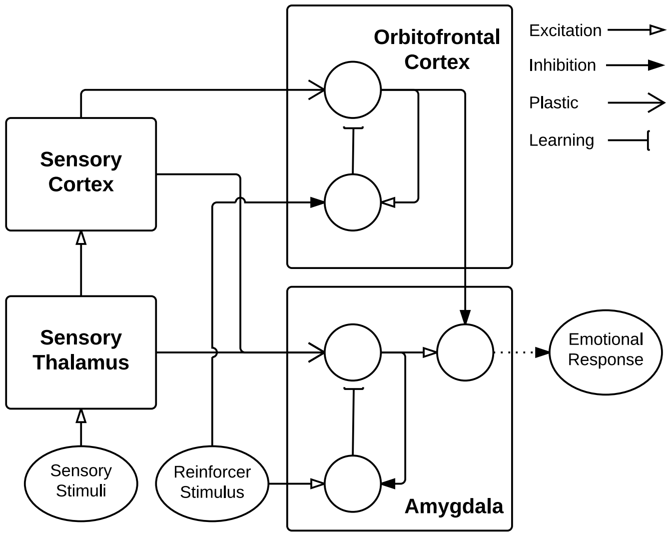


Figure 2.3: Fear-learning model proposed by Morén and Balkenius (2001). Each component of their model represents an ANN. Circles represent individual ANNs internal to the respective component.

road'), it provides the amygdala with low latency information about environmental stimuli. On the other hand, information projected through the thalamic-cortical pathway takes longer to reach the amygdala (the 'high road'), but provides a higher-level and more accurate representation of the sensed world.

The amygdala is responsible for assessing and predicting the emotional value of stimuli, based on the significance of the accompanied reward. Finally, the orbitofrontal cortex is responsible for inhibiting emotional associations of the amygdala that are no longer valid. This model has been tested for the most basic effects of classical conditioning – such as *fear acquisition*, *fear extinction*, *blocking*, *habituation* and *spontaneous recovery* – showing satisfactory results.

The BEL model was later improved by Morén (2002), with the addition of a module that simulates the contextual processing performed by the brain's hippocampal regions. BEL's hippocampus module has four main components: the *Bind subsystem*, the *Mem system*, the *Match system* and the *Context system*. The Bind subsystem is responsible for binding stimuli that are simultaneously detected. The Mem system generates expectations about stimuli manifestation at specific locations. These expectations are later compared with the actual stimuli in the Match system. Lastly, the Context system combines information from the Match and Bind systems to generate a contextual code that feeds the amygdala and orbitofrontal cortex.

With the aid of the hippocampal module, BEL is able to express fear responses based on contextual information. For example, one of the experiments performed by Morén (2002) consisted on presenting two different stimuli to the system, CS_0

and CS_1, sometimes separately and sometimes together. All single presentations of either CS_0 or CS_1 were followed by a reinforcing signal, whereas all simultaneous presentations were followed by nothing. The model gradually learned to differentiate between single and joint stimulus presentation. Further experiments performed by Morén (2002) with other patterns of stimulus presentation and location were also successful.

Despite BEL's success in discriminating sets of simultaneously presented stimuli, a few important questions were left unanswered. For instance, what would happen if the reinforcing signal was presented only after CS_0 was followed by CS_1 (represented by $CS_0 \rightarrow CS_1$)? Would the model understand that $CS_1 \rightarrow CS_0$ is different from $CS_0 \rightarrow CS_1$? According to Morén (2002), context 'can be either an abstract sequence of stimuli or a place defined by a number of stimuli at different locations around the animal', where 'sequence of stimuli' means a collection of stimuli values at a given time. It is clear that temporal factors are not considered in Morén's conceptualization of context, which is possibly the reason why the temporal order of stimulus presentation is never evaluated in his experiments. As a consequence, the BEL model does not meet all the requirements of a situation-aware intelligence, as specified in Section 1.3.1.

The simplest version of the BEL model (i.e., the version proposed by Morén and Balkenius (2001), which has no hippocampus module) became more popular among researchers. Based on the BEL model (Morén and Balkenius 2001), Lucas, Shahmirzadi and Sheikholeslami (2004) proposed a Brain Emotional Learning Based Intelligent Controller (BELBIC), which was later applied (somewhat adapted) to a large range of industrial (Babaie, Karimizandi and Lucas 2007; Lucas, Milasi and Araabi 2006; Jamali et al. 2010; Ravi and Mija 2014; Sharma and Kumar 2015), engineering (Azizur Rahman et al. 2008; Markadeh et al. 2011; Daryabeigi, Abjadi and Arab Markadeh 2014; Lotfi and Akbarzadeh-T. 2014a; El-Garhy and El-Shimy 2015) and robotics (Mehrabian, Lucas and Roshanian 2006; Jafari, Shahri and Shouraki 2013; Kim and Langari 2009; Jafarzadeh et al. 2008; Sharbafi, Lucas and Daneshvar 2010; Garmsiri, Najafi and Saadat 2013) applications. Most of these works have compared their BELBIC controllers with conventional controller approaches (e.g. PID, MLP, ANFIS and LLNF) and observed meaningful improvements in varied performance aspects when using BELBIC.

In 2010, Beheshti and Hashim (2010) published a review on BELBIC systems and demonstrated their performance for engineering ends. They compared BELBIC with a range of conventional controller approaches (such as PID, ANFIS and feedback linearization controller) for several engineering applications (such as micro heat exchanger, intelligent control of washing machine, dynamic power management, intelligent predictor for geomagnetic activity, and speed and flux control of an induction motor). Their analysis concluded that BELBIC showed better performance and results than the tested conventional approaches for real time control and decision systems.

BELBIC's popularity and performance improvement over traditional approaches in several application areas demonstrate its great potential as a controller. However, it is important to keep in mind that BEL, BELBIC and methods derived from them (Lotfi and Akbarzadeh-T. 2014b) are supervised learning algorithms with main application to engineering applications. Therefore, these models are not suitable for addressing the problem of online fear-learning for autonomous robotics. Additionally, these models do not handle the temporal properties of contextual information, thus not fulfilling the requirements specified in Section 1.3.

2.2.3 Adaptive Behaviour in Robot Soccer

The *Robot World Cup* (RoboCup) is an international scientific initiative that encourages the development of new technologies in artificial intelligence and autonomous robotics. It challenges researchers to advance the state-of-the-art robotics so that robots can play soccer in a realistic environment without human intervention. Such challenge promotes a highly dynamic and competitive scenario that demands the development of better adaptive skills and flexible decision-making at execution time. For this reason, we consider that the RoboCup competition represents an exemplary scenario for the application of a situation-aware fear-learning model of artificial intelligence. This is, in fact, the scenario of the case study that we have performed to evaluate our model, which is discussed in Chapter 6. Here, we present a brief summary of related work in the literature that aims at providing adaptive behaviour within the RoboCup context and may not necessarily be correlated models of emotions.

Because of the inherent teamwork nature of soccer, most research related to intelligent behaviour and decision making in robot soccer focuses on improving collaborative behaviour and pre-coordination (Nitschke 2005; Genter et al. 2016; Whiteson et al. 2003). These approaches are commonly based on pre-determined coordination strategies learned by means of supervised machine learning algorithms trained with a series of possible soccer situations generated via simulation. Consequently, an immutable strategy defined prior to the actual match is equally delivered to all opponent teams in the RoboCup competition. Nonetheless, different teams may use different tactics, and a specific pre-trained approach may fail against a particular opponent while being successful against another opponent team.

In real-life soccer, human players commonly use both pre- and post-coordinated strategies in conjunction. Soccer tactics usually involve the training of an agreed formation and strategy prior to the match, which is the pre-coordination phase. Nevertheless, unforeseen events may occur during the match, forcing teammates to communicate and adapt the team's strategy, which can be seen as a post-coordination phase. While pre-trained coordination is well developed and studied in the RoboCup competition (Nitschke 2005), the development of effective techniques for post-coordination is still overlooked (Ferrein and Steinbauer 2016).

The need for real-time adaptation capabilities has been previously addressed using case-based reasoning (Ahmadi et al. 2003; Ros et al. 2009). In these works, case-based reasoning approaches are used for post-coordination as a means to optimise players' positioning during the match. These works represent a great contribution towards post-coordination, flexible decision making and real-time adaptation. Among these, we highlight the work of Ros et al. (2009), consisting of a case-based approach for real-time adaptation in team coordinated attack when in the presence of opponent defenders. In their work, a *case* describes possible game plays, which are mainly defined in terms of problem and solution descriptions. The *problem description* represents the state of affairs and takes into consideration information about the ball's global position, the defending goal, the teammates' global positions and the opponents' global positions. The *solution description* dictates the sequence of actions that the robot team should perform in order to solve the described problem. The set of possible actions for the team is predefined and includes individual actions (e.g., kick the ball) or joint actions (e.g., pass the ball to a specific teammate).

The selection of a solution is mainly based on the similarity between the current and the previously solved problems, where the similarity measure is calculated using a Gaussian function. Ros et al. (2009) conducted a comprehensive experiment which demonstrated that their case-based approach performs better than a simpler pre-coordinated reactive approach (in which robots always try to go after the ball and attack as fast as possible individually) for reducing the chances of the opponent defence stealing the ball.

These results clearly demonstrate the relevance of real-time adaptation in the RoboCup scenario. Nonetheless, the approach proposed by Ros et al. (2009) cannot be applied to other action selection scenarios, as the problem and solution definitions are based on the evaluation of predefined features and actions. In addition, temporal information is not considered in the problem and solution descriptions, thus not meeting the requirements described in Section 1.3.1. In their approach, problem-solving is mostly based on independent snapshots of context, neglecting relevant information about context variation over time.

2.2.4 Context and Situation

Section 2.2.1 and Section 2.2.2 focused mostly on discussing studies that are mainly concerned with simulating the fear-learning phenomenon for robotics. In this section, we focus on discussing cognitive models particularly concerned with the contextual perception and processing of computational systems (not necessarily robots), though some of the mentioned studies may also address fear learning.

Rudy and O'Reilly (2001) proposed a contextual fear-conditioning model that relies on a theoretical framework (O'Reilly and Rudy 2001) based on the cortical and hippocampal regions of the brain. In their model, the cortex represents the context as a set of independent features, whereas the hippocampus binds these features into a unitary representation. Rudy and O'Reilly (2001) have implemented their framework on an ANN model, which was evaluated on a scenario that simulates a context fear-conditioning experiment performed with rats. The experiment aimed at evaluating the model regarding its capability to (1) enhance fear conditioning via pre-exposure to context and (2) induce pattern completion (when a subset of a learned pattern can recover the entire pattern).

Although successful in reproducing many fear conditioning effects, the contextual fear-conditioning model of Rudy and O'Reilly (2001) also disregards the temporal properties of context. According to Rudy and O'Reilly (2001), 'either context can be represented as a set of independent features (the features representation view) or these features can be bound into a unitary encoding that represents their co-occurrence (the conjunctive representation view)'. This implies that their unitary representation of context considers features that co-occur only, which excludes a large range of temporal possibilities between distinct features that are essential for a thorough contextual perception.

A model that considers temporal sequences has been designed by Harrison, Duggins and Friston (2006). Their study aimed at evaluating hippocampal responses to changes in probabilistic context by submitting subjects to a first-order Markov sequence, where the current event E_t is conditionally dependent on the previous event E_{t-1}, and the probability of transition between them is given by $p(E_t|E_{t-1})$. To model the task, they assumed that the subject was an ideal Bayesian observer, who starts with the belief that all events are equally likely and consecutive events are independent. As samples of events are sequentially presented, this ideal observer constructs a transition matrix consisting of the probabilities of transition between consecutive events.

Their model is similar to ours in the sense that learning and prediction are based on the temporal relationship of events. However, the design of the task given to their subjects, which reflects on their model of an ideal observer, considers that every event consists of only one stimulus. Although sufficient for the purpose of their experiment, which is analysing hippocampal responses to temporal context, this simplistic design does not reflect real world situations, in which events may consist of multiple stimuli.

Among recent research, we highlight the work of Subagdja and Tan (2015). They propose a model for *episodic memory* (a type of long-term declarative memory mainly processed by the hippocampus) using an extended adaptive resonance theory (ART) network. They argue that the accuracy of memory retrieval depends on the order and latency between memory cues, which matches the conceptual foundation of our work. They evaluate their approach in a transitive inference problem, which is a classical logical problem of comparing the value of things (e.g., given that A weighs more than B and B weighs more than C, then it can be inferred that A weighs more than C).

Among related work, Subagdja and Tan (2015) may be the most similar to our model with regards to temporal context. For instance, their definition of a situation (which they call as an *episode*) is equal to ours. However, as the authors themselves observe, their model's performance for tasks other than simple transitive inferences is still undetermined. An investigation on whether the model can be applied independently of the domain is still pending, as well as if it can handle more complex and real-world contextual information.

Additionally, our approaches also differ in the final purpose of temporal context. We are mostly concerned with predicting aversive events by creating a link between the 'feeling of fear' and the events that preceded an aversive stimulus in a past experience. This would provide robots with the chance to react and prevent unpleasant (possibly harmful) situations, as well as to increase their adaptation capabilities. On the other hand, the work of Subagdja and Tan addresses neither fear conditioning nor danger prediction/prevention. In their work, events' order has no emotional meaning and is not used for predicting future events. Their main focus is to facilitate retrieval, creation and update of neutral contextual memory.

2.3 The SAFEL Model – An Overview

SAFEL stands for *Situation-Aware FEar Learning.* It is a novel situation-aware computational system capable of providing robots with fear-learning skills in order to predict threatening situations to their own well-being or to their goals. SAFEL's model has been first proposed by us in Rizzi Raymundo, Johnson and Vargas (2015), partially implemented and tested in Rizzi et al. (2017) and improved by us in Rizzi, Johnson and Vargas (2016). In this section, we briefly introduce SAFEL's biological inspiration and design. A detailed explanation of SAFEL's model, implementation and performance analysis are presented in the next chapters.

SAFEL is a hybrid computational architecture inspired by the LeDoux's fear-learning model of the human brain (LeDoux 2003, 1999). According to LeDoux, fear learning greatly relies on two brain regions known as the *amygdala* and the *hippocampus*, as well as on a cognitive function known as the *working memory.*

Considerable evidence indicates the amygdala as an essential brain region for fear learning and memory (LeDoux 2003, 1999). It is responsible for processing the emotional significance of sensed stimuli by creating associations between neutral and aversive stimuli. On the other hand, the hippocampus is believed to be the main brain region involved in context processing (LeDoux 1999). In the hippocampus, sensory information is put together in order to form a unitary representation of the current state of affairs. Unlike information processed in the amygdala, representations formed in the hippocampus are not just visual, auditory or olfactory, but all of these at once, and include the way these sensations relate to each other both in intensity and temporal order. Finally, the working memory creates associations between the contextual memory formed in the hippocampus with the emotional memory formed in the amygdala, giving emotional meaning to the contextual information acquired in past experiences.

SAFEL's architecture is based on the task division proposed by LeDoux. Therefore, analogous to the LeDoux model, SAFEL is divided into three modules that work in an integrated and parallel manner: the *Amygdala Module* (AM), the *Hippocampus Module* (HM) and the *Working Memory Module* (WMM). Fig. 2.4 depicts the SAFEL model, illustrating how the three modules of the architecture are interconnected.

Environmental stimuli detected by the robot (e.g., by means of sensors' input or direct user input) are categorised into aversive and neutral stimuli by the robot's controller and delivered to the AM and HM. The AM is responsible for detecting threats by analysing the current values of aversive stimuli and associating them

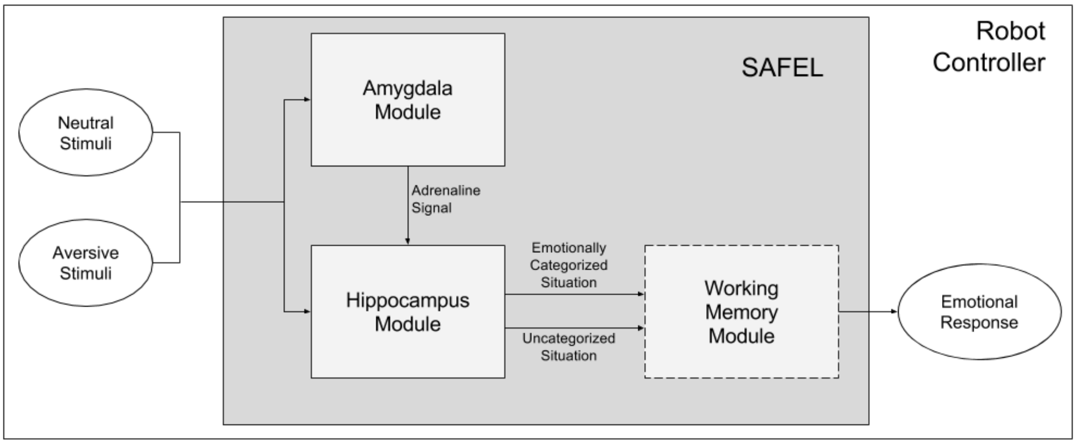


Figure 2.4: SAFEL model. Solid-border boxes represent areas of the brain whereas dotted-border boxes represent cognitive functions of the brain. The model receives neutral and aversive stimuli as input from the robot controller and outputs the corresponding emotional response back to the robot controller.

to simultaneously occurring neutral stimuli. This learning process is induced by means of a procedure analogous to the *cued fear conditioning* (Pavlov 1927).

In the classical fear conditioning, associative learning is induced by pairing a neutral stimulus (i.e., a stimulus that initially elicits no specific response from the individual) with an aversive stimulus (i.e., a stimulus that naturally elicits fear or discomfort, such as pain, hunger, etc.). Eventually, the previously neutral stimulus acquires emotional meaning and becomes able to elicit the state of fear by itself, even in the absence of the aversive stimulus. When this happens, we say that the neutral stimulus is now a *conditioned stimulus*, which elicits fear as a *conditioned emotional response*.

In SAFEL's model, the AM is also responsible for providing emotional feedback to the HM, which in parallel generates complex contextual representations of the sensed environmental stimuli. In the hippocampus, the amygdala's emotional feedback and the generated contextual information are associated.

Finally, pieces of contextual information and their emotional significance are memorised in the WMM. Later, any previously experienced pattern of contextual information will trigger the retrieval of that stored memory and its emotional meaning. Consequently, if a particular situation preceded the occurrence of an aversive stimulus in a past experience, the WMM will retrieve the same state of fear triggered by that situation in the past, warning the individual that an undesirable situation is likely to happen in the near future.

SAFEL's AM is based on a modified *artificial neural network* (ANN) proposed by Rizzi Raymundo and Johnson (2014), which allows robots to associate environmental stimuli at runtime based on the Pavlovian classical conditioning procedure Pavlov (1927). In the AM, this modified ANN is used to associate neutral and aversive stimuli at runtime. The ANN is pre-trained to generate a high output value whenever any aversive input is also high, and a low output otherwise, regardless of the value of neutral inputs. Associative learning takes place by autonomously adjusting the first-layer weights of the ANN according to the coincidence of input values. In other words, the association takes place whenever a strong neutral stimulus input and a strong aversive stimulus input co-occur. Eventually, the neutral stimulus is turned into a conditioned stimulus, becoming able to trigger by itself the same ANN output that the aversive stimulus would, even in its absence. The output of the ANN is said to be the *adrenaline signal*, which represents the current fear level of the system.

The HM is based on a conceptualization of situation awareness for expert systems formulated by Dey (2001). It is responsible for collecting, understanding and managing the states of the robot over time. To accomplish that, we have modelled and implemented the HM using SCENE (Pereira, Costa and Almeida 2013; Rizzi Raymundo et al. 2014), which is a powerful situation management platform that extends the JBoss Drools rule engine and its CEP (Complex Event Processing) platform (Bali 2013).

The HM receives two inputs: *events*, which are sets of environmental stimuli at a given point in time, and the adrenaline signal relayed by the AM. This module is responsible for assembling these events into pieces of information known as *situations*, which depict the robot's state-of-affairs during a particular period of time. Situations are later categorised in relation to their emotional meaning according to the subsequent emotional feedback from the amygdala. Situations preceding high adrenaline signals are categorised as *aversive situations*, and *safe situations* otherwise. Ongoing situations are left uncategorised and are said to be *neutral situations* because their true emotional meaning can only be determined sometime after their conclusion.

Finally, the WMM is the module of SAFEL where the association between context and "fear" takes place. In the WMM, the temporal patterns of situations are memorised and associated with their respective labels (safe or aversive). Here, two processes take place. First, a feature extraction is performed in order to generate compacted versions of situational information containing only the most relevant characteristics of the situations' temporal patterns. These compacted situations are then delivered to a binary classification tree for learning and prediction.

The tree associates the emotional meaning of a situation with its temporal pattern. Then, whenever an emotionally uncategorised situation arrives, the tree attempts to predict its emotional meaning by comparing the temporal properties of this situation and of those previously learned. If the tree finds a match for that situation pattern, then it returns the emotional category linked to that pattern, which will be either safe or aversive. Ultimately, SAFEL's final output is the emotional category retrieved by the classification tree and indicates whether something aversive is likely to happen in the near future.

SAFEL is designed to meet all the requirements discussed in Section 1.3. It performs flexible associations between aversive and neutral stimuli at execution time while handling the temporal and contextual information contained in the robot's environment. The processes described above for the AM, HM and WMM are discussed in detail in Chapter 3, Chapter 4 and Chapter 5 respectively. Each module of SAFEL is addressed in a dedicated chapter that discusses the biological inspiration, underlying technology, design and preliminary experiments (when applicable) of the respective module of SAFEL.

Chapter 3

Amygdala Module

The *amygdala* comprises two almond-shaped sets of neurons located deep in the medial temporal lobe of the brain (Fig. 3.1). Considerable evidence points the amygdala as the main brain region involved in fear learning and memory (LeDoux 2003, 1999; Phillips and LeDoux 1992; Herry and Johansen 2014).

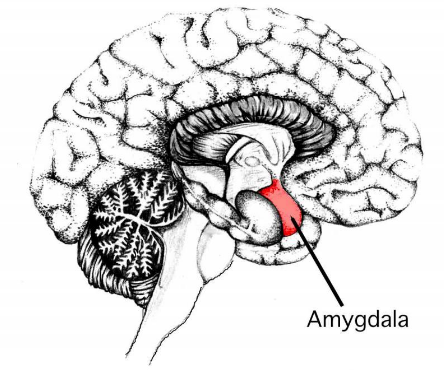


Figure 3.1: Amygdala region in the brain.

This chapter explores the biological background behind cued fear conditioning in the amygdala and proposes a computational model for simulating it, which represents the Amygdala Module (AM) of SAFEL. This chapter also partially contributes to answering the three research questions formulated in Section 1.4 by:

1. addressing the first two requirements of a situation-aware intelligence (Section 1.3.1) in combination with all the requirements of an emotional intelligence (Section 1.3.2);

2. presenting the design and implementation of a modified ANN aimed at providing online associative learning, which is the first approach of a hybrid model consisting of three distinct approaches; and

3. discussing experiments performed with simulated robots that demonstrate the successful use of the AM of SAFEL in a practical robotics application.

3.1 Biological Background

This section introduces the biological background that inspired the modelling and implementation of the AM of SAFEL. We start in Section 3.1.1 with one of the most simple forms of associative learning, known as *classical conditioning*. Section 3.1.2 explains how classical conditioning can be unified with fear learning to generate *classical fear conditioning*. Finally, Section 3.1.3 presents the brain regions and mechanisms behind the phenomenon of classical fear conditioning that have inspired the design of the AM.

3.1.1 Classical Conditioning

Natural environments change often, which makes adaptation an essential skill for the survival of most organisms. For that reason, most animals are equipped with a range of biological systems that facilitate their adaptation to different situations, among which stands out the ability to learn.

Classical conditioning, also known as *Pavlovian conditioning*, is one of the most basic forms of learning and involves the association of a behavioural response with an event that normally does not trigger that response. This phenomenon, first documented by Pavlov (1927), is recurrent among humans and other animals.

In the 1900's, the Russian physiologist Ivan Pavlov was studying the digestion of dogs by observing the salivation processes in dogs when being fed. While performing experiments with a dog, Pavlov observed that the dog would salivate whenever his laboratory assistant, who used to feed the dog, entered the room. The presence of the assistant would trigger the dog's salivation even when he was not holding any food. Pavlov suspected that the dog had associated the idea of food with his assistant's presence. As a consequence of this association, the assistance's presence came to induce in the dog the same behavioural response triggered by the presence of food: salivation.

According to Pavlov, some reflexes are 'hard-wired' and, therefore, do not need to be learned. For example, dogs do not need to learn to salivate when they smell food, because they are born with this behaviour. This kind of reflex, which is natural and automatic, is called *unconditioned response* (UR). The stimulus that triggers an unconditioned response is called *unconditioned stimulus* (US). In the example of Pavlov's dog, the smell of food is an US that triggers salivation as an

UR.

After such unexpected observation, Pavlov decided to start a monitored experiment with his dog. First, he measured the dog's salivation whenever he presented a bowl of food and whenever he rang a bell. Salivation was significantly increased with the presentation of food, but not with the ringing of a bell. In this case, the ringing of a bell is said to be an *neutral stimulus* (NS) in regards to the salivation response because it does not naturally trigger salivation.

Next, Pavlov repeatedly paired the neutral and unconditioned stimuli by consistently presenting these two stimuli simultaneously to the dog. After repeatedly pairing the NS with the US (bell ringing and food smell, respectively), Pavlov rang the bell without presenting food. The salivation response meaningfully increased this time, even in the absence of food. Pavlov concluded that the repeated NS-US pairing worked as a *conditioning procedure*, inducing the dog to learn a new stimulus-response relationship, which is the salivation in response to the ringing of a bell.

A neutral stimulus that comes to trigger a response after being associated with an US is called *conditioned stimulus* (CS). The respective response, when triggered by the CS, is called a *conditioned response* (CR). After association with the smell of food (which is an US), the ringing of a bell becomes a CS, which triggers salivation as a CR.

Unlike URs, a CR can be extinguished if the respective US is persistently presented in the absence of the CS and vice-versa. For example, the dog will diminish its salivation response to the bell if food is repeatedly presented on the absence of the bell's sound and vice-versa. For that reason, CRs are considered to be unstable. On the other hand, URs are said to be stable, because they are native and cannot be extinguished, regardless the circumstances in which the respective US is induced.

3.1.2 Fear Learning and Conditioning

Conditioning procedures like the one described in Section 3.1.1, which rely on the association with positive reinforcing stimuli, are also known as *appetitive conditioning*. The positive reinforcing stimulus is said to be the *appetitive stimulus* (or *appetitive US*), which in the example of Pavlov's dog is food. Other examples of appetitive US are water, warmth, breeding, etc. A natural behaviour for animals conditioned with an appetitive stimulus, such as food, is to often seek for the CS that signals the availability of that appetitive US (Andreatta and Pauli 2015).

Similarly, any conditioning procedure that relies on an aversive reinforcing stimulus is said to be an *aversive conditioning*. In aversive conditioning, an animal associates the feeling of fear with an NS after it is repeatedly paired with an aversive US. An *aversive unconditioned stimulus* is any stimulus that naturally elicits fear or anxiety in the animal. In other words, the animal is born with the knowledge that such stimulus is aversive. Some examples of aversive US are pain, hunger, sensory impairment (such as losing visibility in dark places), aggressive facial expression of other animals, etc. A natural behaviour for animals conditioned with an aversive stimulus, such as pain, is to avoid contact with the CS that signals the presence of the aversive US (Andreatta and Pauli 2015).

By pairing an NS and an aversive US in a process similar to that of classical conditioning, the NS can acquire emotional value and become able to trigger fear reactions by itself, even in the absence of the aversive US. Since the NS did not trigger fear reactions before, we say that the animal has learned to fear it through a *fear conditioning* procedure. As a consequence, the NS becomes a (aversive) CS.

The classical foot-shock experiment performed with rats and mice demonstrates this phenomenon (Phillips and LeDoux 1992). In the experiment, a rat is placed into an apparatus and receives auditory cues (the NS) paired with a mild electrical foot shock (the US). The shock naturally elicits fear in the rat, which freezes in response (the UR). After repeating this procedure a few times, the rat associates the NS with the US and starts to freeze in response to the auditory cue even in the absence of an electrical shock. At this point, the auditory cue has become a CS. The freezing reaction in response to the CS is said to be either a CR or a *conditioned emotional response* (CER). Fear learning is called as *cued fear learning* when it induces association of a CS that is a discrete stimulus, such as a tone, with an aversive US.

3.1.3 Brain Mechanisms of Fear

The famous *Hebbian Theory* (Hebb 1949), also know as *Hebbian Rule*, postulates that:

> When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

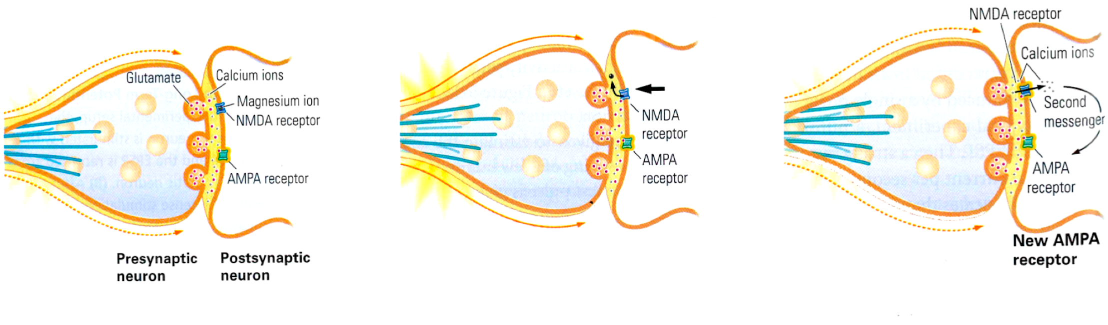
In other words, according to Hebb (1949), associative learning at neural level involves the simultaneous electrical stimulation of two interconnected neurons. This theory has been demonstrated by numerous studies (Stuchlik 2014; Bliss and Lømo 1973; Bliss and Collingridge 1993) and is believed to be carried out by a phenomenon known as *long-term potentiation* (LTP) (LeDoux 1995; Stuchlik 2014), which strengths the communication between two neurons and can last for hours, weeks or even longer (Cooke and Bliss 2006).

Neural communication takes place at the *synapse*, which is the structure that composes the junction between two neurons. Signals are transmitted between neurons by means of chemical substances called *neurotransmitters*, which are released in the synapse by one neuron's *axon* and forwarded to the *dendrites* of another (the target) neuron. In the membrane of the target neuron's dendrite, neurotransmitters are bound to specific *receptors*, leading to either *excitation* or *inhibition* of the target neuron, completing the neural messaging cycle. The LTP phenomenon consists in enhancing the reception or the release of neurotransmitters (Kolb and Whishaw 2004).

There are different types of LTP, among which *NMDAR-dependent LTP* is the most extensively studied (Malenka and Bear 2004; Martin, Grimwood and Morris 2000). *NMDAR* is a receptor in the membrane of the target neuron that acts as a detector of neural activity coincidence. It is usually blocked by a magnesium ion, which only strong electrical stimulation is able to remove (Fig. 3.2a). For this reason, weak stimulation is able to activate *AMPA receptors* only. Nonetheless, weak electrical stimulation becomes able to activate the NMDA receptor after the magnesium ion is removed by a strong electrical stimulation (Fig. 3.2b). The activation of NMDAR allows Ca^{2+} influx in the target neuron, which in turn increases the function or number of AMPA receptors in the membrane of the target neuron, enhancing the responsiveness of that neuron to weak electrical stimulation (Fig. 3.2c).

LTP is not the only process that affects synapses' efficiency. *Long-term depression* (LTD) is a process similar to LTP, but instead of strengthening, it weakens synapses' ability to transmit signals between neurons. Both LTP and LTD are induced by influx of Ca^{2+} in the target neuron. The factor that determines wheter LTP or LTD will occur is related to the concentration and duration of Ca^{2+} influx in the target neuron. High and brief concentrations of Ca^{2+} induce LTP, whereas prolonged and moderate concentrations of Ca^{2+} induce LTD (Yang, Tang and Zucker 1999).

Synapses' ability to change their strength in signal transmission according to neural activity level, called as *synaptic plasticity* or *Hebbian plasticity*, is known to play an important role in classical conditioning (Roberts and Glanzman 2003). LTP has been demonstrated to occur in the amygdala regions and is believed to underlie fear conditioning mechanisms mediated by the amygdala (LeDoux 1995; Barad, Gean and Lutz 2006). In addition, the NMDAR mechanism is considered



(a) Weak electrical stimulation is not able to remove the magnesium ion from NMDA receptor and, thus, activate only AMPA receptors.

(b) A strong electrical stimulation can remove the magnesium ion from the NMDA receptor.

(c) Now glutamate released by weak electrical stimulation can activate the NMDAR, which allows the influx of Ca^{2+} , increasing the number and function of AMPA receptors.

Figure 3.2: LTP process in the synapse (Kolb and Whishaw 2004).

by researchers as the 'neural instantiation' of the Hebbian Theory (LeDoux 1995).

For a simplified example of this relation, consider the foot-shock experiment discussed in Section 3.1.2. A weak electrical stimulus could come from a conditioned stimulus (CS), such as the auditory cue for the rat. On the other hand, a strong electrical stimulation could come from an aversive unconditioned stimulus (US), such as the foot shock. The target neuron, in turn, could be a neuron that meaningfully contributes to triggering fear responses, such as freezing. Then, the pairing of weak (from the auditory cue) and strong (from the shock) electrical stimuli generates LTP, which makes the target neuron more responsive to the weak stimulus. In the future, the weak stimulus will be able to activate the target neuron by itself, allowing the auditory cue to trigger freezing responses. LTD would occur if the CS and the US are repeatedly presented in the absence of each other, leading the rat to stop responding to the auditory cue.

3.2 Underlying Technology

This section presents the algorithm used to simulate classical fear conditioning in an artificial agent. Section 3.2.1 briefly introduces the artificial neural network (ANN) algorithm while Section 3.2.2 discusses at a higher level of abstraction our approach for implementing the LTP and LTD phenomena in an ANN. This approach is formally defined later in Section 3.3.

3.2.1 Artificial Neural Networks

Artificial neural networks (ANNs) are collections of units known as *neurons* or *nodes* (Callan 1999), which are interconnected according to a pre-specified architecture that determines the network's style (for example, in this work we use the classical *feedforward* neural network). Each neuron combines a number of input signals and calculates an output value, which is sent to other neurons by means of weighted connections.

ANNs are divided into *layers*, each containing a number of neurons. An ANN receives information from the environment through its first layer, known as the *input layer*, and communicates back to the environment through its last layer, known as the *output layer*. All other layers are called *hidden layers*. Fig. 3.3 shows an example of ANN.

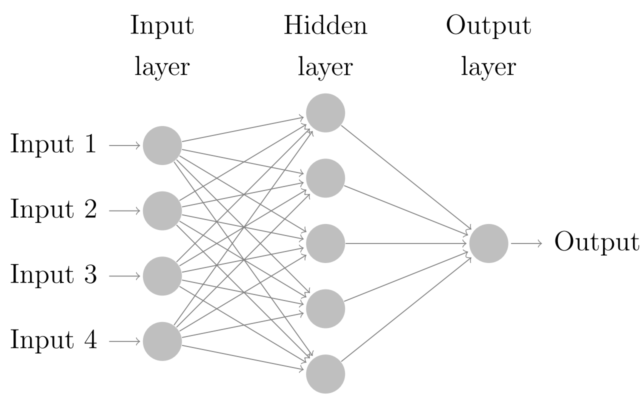


Figure 3.3: Example of a feedforward artificial neural network.

Each neuron, except by those in the input layer, receives a series of input values from the previous layer, which are summed according to equation Eq. 3.1. The input to a neuron i is, therefore, the result of this summation given by net_i in Eq. 3.1, called *net input*.

$$net_i = b_i + \sum_j x_j w_{ij} \tag{3.1}$$

where x_j is the output of neuron j from the previous layer, w_{ij} is the weight of the connection between neurons j and i, and b_i is the *bias* of neuron i. The net input is then processed by an *activation function* that provides the neuron's output, which in our case is the *sigmoid function* (equation Eq. 3.2).

$$f(net_i) = \frac{1}{1 + e^{-net_i}} \tag{3.2}$$

Connections' weights are initially random, and thus need to be specifically adjusted to the ANN's task. This adjustment is performed in a sequence of gradual steps by a *training algorithm*, such as the *backpropagation* algorithm. After the *training phase*, the ANN is expected to output the correct response with small margin of error given a set of input values.

3.2.2 Artificial Synaptic Plasticity

The Amygdala Module (AM) is based on the concepts of classical fear conditioning, introduced in Section 3.1, which in this work we simulate using a modified ANN. As discussed in Section 3.1.3, the neural mechanism of classical conditioning consists in strengthening the signalling efficiency of synapses. On the other hand, synapses are represented by connections between neurons in the ANN, while their signalling strength is represented by the weights of these connections. Therefore, the synaptic plasticity phenomenon can be simulated in a simplified manner by gradually changing the weights of the ANN according to their activity coincidence.

The AM is based on a modification to the classical ANN to generate an *artificial synaptic plasticity* (ASP) mechanism. This ASP mechanism, proposed by us in a previous work (Rizzi Raymundo and Johnson 2014), induces associative learning by adjusting the weights of the first-layer neurons of the ANN. The ANN's inputs, outputs and parameters are defined and the ANN is trained as normal, according to the task it is designed to perform. The input set included in the pre-training phase is said to be the set of unconditioned stimuli (US's), and the output of the ANN at this point is said to be the unconditioned response (UR). After the pre-training phase, new input neurons representing the neutral stimuli (NS's) are added to the ANN in an *ad-hoc* manner with zeroed first-layer weights. Because their weights are zeroed, NS inputs are unable to influence the ANN's output. After a procedure similar to classical conditioning, in which the US and the NS are jointly presented to the robot a few times, the first-layer weights of NS input neurons are gradually adjusted. Eventually, NS input neurons become able to influence the output of the ANN in the same way US input neurons would, at which point we say the NS inputs have been turned into conditioned stimulus (CS) inputs, and the output of the ANN is a conditioned response (CR). The design and implementation of the ASP mechanism described above is formally defined in Section 3.3.

For instance, in one of our experiments, a simulated robot is equipped with five touch sensors and 16 distance sensors around its body, as well as two wheels for locomotion. The ANN controlling the robot's movements is initially trained to make turns whenever the robot's touch sensors detect a collision. At this point, the ANN has five inputs, whose values come from the five touch sensors and indicate collisions detected around the robot's body; and two outputs, which provide the speeds for the left and right wheels of the robot. After the training phase, 16 new inputs are added to the ANN (with zeroed weights), depicting obstacles' distances detected by the 16 distance sensors of the robot. Therefore, inputs coming from the five touch sensors represent the US's, while inputs coming from the 16 distance sensors represent the CS's. After a few bumps during environmental exploration, the robot associated the inputs of the distance sensors with the inputs from the touch sensors and became able to make the turns before colliding, using the information from the distance sensors only, which were learned at runtime. This experiment and its results are discussed in more details in Section 3.4.2 and have also been published in Rizzi Raymundo and Johnson (2014).

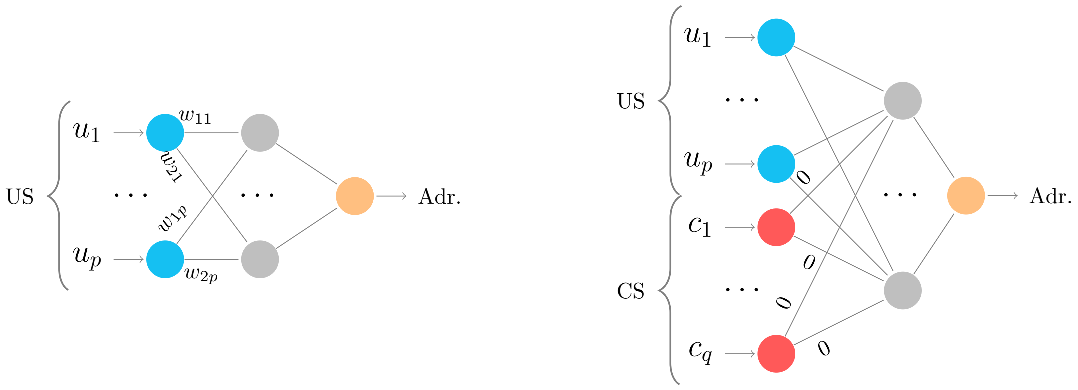
The ASP mechanism, as initially proposed by us in Rizzi Raymundo and Johnson (2014), induces general associative learning, such as the food-bell association of Pavlov's dog discussed in Section 3.1.1. In this thesis, we modify our ASP mechanism to induce specifically cued fear learning, such as the shock-tone association discussed in Section 3.1.2. Other learning approaches using a similar mechanism (usually referred to as *Hebbian learning*) have been previously proposed (Balkenius and Morén 1998; Miller, Barnet and Grahame 1995; Timmis, Neal and Thorniley 2009). However, our approach (Rizzi Raymundo and Johnson 2014) provides a number of additional features that better accommodate SAFEL's design requirements and goals. In Section 3.5, these additional features are discussed in details and compared with traditional Hebbian learning approaches.

3.3 Model

As discussed in Section 3.1, an aversive unconditioned stimulus (US) is any stimulus that naturally elicits fear in the animal, such as pain, hunger and loss of senses. Animals do not need to learn to fear aversive US's because they are already born with this behaviour. This is analogous for the Amygdala Module (AM) of SAFEL, which should be pre-trained to recognise aversive US's and output an appropriate fear response.

Fig. 3.4 depicts the initialization process of the ANN in the AM. Initially, the ANN is trained to output an *adrenaline signal* as the unconditioned response (UR), which indicates the current level of fear of the robot. For the experiments discussed in Section 3.4 and Chapter 6, we have extended the standard MATLAB (MATLAB 2014) implementation of the feedforward ANN, which uses the *Levenberg-Marquardt* algorithm (Moré 1978) to train the ANN.

During the training phase, the input set of the ANN includes aversive US's only (Fig. 3.4a), which may be any stimuli considered undesirable to the robot by



(a) The ANN is trained with aversive unconditioned stimuli only to output an adrenaline signal according to the averseness level of each input stimuli.

(b) After the training phase, neutral stimuli are added to the input set of the ANN, with zeroed first-layer weights.

Figure 3.4: Initialization of the artificial neural network of the Amygdala Module.

its designer (e.g., collision, fall, low visibility, low battery), and should take into consideration the specific task that the robot has to perform. The ANN should be trained to output a high adrenaline signal whenever one or more aversive US inputs have high values, where each input should be normalised in the interval [0, 1].

After the training phase, any other stimuli that are not considered aversive by the robot's designer are added to the ANN input set with zeroed first-layer weights (Fig. 3.4b). These are the neutral stimuli (NS's). At runtime, while the robot explores the environment, the ANN of the AM uses the artificial synaptic plasticity (ASP) mechanism proposed by us in Rizzi Raymundo and Johnson (2014) to create associations between any persistently co-occurring pairs of NS and aversive US. These NS's are eventually transformed into aversive CS's, thus becoming able to independently generate high adrenaline signal outputs. The mechanism through which this associative learning takes place is formally described next.

The inputs of the ANN are divided into two groups: the aversive US group, depicted by the vector $\vec{u}$, of size p; and the CS group, depicted by the vector $\vec{c}$, of size q. Together, these two stimuli vectors compose the input of the ANN, which is a vector $\vec{s}$ of size $p+q$ representing the set of all environmental stimuli that the robot is able to sense. Therefore:

$$\vec{s} = [u_1, u_2, ..., u_p, c_1, c_2, ..., c_q] \,, \tag{3.3}$$

$$\text{where } \forall s_i \in \vec{s}\,, s_i \in [0,1] \text{ , and}$$

$$net_i = b_i + \sum_{k=1}^{p} u_k w_{ik} + \sum_{j=1}^{q} c_j w_{ij} \,. \tag{3.4}$$

A particular CS cannot be an US at the same time and vice-versa. Therefore $\vec{u} \cup \vec{c} = \vec{s}$ and $\vec{u} \cap \vec{c} = \emptyset$. From now on in this section, we reserve the variables $k \in \{1, 2, ..., p\}$ and $j \in \{1, 2, ..., q\}$ for indexing US and CS elements, respectively.

The ANN is expected to be pre-trained so that the higher the value of an US u_k, the stronger (or more aversive) it is. The fear learning mechanism of the AM consists of gradually changing the first-layer weights of neutral stimuli so that they become able to activate the second-layer neurons with the same pattern that unconditioned stimuli would. Therefore, after each query of the neural network, all first-layer weights related to CS inputs should be updated by a delta. Note that weights w_{ik}, which are related to US inputs, should not be changed. The new weight values are given by:

$$w_{ij} = w_{ij} + \Delta w_{ij} \,. \tag{3.5}$$

The value of Δw_{ij}, which represents the synaptic adjustment, should consider not only the amount by which a given pattern is associated (determined by long-term potentiation (LTP)) but also the amount by which the same pattern is extinguished/dissociated (determined by long-term depression (LTD)). The variables LTP_{ij} and LTD_{ij} in Eq. 3.6 control the level of association of stimulus c_j by strengthening (if $LTP_{ij} > LTD_{ij}$) or weakening (if $LTD_{ij} > LTP_{ij}$) the weight w_{ij} of neuron c_j.

$$\Delta w_{ij} = \alpha_j (LTP_{ij} - LTD_{ij}) \,, \tag{3.6}$$

where $\alpha_j \in [0, 1]$ is the rate at which the neural network learns to associate/dissociate stimulus c_j. Hence, $\alpha_j = 0$ means that no association will occur, and the closer α_j is to 1 the faster is the system's associative learning. α_j is a user-defined parameter of SAFEL called as the *association rate* (AR) of stimulus c_j.

The value of w_{ij} cannot be increased/decreased indefinitely because the ANN outcome could be much higher or lower than the outcome produced by aversive US's in the same situation, diverging from the concept of cued fear conditioning. In order to avoid that, w_{ij} must be kept in a range $[w'_{ij}, w''_{ij}]$, where w'_{ij} is the initial value of w_{ij} and w''_{ij} is the desired maximum value of w_{ij} after the conditioning procedure. Note that, in our case, $w_{ij} = 0$ because all NS inputs are initialized with zeroed first-layer weights.

The closer w_{ij} is from w''_{ij}, the closer it is from a complete association and the more capable it is to influence the adrenaline output of the ANN. Analogously, the closer w_{ij} is from $w'_{ij} = 0$, the closer it is from a complete dissociation. From this reasoning it follows that:

$$LTP_{ij} = \left(w''_{ij} - w_{ij}\right) \times \Delta a_j \,, \tag{3.7}$$

$$\begin{aligned} LTD_{ij} &= \left(w_{ij} - w'_{ij}\right) \times \Delta d_j \\ &= w_{ij} \times \Delta d_j \,. \end{aligned} \tag{3.8}$$

The variables Δa_j and Δd_j, both in the interval [0,1], dictate the degree of synaptic activity coincidence between c_j and $\vec{u}$, and how it affects the pace of association and dissociation, respectively. We will return to these variables later on.

We must also consider that a given CS c_j may have a stronger association with a particular aversive US u_k than to the others. The level of association between c_j and u_k is called as the *sensitivity* of stimulus c_j to stimulus u_k. The mapping of sensitivities from $\vec{c}$ to $\vec{u}$ is given by the matrix Θ, of size $q \times p$. The element $\theta_{jk} \in [0, 1]$ of Θ is the sensitivity of stimulus c_j to stimulus u_k, where 0 means no association at all and the closer θ_{jk} is to 1 the stronger the association between c_j and u_k. The sensitivity matrix Θ is an optional user-defined parameter of SAFEL and, if correctly calibrated, can prevent the robot from learning 'superstitions', i.e., patterns that are no more than random coincidences. The relevance of the sensitivity matrix, as well as how it should be defined, is discussed with a practical example in Section 3.4.2.

According to our definition, the value of w''_{ij} should be defined so that, after a complete association, c_j can activate neuron i of the second layer with the same pattern that vector $\vec{u}$ would, which implies Eq. 3.9:

$$c_{j_{max}} \; w''_{ij} = \sum_k u'_k \; w_{ik} \,, \tag{3.9}$$

where the constants $c_{j_{min}}$ and $c_{j_{max}}$ are, respectively, the minimum and maximum values that c_j can assume. The problem with this approach is that all CS's are considered to assume their minimum values in normal conditions, which should increase whenever something uncommon or undesirable occurs, analogous to aversive stimulus. However, unlike aversive US's (whose behaviour is known beforehand by the robot's designer and learned by the ANN at the training phase), the behaviour of an NS or CS is only revealed at runtime and, therefore, is unpredictable before that. In the real world, there are many examples of NS that assume their average or highest values in neutral situations. For instance, big cities commonly suffer from high levels of noise pollution. Therefore, input values coming from a sound sensor would be, in average, high for robots living in big cities. This does not imply that the robot is constantly in danger.

To deal with this limitation, we have adapted the mechanism proposed in Rizzi Raymundo and Johnson (2014) to consider the average values of an NS, instead its minimum and maximum values only. Therefore, Eq. 3.9 becomes

$$\begin{aligned} c'_j \; w''_{ij} &= \sum_k u'_k \; w_{ik} \\ w''_{ij} &= \frac{\sum_k u'_k \; w_{ik}}{c'_j} \,, \end{aligned} \tag{3.10}$$

where c'_j is the farthest value that c_j can assume from its median value $\tilde{c_j}$. Considering that $c_{j_{min}} = 0$ and $c_{j_{max}} = 1$ (see Eq. 3.3), then c'_j is defined by Eq. 3.11:

$$\begin{aligned} c'_j &= \max((c_{j_{max}} - \tilde{c_j}), (\tilde{c_j} - c_{j_{min}})) \\ &= \max((1 - \tilde{c_j}), \tilde{c_j}) \,, \end{aligned} \tag{3.11}$$

where $\tilde{c_j}$ is the median of the latest pre-defined number of input values for c_j.

We use the latest pre-defined number of input values (instead of all the previously occurring input values) to allow for environmental adaptation. Robots may switch environments, or their environment may be altered in some way. The actions of a robot should consider mainly its most recent environmental conditions. For example, a robot living in a big city with high levels of noise pollution may be moved to a peaceful countryside place. In this new environment, loud noises have a different meaning and may be crucial to detect threats. Therefore, the robot's basis for comparison should be adjusted. The old sound level detections are not as relevant as the most recently detected sound levels when identifying imminent threats and, thus, should be gradually forgotten over time.

The strength τ and weaknesses φ of stimulus c_j are given, respectively, by Eq. 3.12 and Eq. 3.13:

$$\tau(c_j) = \left| \frac{c_j - \tilde{c_j}}{c'_j} \right| \quad \text{and} \tag{3.12}$$

$$\varphi(c_j) = \frac{c'_j - |c_j - \tilde{c_j}|}{c'_j} \,. \tag{3.13}$$

In other words, the higher the value of $\tau(c_j)$ the stronger the stimulus c_j, and the higher the value of $\varphi(c_j)$, the weaker the stimulus c_j.

In biological synaptic plasticity, an association between a pair of CS and US occur when their signals are simultaneously strong. This is analogous for the ANN of the AM. Therefore, the higher the values of $\tau(c_j)$ and u_k, the higher the association between c_j and u_k (i.e., the higher the association factor Δa_j). However, c_j may be associated with more than one US, at different sensitivity values. Thereafter, it is more accurate to state that Δa_j is proportional to the average strength of $\vec{u}$ weighted by the respective sensitivities. This implies Eq. 3.14:

$$\Delta a_j = \tau(c_j) \times \frac{\sum_k \theta_{jk} u_k}{\sum_k \theta_{jk}} \,. \tag{3.14}$$

Analogously, the dissociation (i.e., the extinction of an association) of a CS c_j with an US u_k should occur when these stimuli are no longer paired. Therefore, the higher the value of $\varphi(c_j)$ and the higher the mean of $\vec{u}$ weighted by the respective sensitivities, the higher the dissociation between c_j and $\vec{u}$ (i.e., the higher the dissociation factor Δd_j), which leads to Eq. 3.15:

$$\Delta d_j = \varphi(c_j) \times \frac{\sum_k \theta_{jk} u_k}{\sum_k \theta_{jk}} \,. \tag{3.15}$$

According to classical conditioning, the stronger the CS and the weaker the US signals, the higher the dissociation between them as well. However, the dissociation factor of the AM does not take this into consideration, as it is not of our interest that the robot forgets the learned association when the US becomes absent. This is because, as the robot becomes better in predicting an aversive US and perhaps acting towards avoiding it, the aversive US will become increasingly absent as a consequence of the robot's preventive actions. By forgetting the learned fear associations because of its own preventive actions, the robot would likely enter into an endless learning and forgetting cycle. For this reason, SAFEL's AM only promotes dissociation when the US is present in the absence of the associated CS.

By replacing Eq. 3.7, Eq. 3.8, Eq. 3.14 and Eq. 3.15 in Eq. 3.6 and simplifying, we find Eq. 3.16, which provides the synaptic adjustment Δw_{ij} necessary to generate the associative learning as discussed in Eq. 3.5.

$$\Delta w_{ij} = \alpha_j \times \frac{\sum_k \theta_{jk} u_k}{\sum_k \theta_{jk}} \times \left[\tau(c_j)\left(w''_{ij} - w_{ij}\right) - \varphi(c_j) w_{ij}\right] \,, \tag{3.16}$$

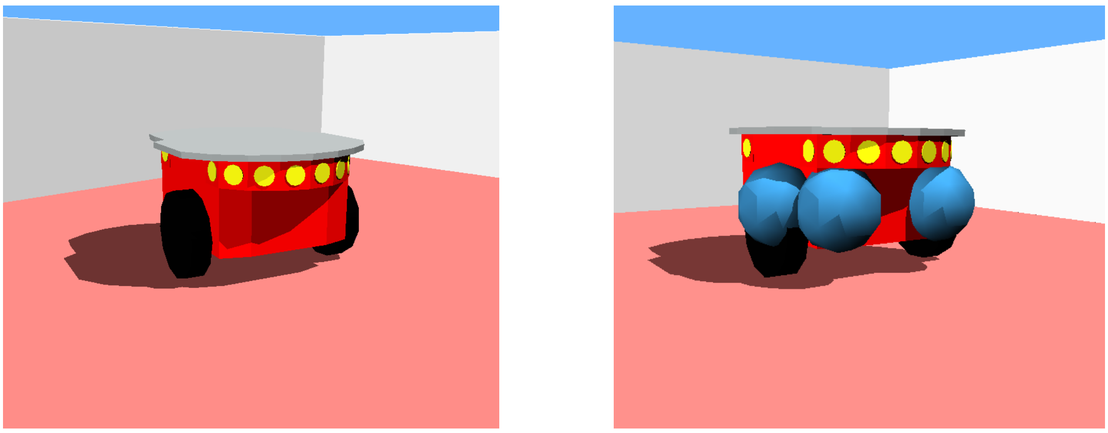
where w''_{ij} is given by Eq. 3.10. If a particular pair of associated stimuli, say c_j and u_k, have high input values at the same time, a net-input extrapolation may occur. This is because after being associated with u_k, c_j is able to mimic the effect of u_k in the ANN. Therefore, if both inputs are high, the second-layer neurons will receive a total input twice as high as they would if association had not occurred. For this reason, the net input values of second-layer neurons must be restricted according to Eq. 3.17, where $(u_k w_{ik})_{min}$ and $(u_k w_{ik})_{max}$ are, respectively, the minimum and maximum values that $u_k w_{ik}$ can assume.

$$b_i + \sum_k (u_k w_{ik})_{min} \; < \; net_i \; < \; b_i + \sum_k (u_k w_{ik})_{max} \, . \tag{3.17}$$

3.4 Preliminary Tests

As mentioned in Section 3.2.2, the artificial synaptic plasticity (ASP) mechanism introduced in Section 3.3 was initially meant for general purpose associative learning and conditioning. Modifications to this mechanism for providing specifically fear learning were performed later by us in Rizzi, Johnson and Vargas (2018). The preliminary experiments presented next evaluate the original ASP mechanism, as proposed by us in Rizzi Raymundo and Johnson (2014). The modified mechanism (Rizzi, Johnson and Vargas 2018) is evaluated in the AM together with the rest of SAFEL's modules in Chapter 6.

We have used Webots (Michel 2004) for performing the experiments discussed in this section. Webots is a robot-simulator software able to reproduce a great variety of robots, devices and environments. Through Webots, we have created and used a customized version of the simulated *Pioneer 2* robot (ActivMedia Robotics 1999). Webots' original simulation of Pioneer 2 includes a two-wheel differential-drive mobile robot, equipped with 16 distance sensors (Fig. 3.5a). However, our experiments require that the robot is equipped with touch sensors as well. For this reason, we have added five touch sensors to the robot, designed as spheres in the front, sides and back of the robot's body (Fig. 3.5b).



(a) Webots' original simulation of the Pionner 2 robot.

(b) Customized addition of 5 touch sensors in the Pioneer 2 robot.

Figure 3.5: Simulations of the Pioneer 2 robot in the Webots simulator software.

We have performed two sets of experiments[1]. The first, discussed in Section 3.4.1, has been based on Pavlov's dog example, in which the conditioning process is very simple and requires an one-to-one stimulus association (e.g., association of a sound to the smell of food). The second set of experiments, discussed in Section 3.4.2, consists of more complex associative learning, where conditioning involves the association of multiple stimuli.

3.4.1 Experiment I – Single CS Association

The simplicity of Pavlov's dog experiment allowed him to easily observe the behavioural changes derived from both association and dissociation processes. By limiting the number and complexity of stimulus associations (food smell and bell sound), Pavlov considerably facilitated the control and observation of the experiment progression. In other words, to generate associative or dissociative behaviour, Pavlov needed to change only a few variables in the experiment, such as adding or removing the presence of a particular stimulus. In order to clearly observe the effects of associative and dissociative learning when using the ASP mechanism, we have conducted an experiment similar to that performed by Pavlov, where the robot associates collisions with a direct human feedback.

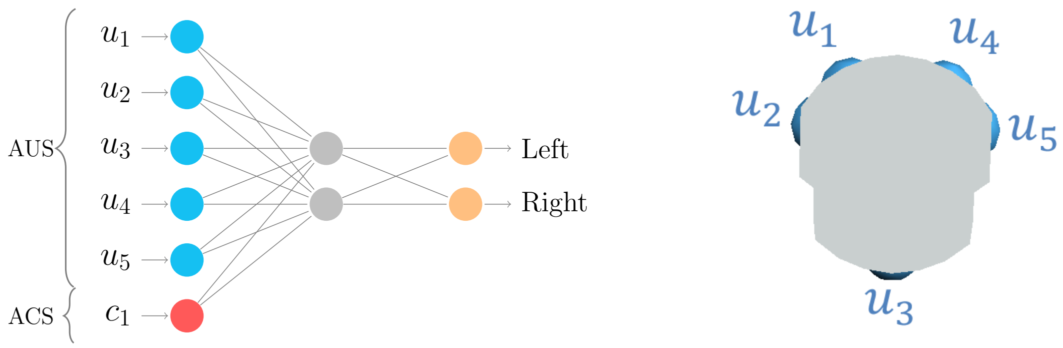
Experiment Setup

In this experiment, the wheels of the robot from Fig. 3.5b are controlled by an ANN. We have trained this ANN to turn the robot away from obstacles based on information coming from the robot's five touch sensors. The robot also receives a direct feedback from a human supervisor, which here we will call as the 'external feedback' input and is given by the pressing of a numeric key.

For our example, the robot's touch sensors represent the set of unconditioned stimuli (US's) (analogous to the dog's sense of smell in Pavlov's experiment), while the external feedback represent a conditioned stimulus (CS) (analogous to the ringing of a bell in Pavlov's experiment). The external feedback initially has no influence in the robot's decision-making process. We expect that the robot will associate the US (touch sensor) and the CS (external feedback) stimuli after a conditioning-like procedure and learn to reduce speed in response to the external feedback, even if no collisions are detected by the touch sensors.

Fig. 3.6a depicts the ANN that controls the movements of the robot. The first five inputs of the ANN (u_1 to u_5), representing the US's, come from the robot's touch sensors (Fig. 3.6b) and accept only binary values, where one means that a

[1]Videos of the experiments in Webots are available at `http://carolrizzi.github.io/ASP`.



(a) Neural network that controls the robot. Inputs are divided into unconditioned ($\vec{u}$) and conditioned ($\vec{c}$) stimuli. Inputs belonging to vector $\vec{u}$ come from touch sensors, while input c_1 comes from the keyboard.

(b) Aerial view of robot's sensors disposition. Blue spheres represent the contact area of touch sensors.

Figure 3.6: Robot's controller setup for the single CS association experiment.

collision has been detected and zero means the opposite. Input c_1, representing the CS (which is initially an NS), comes from a numeric keyboard and may assume any value between one and nine (inclusive). The ANN's outputs provide the speeds for the robot's left and right wheels, respectively, in radians per second.

Results

Each test execution lasted 5 minutes and has been divided according to the phases and procedures depicted in Table 3.1. The sensitivity matrix has been configured so that the sensitivity of the external feedback stimulus is equally divided among the five touch sensors (i.e., 0.2 for each CS-US pair). This experiment is a good example of an environmental setup and task that do not require the rigorous adjustment of the sensitivity matrix. We discuss an example where the careful adjustment of CS-US sensitivity is needed in Section 3.4.2.

Fig. 3.7 shows the variation of the robot's speed over time for the first, third and fifth minutes of the experiment (before association, after association and after dissociation phases, respectively). Pink solid areas represent moments when the external feedback was presented to the robot, while hatched areas indicate moments when at least one of the robot's touch sensors detected collision.

Fig. 3.7a (before association) clearly shows that the robot reacts only to the sensors' indication of collision, reducing speed as needed to make a turn. As expected, the robot never reduces speed in response to the external feedback during this first phase.

Fig. 3.7b depicts the robot's speed over time after the association procedure. In

Table 3.1: Phases and procedures of the robot's conditioning in the single CS association experiment. Each test execution takes five minutes and is divided into five conditioning phases, each lasting one minute. Note that association rate (AR) equals 0 is analogous to executing the traditional ANN without ASP.

Minute	Phase	Procedure
1	Before Association	AR is set to 0.0. The key is pressed for 3 seconds every 10 seconds.
2	During Association	AR is set to 0.1. The key is pressed whenever a collision is detected.
3	After Association	The key is pressed for 3 seconds every 10 seconds and whenever a collision is detected.
4	During Dissociation	The key is never pressed.
5	After Dissociation	The key is pressed 3 seconds every 10 seconds if no collision is detected.

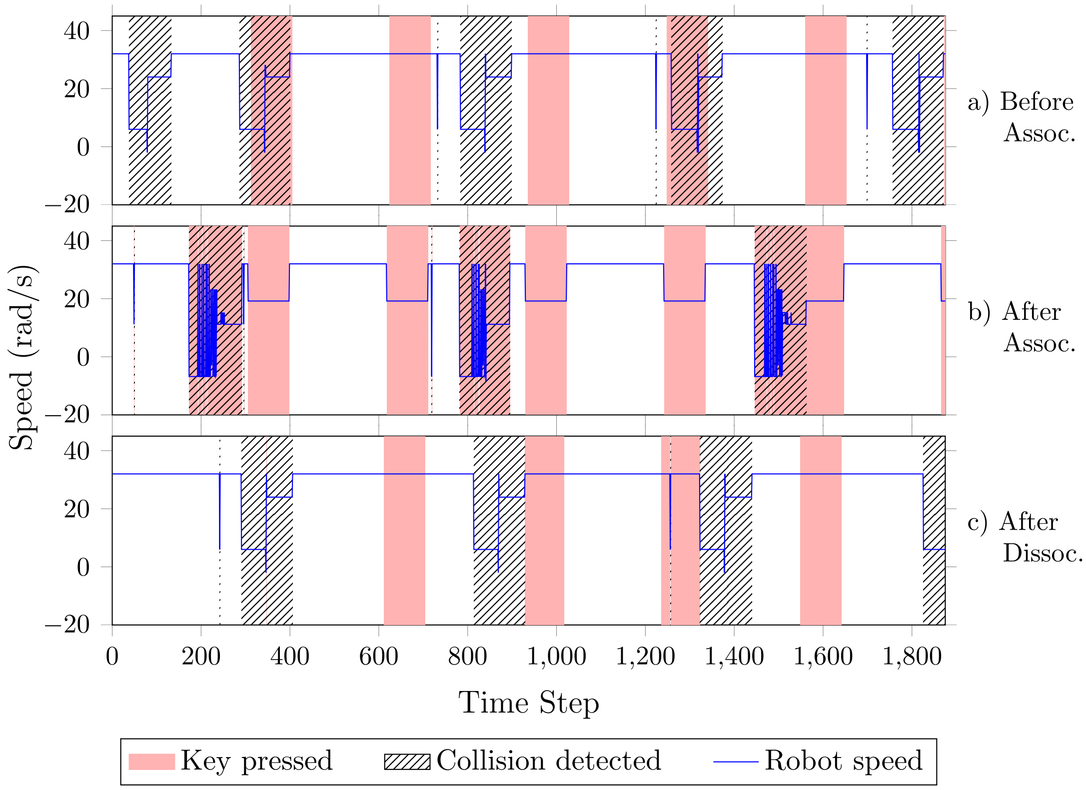


Figure 3.7: Robot's speed (in radian per second) over time for three of the conditioning phases: before association, after association and after dissociation. Pink solid regions show the moments when the external feedback was presented to the robot. Hatched regions indicate moments when one or more touch sensors detected a collision.

this phase, the robot reacts not only to collision, but also to the external feedback. Note that the robot's speed decreases in the beginning of all solid areas (when the external feedback is presented), and increases again in the end of solid areas (when the external feedback becomes absent), meaning that the robot slows down during the whole period in which the external feedback is presented.

Finally, Fig. 3.7c depicts the phase after the dissociation process, when the robot is expected to forget the learned association. Observe that the graphs from Fig. 3.7c and Fig. 3.7a are similar in the sense that the robot responds only to the detection of collision and ignores any external feedback. This is evidence that the robot successfully forgot the learned association between collision and the external feedback after the dissociation procedure took place.

Fig. 3.8 shows a comparison of the robot's speed according to the intensity of the external feedback, represented by the value of the numeric key activated by the human supervisor, where the higher the key number the stronger the external feedback. This graph takes into consideration the average speed of the robot at particular intensities of external feedback during the third minute of the experiment execution (i.e., after the association conditioning procedure took place). Fig. 3.8 shows a substantial difference in the speed for each tested external feedback intensity, which consistently decreases as the feedback intensity increases. This means that the intensity of the conditioned stimulus (CS) influences the intensity of the robot's conditioned response (CR).

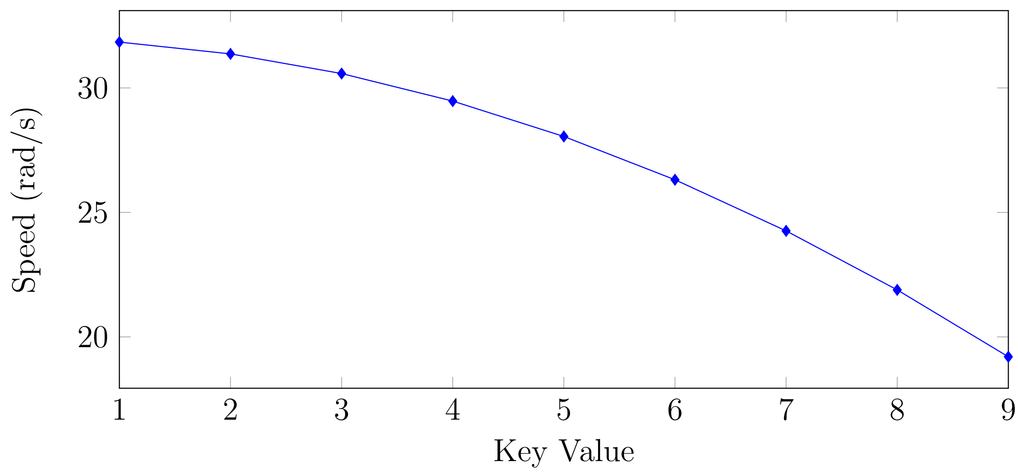


Figure 3.8: Robot's speed after the conditioning procedure, for each tested intensity of external feedback, depicted by the value of the pressed numeric key. The observed speeds are an average of the sum of the robot's wheels speed when the external feedback is presented.

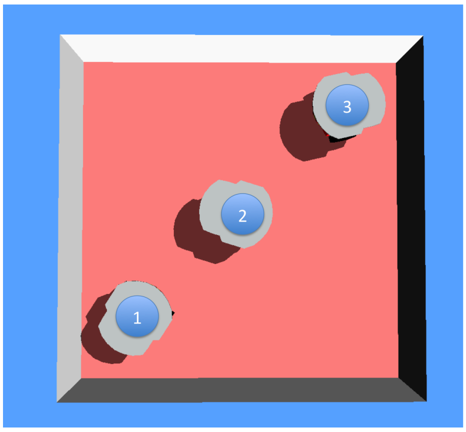
3.4.2 Experiment II – Multiple CS Association

The every-day learning and conditioning processes to which we are exposed during our lives usually involve much more complex associations than that studied in Pavlov's classical experiment. The process of learning how to handle objects is an example of a complex and natural conditioning, which is part of humans' growth process. For babies, the first attempts of catching an object are usually unsuccessful. It is probable that they will knock it over several times before successfully catching it. Each attempt provides the baby with vision and touch stimuli feedback. The repeated pairing of these stimuli while the baby continuously tries to catch the object is an example of conditioning, which eventually allows him or her to make an association between collision and proximity. After some attempts, babies start to understand that the closer their hands are to an object, the more likely they are to hit it. If the intention is to grab the object, then their hands must stop close enough to wrap it, but far enough to avoid knocking it over. Eventually, the baby finds a composition of approximation and movement speed that satisfactorily reaches the goal: grab the object.

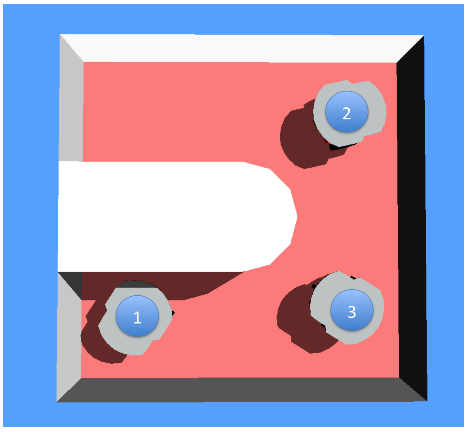
Unlike the simpler conditioning procedure conducted with Pavlov's dog, the baby conditioning in this example depends on the association of a series of complex stimuli, such as vision, movement speed and control of independent moving body parts (e.g., fingers and arms). All these neutral stimuli, which later become conditioned, must be associated with the feedback from the baby's hands touch. This results in a complex multi-stimulus association relationship, where the observable conditioned response depends on the association of a series of stimuli.

Experiment Setup

In this experiment, we investigate the outcome of the ASP mechanism in a multi-stimulus association scenario that is inspired by the setup of the experiments carried out by Timmis, Neal and Thorniley (2009), discussed in Section 2.2.1. In this experiment, the robot is equipped with both distance and touch sensors (as in Fig. 3.5b), but is trained to avoid obstacles based on its touch sensors information only. Environmental exploration will provide to the robot a natural conditioning because it is likely that at least one distance sensor will measure high proximity to an obstacle whenever a collision occurs. Therefore, no human direct feedback or intervention is required in this experiment. The robot is expected to gradually and autonomously associate collision with proximity during its operational cycle and eventually start using information from its distance sensors to avoid obstacles before touching them.



(a) First scenario: open space.



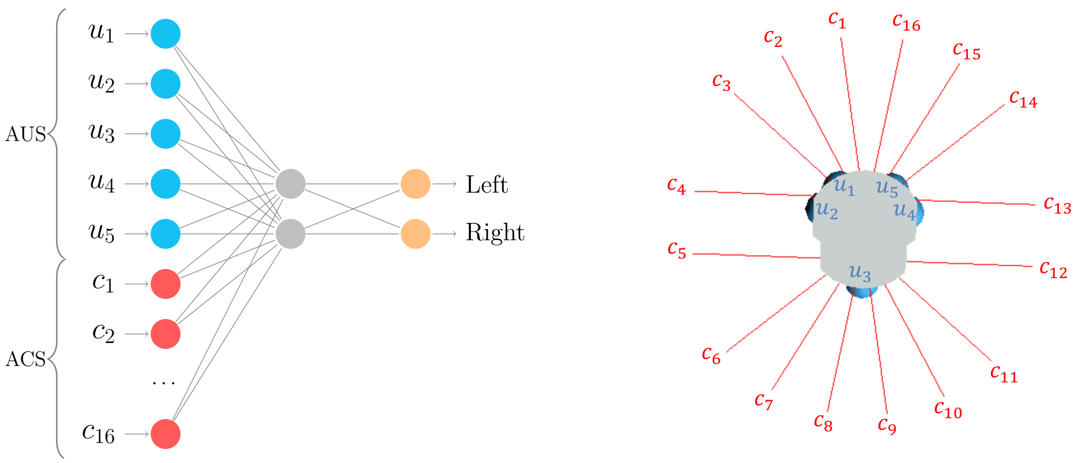
(b) Second scenario: corridor.

Figure 3.9: Aerial view of the robot's initial positions in the multiple CS association experiment.

To evaluate the robot's behaviour under different levels of environment complexity, we have executed the experiment in two scenarios, each with different obstacle arrangements. In each scenario, the robot has been initialized in three distinct positions/angles and evaluated for five association rates (ARs): 0, 0.001, 0.01, 0.1 and 1. AR $= 0$ is analogous to executing the traditional ANN without the ASP mechanism. Fig. 3.9 depicts the robot's initial positions in the two scenarios.

For this experiment, Webots has been configured to simulate measurement noise for the robot's 16 distance sensors. Noise errors are randomly generated and may deviate a distance measurement in up to $\pm 10\%$ of its correct value. We have executed each test 30 times to consider possible performance fluctuations derived from sensor noise, where each test execution lasted five minutes. Therefore, consider that all the results presented next (Fig. 3.12 and Fig. 3.13) shows the average values among the 30 simulations executed for each of the scenarios depicted in Fig. 3.9.

Fig. 3.10a depicts the architecture of the neural network that controls the robot's movements, whose inputs come from the sensors depicted in Fig. 3.10b. The first five inputs, representing the set of US's, come from the robot's touch sensors and assume only binary values, where one means that a collision has been detected and zero means the opposite. The last 16 inputs, representing the set of CS's (which are initially neutral), come from the robot's distance sensors and assume integer values ranging from 0 to 1024, where the higher the input value the closer the robot is to an obstacle. In our experiment, the robot's maximum detection range is 0.5 meters. Therefore, a distance-sensor measurement of zero means that this sensor is more than 0.5 meters away from detectable obstacles, while a measurement of 1024 means that it is touching an obstacle. The ANN's



(a) Artificial neural network that controls the robot. Inputs are divided into unconditioned (u_1 to u_5) and conditioned (c_1 to c_{16}) stimuli. Information of US inputs come from touch sensors, while information of CS inputs come from distance sensors.

(b) Aerial view of robot's sensors disposition. Red lines represent distance sensor rays and blue spheres represent the contact area of touch sensors.

Figure 3.10: Robot's controller setup for the multiple CS association experiment.

outputs provide the speeds for the left and right wheels, respectively, in radians per second.

By contrast to the experiment discussed in Section 3.4.1, this experiment requires the careful calibration of the sensitivity matrix. We may intuitively induce that sensors c_1, c_2 and c_3 in Fig. 3.10b should be associated with sensor u_1, because they have similar disposition in relation to the robot's body. Analogously, c_{14}, c_{15} and c_{16} should be associated with u_5; c_4 and c_5 should be associated with u_2; and so on. However, the knowledge of this additional stimuli relationship derived from the sensors disposition is not yet known by the ANN and will be ignored during environmental exploration if not provided in some way.

For instance, consider the example illustrated in Fig. 3.11. The left distance sensors are very close to the wall at the same time that the frontal touch sensors are active. As a consequence, the robot would associate with the frontal touch sensors (u_1 and u_5) not only the frontal distance sensors (c_1 to c_3 and c_{14} to c_{16}), but also the left distance sensors (c_4 to c_6). Because of the incorrect association, the robot would react as if it was frontally blocked whenever its left distance sensors indicated high proximity, going backwards when it should actually turn to the right.

As discussed in Section 3.3, the sensitivity matrix allows us to customize the degree of association between each pair of CS and US, which in our experiment are represented by the distance and touch sensors, respectively. The sensitivity

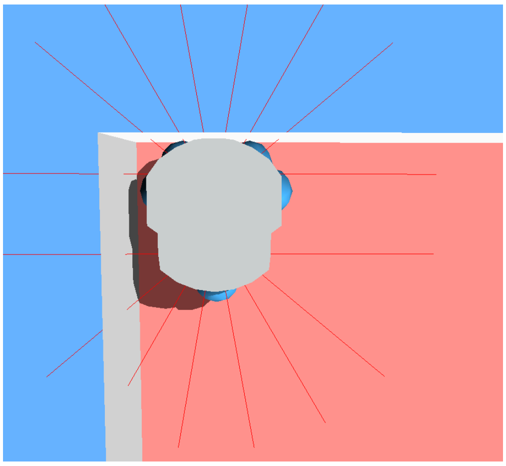


Figure 3.11: Situation in which the non-calibration of the sensitivity matrix would lead to wrong CS-US association.

matrix may be calibrated according to the developer's judgement, depending on the purpose of the associative learning, as well as the robot's design, environment and task. Here, we have based the sensitivity matrix on the sensors disposition, so that distance sensors are associated with the nearest touch sensor.

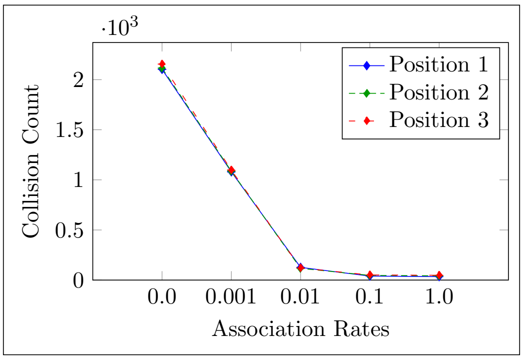
Table 3.2 depicts the sensitivity matrix used in this experiment (for the sake of readability, we omitted the value of zeroed cells). Some distance sensors, such as c_3, are close to more than one touch sensor and, therefore, have their sensitivity values divided between two touch sensors (u_1 and u_2, in this case). Other distance sensors, such as c_6 and c_{11}, are relatively far from all touch sensors, so they have no sensitivity mapping.

Table 3.2: Sensitivity matrix for the collision-proximity experiment.

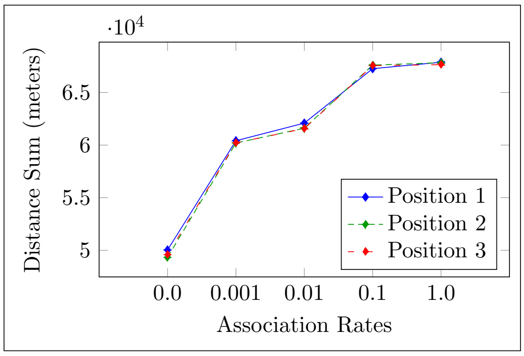
Touch Sensors	Distance Sensors															
	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}
u_1	0.2	0.4	0.4	-	-	-	-	-	-	-	-	-	-	-	-	-
u_2	-	-	0.4	0.4	0.2	-	-	-	-	-	-	-	-	-	-	-
u_3	-	-	-	-	-	-	0.1	0.4	0.4	0.1	-	-	-	-	-	-
u_4	-	-	-	-	-	-	-	-	-	-	-	0.2	0.4	0.4	-	-
u_5	-	-	-	-	-	-	-	-	-	-	-	-	-	0.4	0.4	0.2

Results - Part I

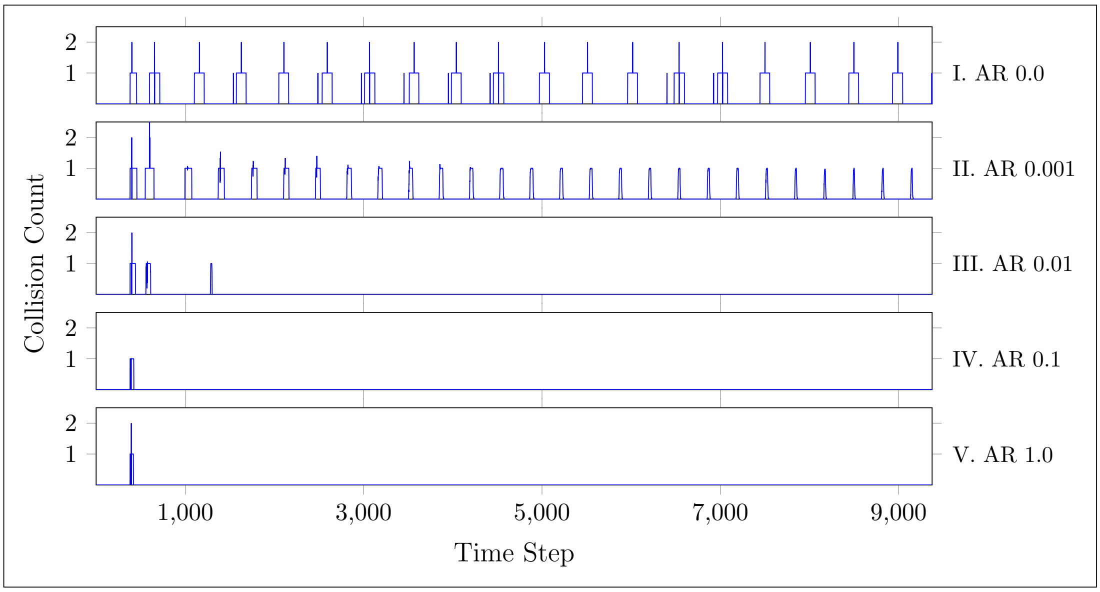
We have opted for starting this experiment with a fairly simple environment, as depicted in Fig. 3.9a, where the robot is unlikely to get trapped. By doing so, we can make a transparent evaluation of how the ASP mechanism affects the robot



(a) Collision Count per AR



(b) Distance Sum (in meters) per AR



(c) Count of detected collisions over time for robot starting in position 1.

Figure 3.12: Average results out of 30 simulation executions for each association rate tested in the scenario of Fig. 3.9a. Error bars in Fig. (a) and Fig. (b) show the standard deviation. Note that, in this case, error bars are hardly seen because the standard deviation in minimal.

behaviour, without interference of unrelated factors in the results, such as physical limitations of the robot.

The graph of Fig. 3.12a shows the collision counting for each tested association rate (AR) and for each of the three initial positions (Fig. 3.9a). Observe that the resulting number of collisions is highly similar among the different initial positions of the robot for all the tested ARs, which indicates that the starting position of the robot does not meaningfully influence the final outcome.

On the other hand, changing the AR drastically affects the incidence of collisions. In comparison to AR = 0 (i.e., when the ASP mechanism is disabled), the number of collisions has been reduced by about 50% when AR = 0.001, and more than 90% when the AR is larger than 0.001. This is a clear evidence that the robot has created a successful association between distance and touch sensors at runtime, leading to a meaningful reduction in the number of collisions during environmental exploration.

The graph of Fig. 3.12b depicts the summed distance measurements for each tested AR and for each initial position of the robot. This summation of distance measurements takes into consideration all the distances between the robot and any obstacle detected by the 16 distance sensors along complete simulation executions. Fig. 3.12b clearly shows that the overall distance from obstacles increases as the AR increases. This indicates that the larger the AR, the earlier the robot learns to use the distance sensors to stay away from obstacles.

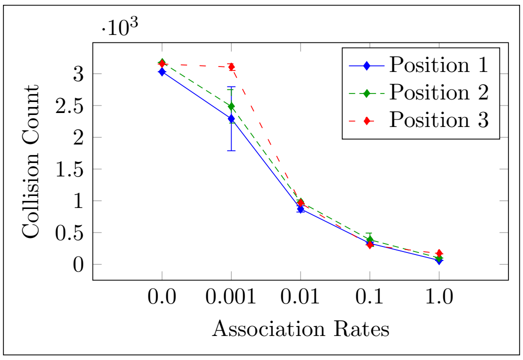
Finally, Fig. 3.12c shows the collision count over time for each AR tested in simulations where the robot started from position 1. There is a clear difference in collision incidence between graphs I, II, III and IV (ARs 0.0, 0.001, 0.01 and 0.1, respectively). However, the difference between graphs IV and V is minimal. This is because, for this particular environment, AR = 0.1 allows the robot to successfully create an collision-distance association after just one conditioning step (i.e., one collision). Therefore, increasing the AR above 0.1 would not considerably change the robot's behaviour or adaptive performance. This effect can also be observed in Fig. 3.12a and Fig. 3.12b, where the difference in total collisions and obstacle distance when AR = 0.1 and when AR = 1 is considerably small if compared with the other tested values of AR.

Results - Part II

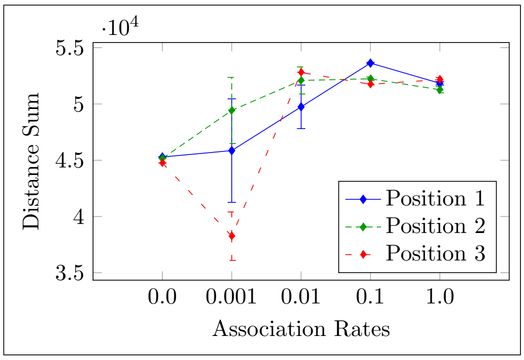
The second part of this experiment has been executed in the scenario of Fig. 3.9b, which consists of a curved corridor that is relatively narrow for the robot to make turns and has two dead ends. In this second scenario, we desire to observe the improvement of the robot's adaptive behaviour under more complex object-avoidance situations. Fig. 3.13a shows the collision counting for each initial position of the robot and tested association rate (AR).

By contrast to the previous experiment, in this scenario the starting position of the robot has yielded slightly different results. The number of collisions when the robot started in position 3 is almost the same for AR = 0.0 and AR = 0.001, which did not occur for the other initial positions. This is because, when AR = 0.001, the robot got trapped in a corner where it continued bumping until the end of the simulation. This trapping situation of the robot can be observed on the videos of the experiment[2].

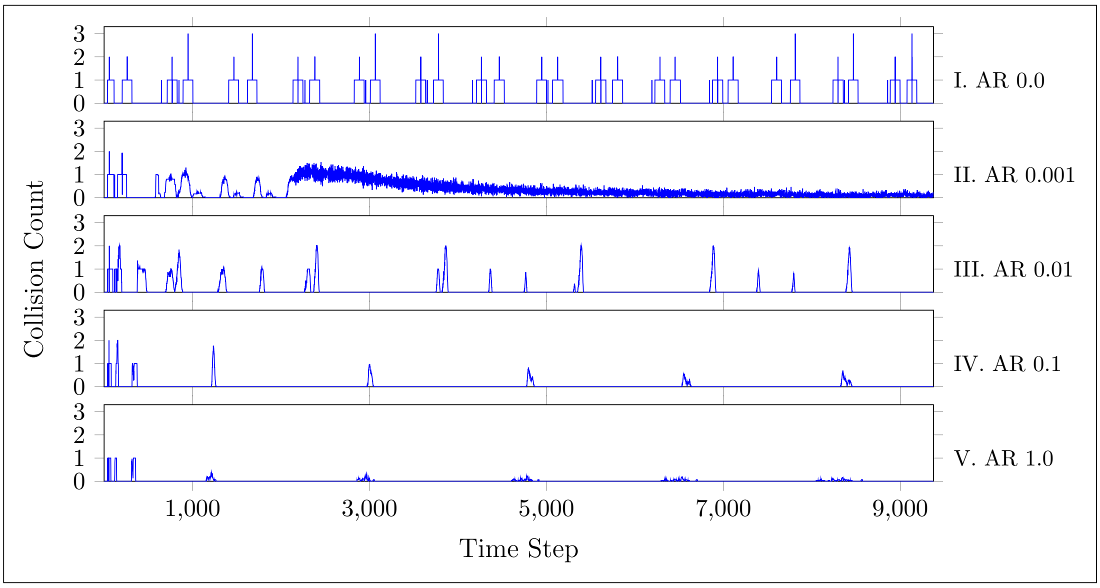
[2]Videos of the experiments in Webots are available at `http://carolrizzi.github.io/ASP`



(a) Collision Count per AR



(b) Distance Sum (in meters) per AR



(c) Count of detected collisions over time for robot starting in position 3.

Figure 3.13: Average results out of 30 simulation executions for each association rate tested in the scenario of Fig. 3.9b. Error bars in Fig. (a) and Fig. (b) show the standard deviation.

The trapping occurred due to limitations of the robot's touch-sensor's disposition in combination with the specific angle that it reached the corner in the specific case where AR = 0.001. The exact moment when the robot got trapped can be observed in Fig. 3.13c, which shows the collision detection over time for simulations where the robot started from position 3. Note that, after the time step 2000 in graph II (AR 0.001), the number of collisions increases and never reaches zero again, indicating that the robot was constantly in collision from that moment on. A similar situation happened for simulations where the robot started from the other initial positions (1 and 2). However, in these cases, the trapping situation occurred only in a few of the 30 simulation executions, this is the reason why the collisions counting presents high standard deviation for positions 1 and 2 when

AR = 0.001 in Fig. 3.13a.

Despite the trapping situation, results for the other ARs (0.01, 0.1 and 1.0) in Fig. 3.13a are relatively stable, presenting a consistent decrease in number of collisions, with low standard deviation. In addition, for these AR values, the graph's lines depicting each of the initial positions are considerably close to each other, indicating that the starting position of the robot did not meaningfully influence the experiment outcome.

The discussed trapping of the robot can also be observed in Fig. 3.13b, which shows the sum of all distances measured by the 16 distance sensors of the robot for each initial position and tested ARs. As expected, because of the trapping situation, the distance sensors detected a minimal distance from obstacles for tests with AR = 0.001, indicating that the robot was predominantly close to the walls during these simulations. The high standard deviation of results when ARs = 0.001 and ARs = 0.1 demonstrates that, in these cases, the robot's exploratory performance varied from one simulation execution to another. Possibly, the robot got trapped for longer in some particular simulation executions only.

Unlike the graph of Fig. 3.13a, which shows a growing slope (indicating that the distance from the walls increase as the AR increases), Fig. 3.13b shows a different result for AR values above 0.001. More interestingly, Fig. 3.13b shows that proximity to obstacles is increasing for ARs above 0.001, but the number of collisions is actually decreasing for the same AR values, as seen in Fig. 3.13a. It follows that, in this scenario, the robot is spending more time near the walls when under high AR values, but is not touching them.

In fact, because the corridor in Fig. 3.9b is relatively narrow for the robot to make 180° turns, it needs to make slower and more 'careful' movements to avoid touching the wall when turning back in dead ends. As a consequence, the robot spends more time close to the wall making the turn, but not necessarily touching it. In Fig. 3.13b, this phenomenon is more noticeable for ARs = 0.1 and ARs = 1.0 in position 1 and for ARs = 0.01 and ARs = 0.1 in position 3.

The robot's behavioural changes were beyond our expectations. Because of locomotion difficulty in the dead ends of the narrow corridor, the robot had to move more 'carefully' in order to make a turn without touching the walls. As a consequence, the more we increased the AR, the more time the robot spent making the turn in the dead ends, and the less it collided. This 'cautiousness' is a positive collateral-effect that was neither deliberately designed, nor predicted.

The explored area is another unexpected and positive collateral-effect. When running without ASP, the robot's awareness of space was limited to touch feedback, so it could not perceive alternative (and perhaps better) paths to avoid an obstacle.

As a consequence, the robot kept doing laps in a small space. By contrast, when using ASP, the increased vision-range of the distance sensors improved the robot's space-awareness, which doubled the robot's explored area.

3.5 Final Considerations

This chapter presented a computational model of artificial synaptic plasticity (ASP) for simulating cued fear conditioning in the amygdala. This model is based on the NMDAR-dependent LTP phenomenon that takes place in amygdala regions and is essential for fear acquisition and extinction. The model relies on a modification to the classical feedforward neural network proposed by us in Rizzi Raymundo and Johnson (2014), where the first-layer weights of the ANN are updated according to the activity coincidence of the input neurons. The original ASP model as proposed in Rizzi Raymundo and Johnson (2014), which simulates general conditioned learning and association, has been later adapted to provide specifically fear learning in order to meet SAFEL's needs.

Experiments with a virtually simulated Pioneer robot demonstrated that the proposed ASP mechanism successfully generates online associative learning by means of a cued fear-conditioning procedure. The robot autonomously learned during environmental exploration to use input information added post-training, in a *ad-hoc* manner. Stimuli association with ASP has been tested at different complexity levels and has been shown to successfully improve the robot's object-avoidance capabilities in all the tested situations. Additionally, both association and dissociation phenomena have been demonstrated in the simulations.

Interesting behavioural modifications of the robot have been observed, which resembles the expression of behaviours such as 'fear' and 'carefulness'. In the experiment of Section 3.4.2, the robot has been trained to avoid obstacles whenever possible using information from its touch sensors. After associating the distance sensor inputs with the touch sensor inputs, the robot became able to express the same obstacle-avoidance behaviour in response to the distance sensor as well. However, different from the touch sensors, whose inputs are binary, distance sensor's inputs are continuous values and, therefore, provide the robot with a more complete information about the environment. As a consequence, we could observe behaviours from the robot in response to the distance sensors inputs that were not expressed when the robot was reacting to the touch sensors only.

For instance, the thorough information from the distance sensors allowed the robot to understand and adjust its speed according to proximity information, which was not possible with the touch sensors only because the robot would only become aware of potential collisions after these have already happened. The speed adjustment learned by the robot led it to express behaviours such as 'carefulness' and 'fear' of collision, which were not previously observed. Interestingly, the intensity of these behaviours could be regulated by adjusting the association rate (AR), so that the higher the AR, the more 'careful' the robot would behave.

Interestingly, the robot's performance in carrying out the given task (i.e., exploring the environment without colliding) was not solely improved by increasing the AR, but instead by finding a balance in the level of 'carefulness' in the robot's resulting behaviour. In real-life daily tasks, to much carefulness can be as detrimental as the total lack of it. Hand-writing a letter with no care for calligraphy may turn it unreadable, thus failing the goal of that task, which is communication. On the other hand, dedicating to much care to calligraphy may take the writer much longer than necessary to finish the letter, which is also detrimental. This is analogous for the experiment discussed in Section 3.4.2. When the AR was too low, the robot would frequently bump into the walls, failing the goal of avoiding collisions. On the other hand, when the AR was too high, the robot would commit almost no collisions but take much longer to complete a 180° turn. Consequently, the robot would have less time to effectively explore the environment. This phenomenon can be clearly observed in the videos of the experiment[3].

For decades, the Hebbian rule has been an inspiration for learning algorithms (Balkenius and Morén 1998; Miller, Barnet and Grahame 1995; Timmis, Neal and Thorniley 2009). In its simplest form, the classical Hebbian learning algorithm consists of modifying the connection between two computing units (e.g., neurons) by an amount proportional to the product of the activation of those units (Sanger 1989), which is mathematically expressed as in Eq. 3.18:

$$W(t+1) = W(t) + \gamma(\vec{y}(t)\ \vec{x}(t)^T) \tag{3.18}$$

where γ determines the rate of change of the weights, $\vec{x}$ is the activation of the input nodes, W is the weight matrix and, consequently, $\vec{y} = W\vec{x}$ is the activation of the subsequent layer of neurons. The ASP mechanism proposed here goes beyond the classical Hebbian algorithm (as defined in Eq. 3.18) in several aspects:

- The ASP mechanism allows the definition of the degree of relationship between pairs of conditioned stimulus (CS) and unconditioned stimulus (US) by means of the sensitivity matrix. This gives the designer of the artificial agent freedom to model possible conditioned associations according to the

[3]Videos of the experiments presented in this chapter are available at `http://carolrizzi.github.io/ASP`

requirements or particular features of the agent, as well as of the agent's environment and task. A relationship based on the input values of the ANN, such as that established through the sensitivity matrix of ASP, could not be easily implemented with the formula of Eq. 3.18 because the classical Hebbian algorithm updates weights based on the relationship between $\vec{x}$ and $\vec{y}$ rather then between $\vec{x}$ and itself.

- The classical Hebbian algorithm updates the weights based on the product $\vec{y}(t)\,\vec{x}(t)^T$. This method considers that every CS will behave equally, where the lowest possible value is a baseline for what is considered to be the standard behaviour of that CS and the highest possible value represents the occurrence of an uncommon event, which may entail a threat to the agent or another reason for associative learning to take place. This approach is incompatible with real world situations because different CS's behave in different manners, which are usually unknown before environmental exploration. As discussed in Section 3.3 (with the example of a robot living under high/low levels of noise pollution), a particular CS could assume medium to high values in normal situations and low values whenever something uncommon happens. Instead of using the lowest possible value as a baseline, the ASP mechanism proposed here uses the average values of that CS over time as the baseline, thus taking into consideration the particularities of each CS and, consequently, being more suitable for real world applications.

- The ASP mechanism also takes into consideration the effect of environmental changes in the behaviour of CS's. For instance, moving an agent from one environment to another may have a meaningful effect in the average values of a particular CS over time. The ASP mechanism takes this factor into consideration by gradually forgetting old average values and giving priority to values that represent the most recent environmental conditions of the agent.

Although the ASP mechanism introduced in this chapter can be considered an emotional cognitive model by itself, it is important to keep in mind that it does not represent SAFEL as a whole and, in fact, constitutes just one of the three modules of SAFEL. The ASP mechanism described here represents the elementary foundation of the SAFEL model, over which more complex knowledge and memory is built on by the higher modules of SAFEL. The ASP of the AM is the threat-detection machinery of SAFEL, which not only monitors the occurrence of foreknown threats (the US's), but also learns and memorizes new primary threats in the environment (the CS's).

Chapter 4

Hippocampus Module

The *hippocampus* consists of two curved regions of the brain, notably shaped like a seahorse, located in the medial temporal lobe (Fig. 4.1). Considerable evidence indicates that the hippocampus is essential for the formation of episodic memory and the processing of context, playing an important role in the phenomenon of contextual fear conditioning (Phillips and LeDoux 1992; LeDoux 1999; Rudy, Huff and Matus-Amat 2004; Fortin, Agster and Eichenbaum 2002).

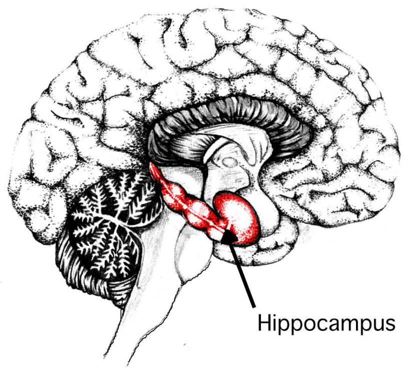


Figure 4.1: Hippocampus region in the brain.

The following sections explore the phenomenon of contextual fear conditioning and its underlying neural mechanisms, especially in the hippocampal regions of the brain (Section 4.1). Section 4.2 and Section 4.3 discuss, respectively, the basic concepts and technologies related to an area of artificial intelligence known as *knowledge representation*, which is inspired by cognition theories on context formation and representation in the brain. Finally, we present the Hippocampus Module (HM) model in Section 4.4, which is based on concepts of knowledge representation.

This chapter partially contributes to answering the first and second of our three research questions formulated in Section 1.4 by:

1. addressing the last two requirements of a situation-aware intelligence (Section 1.3.1) in combination with the last requirement of an emotional intelligence (Section 1.3.2); and

2. presenting a module of SAFEL capable of manipulating stimuli information so to create contextual information. This module is designed and implemented using a rule-based platform for situation management, which is the second approach of a hybrid model consisting of three distinct approaches.

The discussion of experiments related to the HM of SAFEL is addressed in Chapter 5 because such experiments also depend on the working of the Working Memory Module. Therefore, examples of practical robotics applications using the HM are not presented in this chapter and should, instead, be sought in Chapter 5.

4.1 Biological Background

This chapter introduces the biological phenomena and related brain functions that inspired the HM design. Section 4.1.1 introduces the contextual fear conditioning phenomenon, which greatly depends on the proper functioning of hippocampal areas of the brain. Next, in Section 4.1.2, we briefly discuss the functions of the hippocampus that support the acquisition and expression of contextual fear learning and their underlying neural mechanisms.

4.1.1 Contextual Fear Conditioning

In Section 3.1, we discussed how the foot-shock experiment with mice demonstrated the phenomenon of cued fear conditioning. A mouse is placed into an apparatus and is repeatedly stimulated by auditory cues paired with an electrical shock to its feet. Such procedure then conditions the mouse to freeze in response to the auditory cue, even in the absence of an electrical shock.

Nevertheless, the auditory cue is not the only stimulus observed to induce the mouse's fear expression in this experiment. Researchers conducting this experiment observed that if they removed the mouse from the cage where the cued fear conditioning first took place and, after some time, returned it to that same cage, the mouse would also present the freezing response, even in the absence of the auditory cue or the electrical shock (Phillips and LeDoux 1992). This indicates that the mouse has also associated the aversive unconditioned stimulus (US) with the background context, which in this case was the cage where the shock was induced in the first place. The phenomenon of expressing defensive responses in the presence of a specific combination of stimuli (e.g., a situation or place) under which an aversive US has been previously induced is known as *contextual fear conditioning* (Phillips and LeDoux 1992).

In the neuroscience literature, the term *context* is commonly used to refer to any stimuli other than the main conditioned stimulus (CS) cue (e.g., the tone in the experiment with mice) that is static and continuously present in the environment where the US occurs. Stimuli with this characteristic are known as *background contextual stimuli*. However, a contextual stimulus can also be associated with the aversive US without the parallel pairing of a cued CS. In this case, the contextual stimulus is said to be a *foreground contextual stimulus* (Phillips and LeDoux 1994).

Although both types of conditioning (cued and contextual) lead to the same fear responses, their perception and processing mechanisms in the brain are notably distinct. In cued fear conditioning, the CS is restricted to a single stimulus that belongs to a specific sensory modality (smell, touch, taste, hearing or vision), whereas in contextual fear conditioning, the CS is composed of a collection of stimuli, which may belong to different sensory modalities (Phillips and LeDoux 1992). This set of stimuli is bound into a unitary representation of context that depicts not the stimuli per se, but how they correlate in time, space and intensity (LeDoux 1999; Eichenbaum 2004).

4.1.2 Context in the Brain

In the hippocampus, we begin to leave the purely perceptual reasoning about the world and enter the conceptual domain of the brain. Sensory information is put together in the hippocampus to form a unitary representation of the current state of affairs. Unlike information processed in the amygdala, representations formed in the hippocampus are not just visual, auditory or olfactory, but all of these at once, and includes the way these sensations relate to each other in intensity, space and temporal order (LeDoux 1999).

Hippocampal regions are believed to play an important role in the brain's contextual processing (Phillips and LeDoux 1992; Eichenbaum 2004). Eichenbaum (2004) provides a thorough review of the underlying cognitive mechanisms involved in the hippocampus' contextual processing and *episodic memory* (our capacity of recollecting personal experiences). According to Eichenbaum (2004), there are three elemental cognitive processes mediated in the hippocampus, from which two represent the inspiration for the design of the HM. These are the *associative representation* and the *sequential organisation*.

Associative representation regards to the association between stimuli that compose each discrete event in an experienced episode. Evidence from hippocampal lesions in humans and animals suggests the existence of learning mechanisms in the hippocampus that are essential for binding into an integrated fragment of memory the array of features associated with an event (Davachi and Wagner 2002). In Section 3.1, we have discussed the mechanisms behind the phenomenon of NMDA-dependent LTP and how it is involved in amygdala functions. Although the character of information being processed in the hippocampus is different from that being processed in the amygdala, the same long-term potentiation (LTP) phenomenon is believed to underlie both kinds of learning (Martin, Grimwood and Morris 2000). Some hippocampal functions supported by LTP are the multi-stimulus association and the *pattern completion*, which occurs when the presence of one or more stimuli participating in an episodic memory leads to the firing of neurons that retrieve the entire episode. Research has also shown that the hippocampus is activated during multi-stimulus processing, especially when subjects are required to create link between these stimuli by means of systematic comparison (Davachi and Wagner 2002).

Sequential organisation regards to the temporal organisation of events composing an experienced episode. According to Eichenbaum (2004), episodic memories consist not only of the particular information one is attempting to recall but also of the 'experience of events that precede and follow'. A number of studies have demonstrated the relevance of hippocampus' temporal representations in promoting intelligent behaviour and survival. Honey, Watt and Good (1998) has shown that hippocampal lesions disturb the normal orienting response of animals when pairs of stimuli are presented in an order different from the order presented in the training phase. Also, according to Eichenbaum (2004), the firing of hippocampal neurons related to the processing of event sequences have been shown to be active during several learning protocols, including classical conditioning.

As mentioned in Chapter 3, considerable evidence indicates the amygdala as the main brain region involved in fear learning and memory (LeDoux 2003, 1999; Phillips and LeDoux 1992; Herry and Johansen 2014). However, the hippocampus is also believed to be essential for contextual fear conditioning. Research has demonstrated that lesions of the amygdala interfere with the acquisition and expression of cued and contextual fear learning, while lesions to the hippocampus interfere with contextual fear learning only (Phillips and LeDoux 1992; Rudy, Barrientos and O'Reilly 2002).

LeDoux (1995) argues that the hippocampus may function in the process of contextual fear conditioning as a kind of higher-order sensory structure that integrates its own contextual information with the emotional meaning processed in the amygdala. Learning this kind of relationship might allow the individual to distinguish in which situations a given CS may represent a threat. For instance, the appearance of a snake during a walk in the woods may represent a threat and should trigger fight-or-flight bodily responses, while a snake's appearance in the zoo should not be a reason for concern.

According to LeDoux (1999), the amygdala and hippocampal systems work in parallel, forming what LeDoux calls, respectively, as the *emotional memory* and the *memory of emotion*. When a person remembers a past traumatic situation, in addition to the state of affairs, the hippocampus will also coldly remind this person that he/she was afraid at that time, providing an unemotional *memory of emotion*. The amygdala, in turn, will trigger bodily and brain responses (muscles straining, increased heart rate, hormone release, etc.) that allow him/her to re-experience the fear felt during that trauma, thus providing an *emotional memory* of the episode. Exposure to stimuli that were present during the trauma activates both the amygdala and hippocampal systems, which work in parallel to retrieve emotional and contextual memory about the event, respectively. Because these two memories are simultaneously recovered in response to the same stimuli, they are experienced as if they were one single memory.

Different from the model of the AM presented in Chapter 3, which is inspired by the neural mechanisms behind the LTP and LTD phenomena, the HM does not attempt to directly model the underlying physical and chemical phenomena of hippocampal neurons. Instead, we have modelled and implemented the HM at a higher level of abstraction that focuses on the functionalities of the hippocampus, especially associative representation and sequential organisation of events. A considerably large ANN would be required to simulate the complex temporal and multi-stimulus information handled in the hippocampus. For this reason, we decided to model the HM based on psychological conceptualizations of humans' reasoning and context representation. These concepts are presented next, in Section 4.2

4.2 Context and Knowledge Representation – Basic Concepts

As discussed in Section 4.1, the hippocampus is the main brain region involved in the processing of contextual information. Analogously, the Hippocampus Module

(HM) of SAFEL represents the centre of contextual processing in the proposed model. The HM is responsible for collecting individual stimuli and binding them into pieces of information that can describe the robot's state of affairs over time.

To perform this task, the HM is based on techniques used by situation-aware expert systems. Expert systems are those that emulate the decision-making ability of a human expert in a domain-specific problem (Jackson 1998; Sasikumar et al. 2007), and by doing so they require a protocol that specifies how to represent knowledge and reproduce human reasoning. The most popular theories of the human reasoning and behaviour, such as SOAR (Rosenbloom et al. 1991) and ACT* (Anderson 1996), describe the human conscious cognitive process of decision-making as a combination of declarative knowledge (facts about the world), procedural knowledge (facts about how to perform certain actions), and episodic knowledge (rules defining when to execute certain actions) learned in past experiences.

Therefore, to simulate the decision-making process of a human expert, expert systems need a methodology able to recognise and process rules and facts, which involves the creation of declarative representations of relevant knowledge. However, traditional programming languages (Java, C++, C#, etc.) are designed for solving procedural problems, which instead involves the declaration of a sequence of routines. As consequence, modelling the human representation of knowledge by means of procedural languages requires considerable effort, in most cases leading to the so-called 'spaghetti code' (several nested if-else statements).

For this reason, expert systems are commonly developed by means of *rule-based techniques* using *symbolic computation*, i.e., non-numeric computations in which symbols and symbol structures are constructed so to represent concepts and relationships between them (Jackson 1998). Rule-based approaches allow developers to define the system's reasoning at a higher level of abstraction that is close to well-known cognition theories over the human reasoning and knowledge representation. In addition, rule-based approaches also provide a clear separation between the system's logic and other parts of the system (e.g., data manipulation), which facilitates the system's maintenance and readability.

This section presents the main basic concepts behind knowledge representation and situation-aware expert systems, including rule-based programming (Section 4.2.1), event processing (Section 4.2.2), context awareness (Section 4.2.3) and situation awareness (Section 4.2.4).

4.2.1 Rule-Based Programming

Knowledge representation is, according to Jackson (1998), a substantial area of study in the borderline between artificial intelligence and cognitive science. It is concerned with computationally representing and storing information about the world in a manner analogous to the human brain. *Knowledge representation languages* are programming languages that focus on describing objects and ideas instead of defining sequences of instructions or storing simple data elements (Jackson 1998). *Rule-based languages* are a type of knowledge representation language that 'encode empirical associations between patterns of data presented to the system and actions that the system should perform as a consequence' (Jackson 1998).

Rule-based languages consist of a set of rules, which can be repeatedly applied to a set of facts. *Facts* are data representing entities observed in the real world and are classified by their types. Examples of fact type are 'person', 'tree', 'cat', whose fact instances could be, for example, 'John', 'the tree of John's house' and 'Garfield', respectively. Facts may have properties. Some examples of properties for the fact type 'person' are 'name', 'age' and 'temperature', which could assume values such as 'John', '28' and '37°C', respectively.

On the other hand, *rules* are heuristics that define a collection of actions to be executed in a given circumstance, which is conditioned to a collection of fact instances and their properties. An example of a rule is 'when a person enters the building, then open the lift door'. Since 'John' is an instance of the fact 'person', such rule would be satisfied and have its actions executed (i.e., open the lift door), if John happens to enter the building.

Code 4.1 exemplifies a generic rule. Rules are basically two-part structures (*if-then* or *when-then* statements) that reason over knowledge representation using first order logic. The *if* or *when* part of the rule, known as the *head* or the *left-hand side* (LHS), consists of a condition that can be a single expression (i.e., evaluates only one fact type in order to apply the rule) or a set of expressions. Each expression is called a *pattern*. Patterns are joined to each other by logical connectors (i.e., *and*, *or*). A rule is said to be *matched* if all of its patterns are satisfied. The *then* part of the rule, known as the *body* or the *right-hand side* (RHS), is composed of a set of operations, which are executed if the rule is matched.

Code 4.1: Example of generic rule.

```
if pattern_1 and pattern_2 and ... pattern_n      /*left-hand side*/
then action_1 and action_2 and ... action_m       /*right-hand side*/
```

A typical rule-based system (Jackson 1998; Sasikumar et al. 2007) consists of three components, which are:

- *Working memory*: is the space in memory reserved to the facts known about the domain. In other words, the working memory usually contains information about the particular instance of the problem being addressed. Note that the working memory described here is part of the rule-based paradigm and is not to be confused with the brain's working memory or with the working memory module of SAFEL, both of which are discussed in Chapter 5. For disambiguation purposes, here we will call it as the *rule-based working memory* (RBWM).

- *Rule base*: also known as the *knowledge base*, is the set of rules modelling the knowledge about the domain.

- *Inference engine*: also known as the *rule engine*, is the component that evaluates fact instances against rules' patterns, which is a process known as *pattern matching*. When one or more facts satisfy a rule's condition, the inference engine executes the actions defined in that rule's RHS. The inference engine also carries out a process called *conflict resolution*, which dictates the execution order of conflicting rules, i.e., rules that have been simultaneously matched by the same set of facts.

A *fact insertion* is the act of inserting information about a fact instance in the RBWM and, consequently, notifying the rule engine about the existence of the respective fact. After being inserted, the fact instance is matched by the rule engine against all rules in the rule base in order to find which rules' patterns are satisfied by that fact instance. After a fact instance is inserted in the RBWM, it can be updated or retracted.

A *fact update* is the act of warning the rule engine about changes in the information of that fact's properties. For example, consider the fact type 'person', which has properties such as 'name', 'age' and 'temperature'. An alteration to a person's temperature represents an update to the properties of the fact instance representing that particular person. To notify the rule engine about the alteration in that person's temperature, the respective fact instance must be updated in the RBWM. On the other hand, a *fact retraction* is the act of removing the information about a fact instance from the RBWM, so the respective fact instance can no longer be matched against rules.

When the actions of a rule are executed, that rule is said to have been *fired* (or *activated*). Rules may generate new facts when fired. In this case, the newly generated facts may lead to subsequent pattern matches, resulting in a cascade effect that takes place until there are no facts in the RBWM able to activate any of the rules in the rule base.

4.2.2 Complex Event Processing

Some rule-based systems are able to understand and manage *event notifications*, which represent the flow of information that signals changes of state in the system's domain (Cugola and Margara 2012). Such systems make use of a data processing model known as *complex event processing* (CEP), which filters and combines events' notifications to generate higher-level events known as *composite events*. CEP engines usually provide tools for defining and managing different aspects of event abstraction, correlation and hierarchy.

Events occurring in the real world are represented in a CEP system as *event instances* and are categorised by *event types*. For instance, consider a fire detection system that receives temperature information from sensors distributed in a building. The temperature notification from one of the sensors in the building's entrance hall is an *event instance* of the 'temperature' *event type*.

Composite events are events whose existence depends on the notification of other events' occurrence. For instance, the instantiation of an event of type 'fire' could depend on a series of constraints on others events occurrence, such as a temperature greater than 60°C being reported by two different sensors located within an area smaller than 30 m^2, where these notifications occurred within 10 seconds of each other.

Observe in this example that events correlation is not only evaluated in terms of their standard properties (such as the temperature value and sensor location), but also in terms of their temporal properties (such as the fact that temperature notifications occurred within 10 seconds of each other). The computational operations used to correlate events based on their temporal aspects are known as *temporal operations*. Some common examples of temporal operators are *before* (when an event occurs before another one), *during* (when an event occurs during the occurrence of another one) and *starts* (when two events start to occur at the same time but not necessarily ceases to occur at the same time). We will return to the subject of temporal operators in Section 4.3, giving more examples and discussing their usage for comparing the temporal properties of events.

4.2.3 Context awareness

Context is a common word in people's everyday dialogues. The Oxford Dictionaries (2017a) defines it as 'The circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood.' Therefore, being aware of context is the same as being aware of such circumstances in relation to something or someone.

Dey (2001) provides one of the most cited definitions of context for computational applications in the literature. According to Dey (2001):

> Context is any information that can be used to characterise the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

In other words, *context* is any information about the observed entity that is relevant to the purpose of an application. A *context-aware application* is any application able to observe the context of an entity and capture important information about that context in order to autonomously perform inferences, decisions and actions. The gathered information about context is known as *context information* and can be obtained explicitly, by means of users' input, or implicitly, by means of sensors' input.

Context information can be *primitive*, i.e., a discrete piece of information such as the user's GPS location, or *composite*, which is a combination of other pieces of context information (either primitive or composite) and their relationship. For instance, consider an application that gives quick suggestions of nearby places to eat. To make a suggestion, the application must be aware of a composite contextual information, which depends on several other pieces of context data. Firstly, the application will need context information about the time of the day to ensure that it is appropriate to suggest lunch or dinner at the current time. Another essential context information in this scenario is the user's location. Finally, the application will also need information about nearby restaurants' locations, food style and users' rating, which may be gathered from the internet.

Observe in this example that a composite context consists not only of other pieces of context information but also of their relationships. In this specific example, the list of suggested restaurants (which is a context information by itself) is constructed based on the user's location (which is another context information) and a constraint (e.g., these restaurants must be located within 200 meters of the user's location). This relationship is also part of the composite context detected by the application.

According to Prekop and Burnett (2003), there are two dimensions of context: external and internal. The *external dimension* of context, which is the main focus of most context-aware applications, capture elements of the physical environment such as location, proximity, temperature, time and lighting levels. On the other hand, the *internal dimension* of context attempts to capture a more cognitive aspect of context, such as the user's goals, tasks, business processes, personal events, communication and emotional and physical states. Both dimensions are important and meaningfully contribute to capturing the unique pattern of activities performed by an individual that defines his/her context.

In computing, the concept of context awareness is mostly used in *ubiquitous computing* (also known as *pervasive computing*), which is the area of research concerned with enhancing technology use by embedding computers in our everyday movements and interactions with the environment, both physical and social (Hansmann et al. 2013). Ubiquitous computing usually involves the use of mobile devices present in our daily activities to access information about the users and their environment so to provide them with useful applications. Hence, being aware of users' context is essential in ubiquitous computing.

In addition to mobile computing, another popular application for ubiquitous computing is its integration with expert systems. Conventional expert systems depend on users' explicit input to perform their tasks. However, the integration between ubiquitous computing and expert systems allows the creation of proactive systems capable of detecting and processing users' contextual data to automatically generate domain-specific solutions (Kwon, Yoo and Suh 2006).

In our work, the robot plays the role of the user in context-aware applications. Here, we use context-aware concepts and tools to capture the robot's external context regarding environmental threats in order to model the robot's internal context regarding its 'emotional state' of either 'fear' or 'confidence'. However, context awareness by itself, as described in this section, is insufficient for capturing the sequence of events involving the robot and its context. For this reason, we also use concepts and tools for developing *situation-aware systems*, which we introduce next in Section 4.2.4.

4.2.4 Situation Awareness

We have introduced in Section 4.2.3 the definitions of context and context awareness formulated by Dey (2001). His definition of context, however, does not incorporate temporal properties. This is because, according to Dey (2001), the temporal aspects associated with the status of an entity are part of an extended conceptualization of context called *situation*. A situation describes a collection of states of relevant entities, where each state depicts those entities' context at a given point in time. In this sense, the term *situation awareness* could be understood as the act of being aware of the variations in an entity's context during a particular period of time. Note that Dey's definition of a situation has substantial similarity with the notion of episodic memory discussed in Section 4.1.2, which refers to our capacity of mentally recollecting and reconstructing the set of events composing an episode, ie., a personal experience.

Pereira, Costa and Almeida (2013) provide a practical definition for situation from the computational point of view that extends Dey's conceptualization of situation awareness and is the definition used in the HM model. According to Pereira, Costa and Almeida (2013), situations are 'composite entities whose constituents are other entities, their properties and the relations in which they are involved'. Situations also have a duration and can be referred as being current or past. Some examples of a situation are 'John has fever' (current situation), 'John had a fever of 38°C yesterday' (past situation) and 'Mary has been travelling for the last three days' (current situation).

Situations are classified by their types. A *situation type* describes the general characteristics of that situation, which is assigned to every instance of that situation type. An example of situation type is 'Patient has fever', whose instances would include any patient having a fever, for example: 'John has fever' and 'Mary has fever'. In this example, the 'Patient' is said to be a *situation participant*, i.e., an entity that is involved in the situation and may or may not be its main subject.

The process of identifying instances of a situation type is known as *situation detection* and consists basically of detecting instances of entities that are related to the situation and that satisfy constraints of the situation type. For example, a situation of type 'Patient has fever' is detected when an instance of the entity type 'Patient' satisfies the condition that defines the statement 'has fever'. An example of this condition is 'Patient has a temperature higher than 37°C'. Therefore, in this particular example, a situation of type 'Patient has fever' is said to be detected every time an instance of 'Patient' satisfies the condition 'Patient has a temperature higher than 37°C'.

The situation is said to be *active* and is considered a *current situation* while the condition of that particular situation type is satisfied. When this condition is no longer satisfied, the situation is said to be *inactive* and is considered a *past situation*. The point in time in which the situation starts to be active is known as *activation moment*, while the point in time in which the situation starts to be inactive is known as *deactivation moment*. For example, consider the previous example of the situation type 'Patient has fever', in which the 'has fever' statement is based on the condition 'Patient has a temperature higher than 37°C'. In this particular example, the instance 'John has fever' is said to be active and current when John's temperature becomes higher than 37°C. In the same way, the instance 'John has fever' is said to be inactive and past when John's temperature drops below 37°C.

Fig. 4.2 provides a graphical representation of the life cycle of three situations, which are instantiating the same situation type. The vertical axis depicts the possible state-of-affairs of the entities that may participate in the situation. The horizontal axis depicts the passing of time. For example, consider the previous example of the situation type 'Patient has fever' for the entity instance 'John'. The grey area represents the moments when John's temperature lies above 37°C. Fig. 4.2 depicts two past situations (situations 1 and 2, respectively), in which this particular example means that 'John had fever', and a current situation (situation 3), which means that 'John has fever'.

For the model of the HM, we are particularly interested in using techniques of situation awareness for simulating *situation appraisal*. In Rizzi et al. (2017), we define *situation appraisal* as one's ability to "make emotional evaluations and associations over perceived situations". In other words, situation appraisal is an individual's ability to attach emotional meanings to perceived situations and react accordingly. In this case, the HM should provide means for the robot to attach emotional meaning to experienced situations, where the emotional information comes from the Amygdala Module (AM). A detailed explanation on how

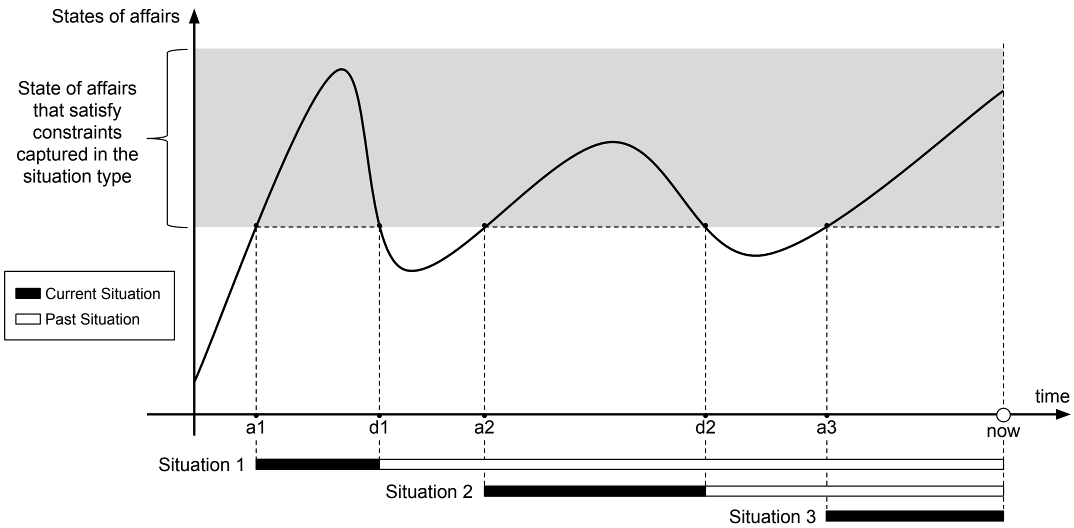


Figure 4.2: Example of situations' life cycle.

the emotional meaning is acquired by the HM and attributed to individual situation instances is given in Section 4.4.

4.3 Underlying Technology

This section presents the platforms for rule, event and situation management used to implement the contextual processing taking place in the Hippocampus Module of SAFEL. Section 4.3.1 briefly introduces the Drools platform, which provides a powerful environment and necessary tools for developing rule-based and complex event processing (CEP) systems. Section 4.3.2 provides a succinct introduction to SCENE, which is a robust platform for situation management that extends Drools.

The Drools platform implements the concepts presented in Section 4.2.1 and Section 4.2.2, facilitating the development of context-aware applications as discussed in Section 4.2.3. On the other hand, the SCENE platform implements the concepts presented in Section 4.2.4, which is essential for managing the temporal relationship between stimuli in the Hippocampus Module (HM).

4.3.1 Drools

Drools (Bali 2013) is a *Business Rules Management System* (BRMS) developed and supported by Red Hat and the JBoss Community (Red Hat 2017a,b). It provides a comprehensive platform for rules management and CEP, which include a rule engine, a rule language, a CEP engine and an integrated development environment (IDE).

Drools is divided into five modules, among which two are of interest to our work. These are the *Drools Expert* module, which implements the rule engine and provides the rule-based coding environment, and the *Drools Fusion*, which implements the CEP engine. The Drools Expert module is also responsible for managing the rule-based working memory (RBWM), which here we call as the *Drools working memory* (DWM) for disambiguation reasons.

Drools' rule engine uses the *Rete* algorithm for performing pattern matching. Rete (Latin for net) is a graph-based algorithm capable of efficiently comparing large collections of objects against large collections of patterns (Forgy 1982). Rules are declared by means of the *Drools Rule Language* (DRL), which also allows the declaration of functions, imports, global variables and rule attributes. For the sake of didactics, this section will focus on briefly explaining the rules mechanics only, since rules are the central concept of DRL.

Code 4.2 depicts the rule structure in DRL. The definition of a rule starts with

Code 4.2: Example of rule structure in DRL.

```
1 rule "Rule Name"
2    <attributes>
3 when
4    <conditions>
5 then
6    <actions>
7 end
```

its name (line 1), after which the rule's optional attributes may be defined (line 2). Some example of rule's attributes are *salience* (to define the rule's priority in relation to other rules in the code), *no-loop* (to prevent the actions of a rule from indefinitely re-executing itself) and *timer* (establishes an interval for recursively firing that rule if its patterns are still satisfied). The *when* keyword (line 3) determines the start point of the left-hand side (LHS), which is the place where the rule's conditions are defined, as explained in Section 4.2.1. On the other hand, the *then* keyword (line 5) determines the start point of the right-hand side (RHS), which is the place where the rule's actions are defined.

The RHS of a rule is implemented using the traditional Java syntax, whereas the implementation of rules' LHS is mostly based on the dialect of the *MVEL* language [43]. MVEL is a runtime expression language written in Java that allows expressing basic logic in Java-based applications. Code 4.3 shows a simple example of the MVEL syntax and how it is used inside a DRL rule.

The rule of Code 4.3, named 'Adult Detection', contains a single pattern stated in the MVEL dialect (line 3), which is matched against facts of the type 'Person' and is divided into two parts. The first part of this pattern contains a restriction, which states that only fact instances of the type 'Person' having the property (which may be called an *attribute* or a *field*) 'age' equals 18 can satisfy the LHS of the rule. If this restriction is satisfied, the second part of this pattern will assign the value of the 'name' field to the binding variable '$name'. The '$' character is optional and commonly used in DRL codes to differentiate variables and fields. The value of the 'name' property of matched fact instances is printed out in the rule's RHS (line 5) using the standard Java syntax.

For instance, consider the rule of Code 4.3 and suppose that the DWM contains two fact instances of the type 'Person'. One fact has the 'name' property equals John and the 'age' property equals 15, while the other fact has the 'name' property equals Mary and the 'age' property equals 20. In this case, Drools output would be: 'Adult detected. Name: Mary'. John's name would not be displayed because he is younger than 18 years, thus not satisfying the pattern's restriction.

Code 4.3: Simple example of a rule with a single pattern in DRL.

```
rule "Adult Detection"
when
  Person (age > 18, $name : name)
then
  System.out.println("Adult detected. Name: " + $name);
end
```

Code 4.4: Example of fact type definition in Drools.

```
public class Person {
    private String name;
    private int age;
    private int temperature;
    private Person father;
    //getters and setters ...
}
```

In Drools, fact types are defined by means of Java classes. Code 4.4 depicts the definition of the fact type ‘Person’ that has been used in the example of Code 4.3. Note in Code 4.4 that one of the attributes of the fact type ‘Person’ is another fact of the type ‘Person’, which establishes a relationship between two fact instances that, in this case, is fatherhood.

Code 4.5 shows a rule that has more than one pattern and uses the fatherhood relationship available in the ‘Person’ fact type class. The first pattern (line 3) is satisfied by instances of the fact type ‘Person’ whose ‘age’ attribute is greater than 18 years. The ‘$person’ variable receives a reference to the fact instance matched in the first pattern. Once the first pattern is satisfied, the second pattern (line 4) matches against each fact instance of the type ‘Person’ existing in the DWM whose ‘father’ attribute is equal to the reference stored in the variable ‘$person’. If a pair of facts exists in the DWM that satisfies the two patterns of this rule, then a message will be printed showing the father’s and offspring’s names. In other words, this rule executes the following reasoning: ‘Print a given message for each person in the DWM who has a father older than 18 years’.

For instance, suppose that the DWM currently has the fact instances shown in Table 4.1, all of which are instances of the fact type ‘Person’. In this case, Drools output would be:

- John is Mike’s father.
- John is Mary’s father.

Drools displays two messages about John because he is older than 18 years and has two sons known by the DWM, which are Mike and Mary. Nonetheless, nothing

Code 4.5: Example of a DRL rule with two patterns and a restriction on a relationship between two fact instances.

```
rule "Father Detection"
when
  $person : Person (age > 18, $fatherName : name)
  Person (father == $person, $name : name)
then
  System.out.println($fatherName + " is " + $name + "s father.");
end
```

Table 4.1: List of fact instances of type ‘Person’ defining the current state of the Drools Working Memory.

Instance Name	Instance Attributes		
	Name	Age	father
john	John	50	null
sam	Sam	30	null
mike	Mike	10	john
mary	Mary	15	john
jake	Jake	8	joe

is said about John’s father because the ‘father’ attribute of the fact instance ‘john’ is empty. In addition, there are no messages about Sam because, even though he is older than 18 years, he has no relatives inserted in the DWM. Finally, although Jake has a father, the fact instance representing his father has not been inserted in the DWM. For this reason, no message is displayed about Jake.

Drools also provides a robust support to CEP by means of the Drools Fusion module. Drools Fusion’s temporal reasoning implements all the temporal operators proposed by Allen (1981, 1983), as well as their logical complement. Allen (1981, 1983) describes *point-in-time events* as being instantaneous events, while *interval-based events* have duration and, consequently, are delimited by start and end timestamps. Table 4.2 shows all the temporal operators supported by Drools Fusion and their respective temporal reasoning for both point-in-time and temporal-based event relationships.

For instance, consider the rule of Code 4.6, which detects when a person’s fever is increasing. Also, suppose this system receives real-time body signals from the person being monitored. The ‘Fever’ class is declared as an event in line 2. The explicit declaration of the ‘Fever’ class as an event fact in line 2 is necessary for the rule engine to understand that instances of the type ‘Fever’ are not normal facts and have temporal properties.

To satisfy the conditions of the rule of Code 4.6, the DWM must contain

Table 4.2: Drools Fusion Temporal Operations.

Operation	Point-Point	Point-Interval	Interval-Interval
A before B B after A	A● B●	A●——● B●	A●——● B●——●
A meets B B met by A		A●——● B●	A●——● B●——●
A overlaps B B overlapped by A			A●——● B●——●
A finishes B B finished by A		A●——● B●	A●——● B●—●
A includes B B during A		A●——● B●	A●——● B●—●
A starts B B started by A		A●——● B●	A●——● B●—●
A coincides B	A● B●		A●——● B●——●

Code 4.6: Example of Drools rule with a temporal operation.

```
1  declare Fever
2     @role(event)
3  end
4
5  rule "Fever Increasing"
6  when
7     $fever : Fever ($person : person, $temp : temperature)
8     Fever (person == $person, temperature > $temp, this after $fever)
9  then
10    System.out.println($person.getName() + ", your fever has increased!");
11 end
```

at least two ‘Fever’ events with different temperatures that reference the same person, and the fever with higher temperature must have happened *after* (which is the temporal operation) the other one. Every time this combination of events happens, a message is printed by the RHS of the rule (line 10), warning the febrile person about his or her fever condition.

4.3.2 SCENE

SCENE (Pereira, Costa and Almeida 2013; Rizzi Raymundo et al. 2014) is an application programming interface that extends the DRL and makes use of Drools' rule and CEP engines to provide a powerful support to situation management.

Code 4.7: Example of situation type definition in SCENE.

```
1  public class Fever extends SituationType {
2
3      @Role(label = "$patient")
4      private Patient patient;
5
6      public Patient getPatient() {
7          return patient;
8      }
9
10     public void setPatient(Patient patient) {
11         this.patient = patient;
12     }
13 }
```

Code 4.8: Example of a situation rule in SCENE.

```
1 rule "Situation Detection: Febrile Patient"
2 @role(situation)
3 @type(Fever)
4 when
5     $patient : Patient(temperature > 37)
6 then
7    SituationHelper.situationDetected(drools);
8 end
```

SCENE specifies both structural and behavioural aspects of situations, which are represented by *situation types* and *situation rules*, respectively. Situation types may be defined in Java code or in DRL. For the sake of simplicity, here we focus on exemplifying the Java definition.

Situation types must extend the *SituationType* abstract class provided by SCENE, which implements the concept of situation type introduced in Section 4.2.4. Code 4.7 depicts the definition of the 'Fever' situation type, where the participant of the situation is the patient having a fever (line 4). The Java annotation in line 3 indicates how the participant (in this case, the patient) is referenced by the rule that manages the fever situation type. Each attribute declared as a participant of the situation must have *get* and *set* methods implemented (lines 06 to 12).

The behavioural part of a situation is characterised by means of conditional patterns defined in the LHS of a specialised Drools rule, which is declared as a *situation rule*. Code 4.8 depicts an example of a situation rule which manages situation instances of the type 'Fever' that has been defined in Code 4.7.

The rule of Code 4.8 matches against patients that have temperature greater than 37°C. Every time a Patient instance satisfying this condition is inserted in the DWM, SCENE creates and activates a new 'Fever' situation instance (line

7) by means of the *SituationHelper* class. The rule attribute declared in line 2 indicates to the rule engine that this is not a normal rule, but a situation rule that should be handled by SCENE. The rule attribute declared in line 3 informs SCENE about the situation type that will be instantiated if the conditions of this rule are satisfied. If the condition of the rule in Code 4.8 is satisfied, the object referenced by the '$patient' binding variable (line 5) becomes the participant of the new 'Fever' situation instance, as defined in line 3 of Code 4.7. After created, situation instances can be used in other rules' conditions, whether they are active or inactive.

Normally, the created 'Fever' instance would have to be explicitly deactivated, which would require the declaration of another rule establishing that a 'Fever' situation instance must be deactivated whenever its patient's temperature drops below 38°C. However, SCENE inherits a Drools feature known as *truth maintenance*, which allows Drools to automatically retract a fact instance when the conditions that instantiated such fact are no longer true. This feature makes rules easier to manage and interpret by approximating their behaviour to the human way of reasoning.

In the case of situations, SCENE automatically deactivates a situation instance if the conditions that led to the activation of such situation are no longer true. Therefore, 'Fever' instances created by the rule of Code 4.8 are automatically deactivated whenever the respective patient's temperature drops below 38°C. Note, however, that even though SCENE's truth maintenance is inherited from Drools, *retraction* and *deactivation* are distinct actions and have different outcomes. By *deactivating* a situation instance, SCENE is indicating to the rule engine that such situation became a past situation. However, information about that situation instance still exists in the DWM, and will only cease to exist after its *retraction*.

SCENE also inherits the temporal operations implemented by Drools Fusion and applies them to situation instances. SCENE's temporal operations rely on the comparison between the activation and deactivation moments of situation instances. By contrast to events, which can only be in the past, situations may be currently active, case in which they will not have a deactivation time stamp. Inactive situations (i.e., those in the past), in turn, have activation and deactivation time stamps. For this reason, inactive situations support all the temporal operations inherited from Drools Fusion, which cannot be said about active situations. Table 4.3 demonstrates the behaviour of SCENE's temporal operations when situation instances are active (dotted lines) and inactive (solid lines).

Table 4.3: SCENE Temporal Operations.

Operation	Active-Active	Active-Inactive	Inactive-Inactive
A before B B after A			
A meets B B met by A			
A overlaps B B overlapped by A			
A finishes B B finished by A			
A includes B B during A			
A starts B B started by A			
A coincides B			

Active Situation

Inactive Situation

Code 4.9: Example of a situation rule using a temporal operation from SCENE.

```
1 rule "Sinusitis Detection"
2 @role(situation)
3 @type(Sinusitis)
4 when
5   $fever : Fever($patient : patient, active)
6   Headache(patient == $patient, this during $fever)
7 then
8   SituationHelper.situationDetected(drools);
9 end
```

Code 4.9 shows a rule whose conditions match against two types of situation instances – 'Fever' and 'Headache' – in order to detected a third one: 'Sinusitis'. This rule performs the *during* temporal operation over situation instances of the types 'Fever' and 'Headache' (line 6). The 'active' constraint of the pattern in line 5 establishes that the 'Fever' situation must be still occurring for the rule to be satisfied. The 'patient == $patient' constraint in line 6 ensure that the patient participating in the 'Headache' situation is the same patient participating in the '$fever' situation instance. Finally, the 'this during $fever' constraint in line 6 establishes that the 'Headache' situation must have occurred (if it is a past situation) or still be occurring (if it is a current situation) during the occurrence of the '$fever' situation instance.

If the conditions of this rule's patterns are satisfied, SCENE creates and inserts in the DWM a new situation of type 'Sinusitis' (line 8), which will have as a participant the same patient participating in the 'Fever' and 'Headache' situations, referenced by the variable '$patient'.

4.4 Model

The concepts discussed in Section 4.2 are used in the Hippocampus Module (HM) to model the robot's external world regarding perceived environmental threats, as well as the robot's internal status regarding its 'fear' and 'confidence' levels depending on the environmental feedback. The tools introduced in Section 4.3, especially SCENE, are used to implement and simulate contextual fear conditioning (discussed in Section 4.1) by gathering and temporally organising pieces of contextual information, which are later assigned an emotional meaning reflecting the status of the environment in relation to aversive stimuli.

The HM receives two distinct inputs. The first input is a set of environmental stimuli represented by the vector $\vec{s}$, which is the same set of stimuli input delivered to the Amygdala Module (AM). However, unlike the AM, the HM does not require input stimuli to be categorized into neutral and aversive. The second input is the adrenaline signal outputted by the AM, a value in the range $[0, 1]$ representing the system's emotional feedback in regards to the robot's current state of affairs. The higher the adrenaline signal, the higher the fear level of the system; and the lower the adrenaline signal, the higher the system's confidence level.

Situation management in the HM is based on the following definitions:

Definition 1. *An event* $\mathbf{e_t}$ *is a collection of all stimuli detected by the robot's sensors at time* t*, so that* $\mathbf{e_t} = [s_1^t, s_2^t, ..., s_n^t]$*, where* s_i^t *is a normalized real value* $s_i^t \in [0, 1]$ *representing the intensity of stimulus of type* i *detected at time* t*.*

Definition 2. *A situation* S *is composed of the sequence of events occurring during its active period, so that* $S_j = [\mathbf{e_{a_j}}, \mathbf{e_{a_j+1}}, ..., \mathbf{e_{d_j}}]^T$*, where* a_j *and* d_j *are, respectively, the times of activation and deactivation of situation* j*. In the case* S_j *is a current situation, then* d_j *is the current time.*

There are four situation types in the HM: *unconditioned aversive* (UA), *neutral,* *safe* and *conditioned aversive* (CA). These four situation types have been implemented using SCENE and the conditions for their activation and deactivation is highly dependent on SCENE's temporal operators. The rules that instantiate these situations are defined in a Drools Rule Language (DRL) file, which is provided in Annex I. The patterns of these rules are constantly matched against the most recent adrenaline signal projected by the AM, which is continually updated in the Drools working memory (DWM). The management of the four situation types here defined is exemplified in Fig. 4.3 and the rules for their instantiation can be summarised as follows:

- *Unconditioned Aversive Situation*: An UA situation indicates the periods of time in which the robot was (or is, if it is a current situation) exposed to aversive stimuli. It is activated when the adrenaline signal rises above a given threshold (which is a predefined parameter of SAFEL) and is deactivated when the adrenaline signal returns to normal levels. Therefore, its duration is flexible and depends on the AM's emotional feedback.

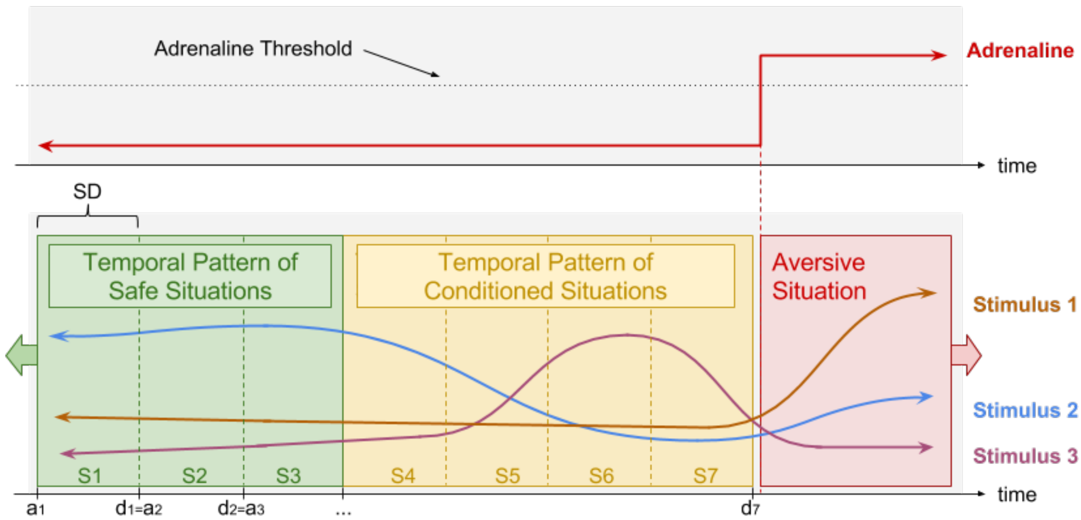


Figure 4.3: Situation Management in the Hippocampus Module (HM). Situations from S1 to S7 are initially neutral. Neutral situations become safe situations if no unconditioned aversive (UA) situation occur in a particular period of time after their deactivation moment (d_j), as exemplified by situations S1, S2 and S3. When the adrenaline signal rises above a pre-determined threshold, an UA situation is instantiated and all preceding neutral situations not yet categorised as safe become conditioned aversive (CA) situations, which in this example are situations S4, S5, S6 and S7. Note that, here we represent the temporal disposition of situations in a simplified manner for the sake of didactics, but in the HM's situation management model, situations' occurrence usually overlap each other.

- *Neutral Situation*: Neutral situations are those that have no emotional meaning to the robot, as they indicate neither safety nor threat. Unlike aversive situations, whose duration may vary and is dictated by the fluctuation of the adrenaline signal, neutral situations have fixed duration, which is given by a predefined parameter of SAFEL named *global situation duration* (GSD).

- *Safe Situation*: Safe situations are previously neutral situations that, after being deactivated, were observed to not precede any aversive situation. Therefore, ongoing safe situations indicate that the robot is not being exposed to aversive stimuli at the current moment and has no expectations to be exposed to aversive stimuli in the near future. The detection of safe situations is only possible after they are inactive (i.e., after they have become past situations), since they depend on the evaluation of events posterior to their own occurrence to be defined as such.

- *Conditioned Aversive Situation*: CA situations are previously neutral situations that, after being deactivated, were observed to precede an UA situation. Thus, the occurrence of CA situations indicates that the robot is likely to be exposed to aversive stimuli in the near future. They are said to be conditioned because, similarly to conditioned stimuli, they were previously neutral and acquired emotional meaning due to an association with an unconditioned aversive situation. Like safe situations, CA situations can only be detected after their deactivation moment, since they also depend on the evaluation of events posterior to their own occurrence to be defined as such.

Code 4.10 shows the rule responsible for instantiating new UA situations. This rule's LHS is satisfied when the level of the latest adrenaline signal received from the AM is above the predefined adrenaline threshold (line 5). If this condition is satisfied, SCENE creates and inserts in the DWM a new situation instance of type 'UnconditionedAversiveSituation' (line 7). Because of the truth maintenance system discussed in Section 4.3.2, the newly created 'UnconditionedAversiveSituation' situation is automatically deactivated when the adrenaline level drops below the predefined adrenaline threshold. The properties of events such as *Adrenaline* and situations such as *UnconditionedAversiveSituation* are defined in Java classes.

All situation instances are relayed from the HM to the Working Memory Module (WMM), with the exception of UA situations. The main task of the WMM, which will be further discussed in Chapter 5, is to predict the occurrence of aversive stimuli. Therefore, the WMM must comprehend and be aware of which event patterns precede the occurrence of aversive stimuli. UA situations are not sent

Code 4.10: Drools rule for instantiating an unconditioned aversive situation.

```
rule "Unconditioned Aversive Situation Instantiation"
@role(situation)
@type(UnconditionedAversiveSituation)
when
   $adrenaline : Adrenaline (level >= hippocampus.getAdrenalineThreshold())
then
   SituationHelper.situationDetected(drools);
end
```

Code 4.11: Example of temporal operation in the Drools rules of the Hippocampus Module.

```
1 rule "Conditioned Aversive Situation Projection"
2 salience 20
3 when
4    $ua : UnconditionedAversiveSituation (active)
5    $neutral : NeutralSituation(projected, this before $ua)
6 then
7    $neutral.projectAs("aversive");
8    retract($neutral);
9 end
```

to the WMM because they co-occur with aversive stimuli instead of preceding it, thus having no valuable information for the WMM in terms of predictions.

The HM projects situation instances to the WMM along with their emotional category, i.e., neutral, safe or CA. Consequently, every situation instance is sent in two distinct moments to the WMM, first when they are still neutral situations, and then a few time steps later when the HM is able to categorise them into either safe or (conditioned) aversive. The dual submission of the same situation instance, but with different situation types, is essential for the WMM to perform its associative learning, which is discussed in Chapter 5.

Code 4.11 shows the rule responsible for projecting CA situations to the WMM when they are detected, which contains a temporal operation. This rule's patterns are satisfied when an active UA situation is detected (line 4) and there are neutral situations (which have already been projected to the WMM categorised as neutral situations) occurring before (which is the temporal operation) the detection of that UA situation (line 5). If these conditions are satisfied, then those same neutral situations are projected once again to the WMM (line 7), but this time categorised as CA situations, and then retracted from the DWM (line 8).

Fig. 4.4 shows an example of situations' life-cycle over time that is more realistic than the example given in Fig. 4.3, where Fig. 4.4a shows the adrenaline signal over time, and Fig. 4.4b, Fig. 4.4c and Fig. 4.4d show situations' status in the system at time t_{10}, t_{13} and t_{14}, respectively. The activation and deactivation moments

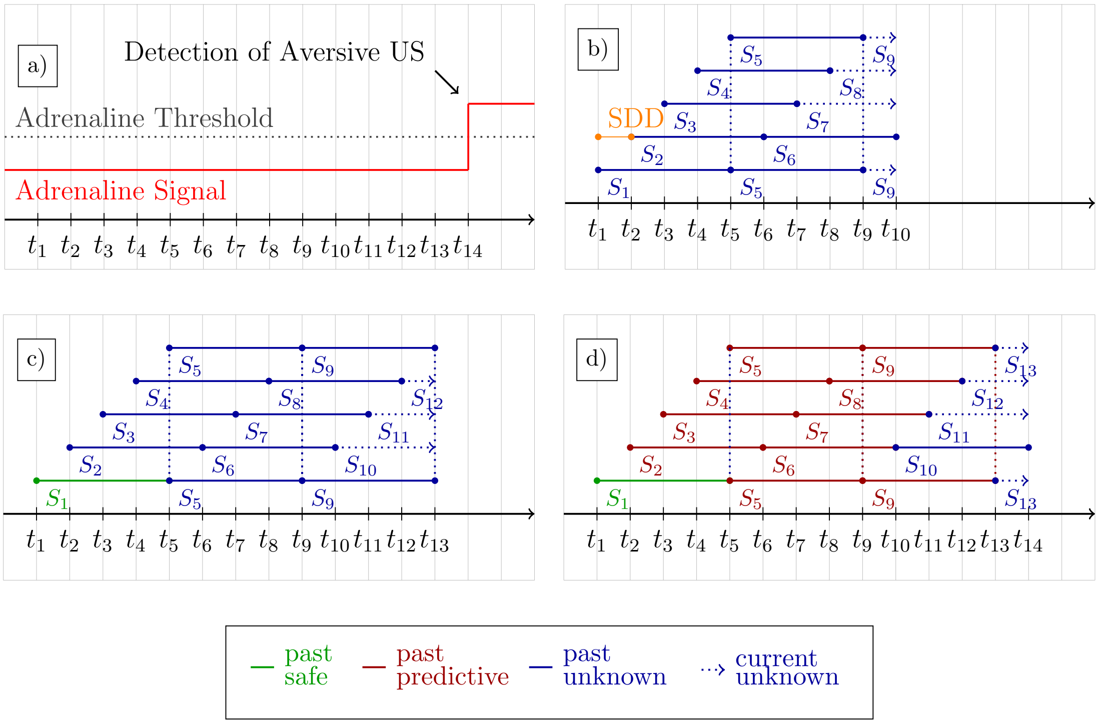


Figure 4.4: Example of situations' status over time. In Fig. (b), (c) and (d), the horizontal axis indicates the time step, and overlapping situations are vertically disposed, for the sake of readability. (a) The behaviour of the adrenaline signal over time. Fig. (b), (c) and (d) show the status of situations' type at times t_{10}, t_{13} and t_{14}, respectively.

of situation S_i, which are a_i and d_i respectively, are given by the time stamps delimiting the duration of situation S_i. For instance, situation S_1 has activation time $a_1 = t_1$ and deactivation time $d_1 = t_5$, situation S_2 has activation time $a_2 = t_2$ and deactivation time $d_2 = t_6$, and so on. Analogously, $S_1 = [\mathbf{e_{t_1}}, \mathbf{e_{t_2}}, \mathbf{e_{t_3}}, \mathbf{e_{t_4}}, \mathbf{e_{t_5}}]$, $S_2 = [\mathbf{e_{t_2}}, \mathbf{e_{t_3}}, \mathbf{e_{t_4}}, \mathbf{e_{t_5}}, \mathbf{e_{t_6}}]$, and so on.

Observe that situations can overlap each other. For example, situation S_2 is activated while situation S_1 is active; situation S_3 is activated while situations S_1 and S_2 are active, etc. Consequently, two or more situations can contain the same event. For instance, event $\mathbf{e_{t_4}}$ belongs to situations S_1, S_2, S_3 and S_4. Also, neutral situations can be either current or past. For instance, in Fig. 4.4b, situations from S_1 to S_6 are past situations because they have already finished by time t_{10}, while situations S_7, S_8 and S_9 are current because they are still occurring at time t_{10}.

A new neutral situation is activated every fixed number of time steps given by an internal parameter of SAFEL called *situation detection delay* (SDD) (Fig. 4.4b), which defines the period of time between the activation of a given situation and the activation of its predecessor situation. The SDD used to be a predefined user parameter in previous versions of SAFEL. In order to transform the SDD into an internal parameter and reduce users' parameter settings, we investigated which ratio between the global situation duration (GSD) and the SDD that yields the highest predictive performance from SAFEL (Rizzi, Johnson and Vargas 2016). This study is presented later, in Section 5.4.4.

As previously mentioned, neutral situations may become safe or (conditioned) aversive after their deactivation moment, but only if certain constraints are satisfied. For instance, all situations detected in Fig. 4.4b are still neutral because nothing can be said about them at time t_{10}. To be considered safe, a situation must be a past situation and be followed by at least two consecutive past neutral situations. This is to ensure that this situation will never precede or co-occur with any CA or aversive situation. To be considered CA, a situation must precede a peak in the adrenaline level and not have been categorised as safe. All situations are still considered neutral at time t_{10} in Fig. 4.4b because none of these conditions has been matched by time t_{10}.

However, at moment t_{13} in Fig. 4.4c, the conditions for categorising a neutral situation into a safe situation are satisfied by the current status of situation S_1. At time t_{13}, situation S_1 is past and precedes the past neutral situation S_5 that, in turn, precedes the past neutral situation S_9. Thus, at time t_{13}, situation S_1 leaves the status of neutral and becomes a safe situation. Similarly, the conditions for categorising a neutral situation as a CA situation are also satisfied by the current status of situations going from S_2 to S_9 at time t_{14} in Fig. 4.4d. Note in Fig. 4.4a that the adrenaline level rises above the predetermined adrenaline threshold at time t_{14}. Thus, at time t_{14}, situations going from S_2 to S_9 leave the status of neutral and are classified as CA situations. This change of status occurs because these situations are the only currently neutral situations preceding the rise of adrenaline at time t_{14} that have not yet been categorised as safe.

Safe and CA situations are immediately projected to the WMM in the moment they are categorised as such, whereas neutral situations are projected to the WMM at their deactivation time. Consequently, every factually safe and CA situation is sent twice to the WMM: first when it is still neutral and has just been deactivated; and then again a few time steps later, when the HM is able to determine whether it is actually a safe or CA situation. In the example of Fig. 4.4, for instance, situation S_1 is sent to the WMM at time t_5 as a neutral situation and at time t_{13} as a safe situation. Analogously, situation S_5 is sent to the WMM at time t_9 as a neutral situation and at time t_{14} as a CA situation. The dual submission of the same situation instance, but with different situation types, is essential for the WMM to perform its task, which is discussed in the next chapter.

4.5 Final Considerations

This chapter presented the design and implementation of the Hippocampus Module (HM), which is the module of SAFEL responsible for context processing. The model of the HM has been first proposed by us in Rizzi Raymundo, Johnson and Vargas (2015), implemented and evaluated in Rizzi et al. (2017) and later improved by us in Rizzi, Johnson and Vargas (2016). SAFEL has suffered minor modifications after the improvement proposed in Rizzi, Johnson and Vargas (2016), which are mostly related to terminology and number of conditioned aversive (CA) situations detected per unconditioned aversive (UA) situation.

The HM model attempts to simulate two essential cognitive processes taking place in the hippocampus, which are the associative representation and the sequential organisation processes, discussed in Section 4.1.2. While associative representation regards to the hippocampus capacity of multi-stimulus association, the sequential organisation addresses the temporal organisation of events composing an episodic memory. Together, these two cognitive processes create the basis for context processing, representation and memory in the brain, and are believed to also underlie the phenomenon of contextual fear conditioning. To simulate these cognitive processes, we have based the HM model on computational concepts of knowledge representation and situation awareness, discussed in Section 4.2, which are inspired by theories on the human cognitive processes of reasoning and context representation.

We used Drools to implement the concepts presented in Section 4.2, which is a comprehensive platform that provides all the necessary tools along with powerful engines for developing systems based knowledge representation techniques. We have also used SCENE, which is a robust platform that extends Drools and facilitates the development of situation-aware systems.

Different from the Amygdala Module (AM), whose model is inspired by the underlying neural mechanisms taking place in the amygdala regions, the HM has been designed at a higher level of abstraction based on the hippocampal functions, instead of physical and chemical phenomena. As discussed in Section 4.4, we have used a rule-based situation-aware approach to capture the external context of the robot in regards to environmental threats. Feedback from this external context is later integrated with feedback from the AM in order to simulate the robot's internal and emotional context.

We have opted for designing the HM based on concepts of knowledge representation because we believe that a considerably large ANN would be required to simulate the complex temporal and multi-stimulus information handled in hippocampal regions of the brain. The long-term potentiation (LTP) process taking place in the hippocampus (Eichenbaum 2004) is considerably more complex than the basic LTP process simulated in the AM. However, we believe that it may be possible to partially simulate hippocampal functions, especially associative representation and sequential organisation, using deep neural networks (Schmidhuber 2015), which we intend to investigate in future work.

Preliminary tests of the HM have been performed (Rizzi et al. 2017; Rizzi, Johnson and Vargas 2016). However, these preliminary tests have been left for discussion in Chapter 5 because they also depend on the functioning of the Working Memory Module (WMM) of SAFEL.

Chapter 5

Working Memory Module

Unlike the previously discussed modules of SAFEL, which are inspired by regions of the brain (the amygdala and the hippocampus), the module presented in this chapter is rather inspired by a cognitive function of the brain, known as the *working memory*. A variety of studies indicate that prefrontal cortex areas and the anterior cingulate region (Fig. 5.1) are involved in working memory functions (LeDoux 2003; Krause-Utz et al. 2014; Spellman et al. 2015). The working memory is believed to play an important role in consciousness, learning and reasoning (Baddeley and Hitch 1974; Baddeley 1995).

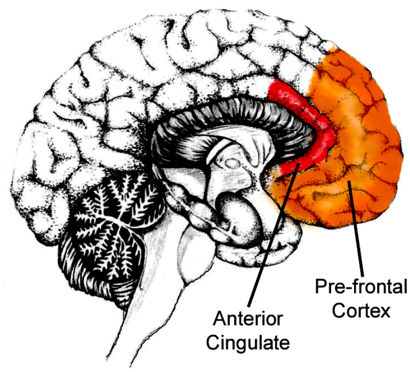


Figure 5.1: Regions of the brain involved in working memory functions.

The following sections discuss neuroscience theories of the working memory functions that have inspired the Working Memory Module (WMM) of SAFEL and how we have computationally designed it. Section 5.1 discusses such theories of the working memory and how it creates the experience of conscious fear. Section 5.2 discusses the underlying technology used in the implementation of the WMM, whose design is later presented in Section 5.3. Finally, Section 5.4 presents the preliminary experiments that have been performed to evaluate the Hippocampus

Module (HM) and the WMM.

In regards to the research questions introduced in Section 1.4, this chapter contributes to answering them by:

1. addressing all the requirements of a situation-aware intelligence (Section 1.3.1) and of an emotional intelligence (Section 1.3.2), by combining and associating the knowledge generated in the Amygdala Module (AM) and HM;

2. presenting the design and implementation of associative learning between contextual and emotional knowledge using a binary classification tree, which is the third and final approach of the hybrid model of SAFEL.

3. analysing experiments performed with a real robot that demonstrate the successful application of the HM and the WMM for robotics purposes.

5.1 Biological Background

In Chapter 3 and Chapter 4 we have discussed cued and contextual fear conditioning, respectively. However, we have not yet addressed the conscious experience of fear that occurs to an individual when he or she is in danger. Theories of consciousness have been proposed to date that relate it with a cognitive function of the brain, known as the *working memory* (Baddeley and Hitch 1974; Baddeley 1995), which LeDoux (2003) describes as 'a serially organized mental workspace where things can be compared and contrasted and mentally manipulated'.

According to LeDoux (2003), sensed stimuli and stored representations of context are fused in the working memory through interactions between brain regions that include the pre-frontal cortex, the hippocampus and related areas in the temporal lobe. In the case of a fearful experience, these interactions will also involve the amygdala and related regions, which in turn warn the working memory about the activation of the fear system of the brain. In fact, the working memory receives a greater number and variety of inputs in the presence of emotional stimuli than in the presence of other types of stimuli. The influence of the amygdala on the conscious perception of an object or event is believed to be the main condition for the subjective experience of an emotional state of fear.

We have discussed in Chapter 4 that the amygdala and hippocampal systems work in parallel, forming what LeDoux (1999) calls, respectively, as *emotional memory* and *memory of emotion*. If an individual is exposed to stimuli that were present during a previously experienced trauma, both the amygdala and hippocampal systems are activated and work in parallel to retrieve emotional and contextual memories about the event, respectively. Because these two memories are simultaneously recovered in response to the same stimuli, they are experienced as if they were one single memory.

According to LeDoux (1999), the working memory is where these memories are retrieved, fused and consciously experienced as if they were a unified memory. In other words, the working memory allows the association of explicit contextual memory formed in the hippocampus with implicit emotional memory formed in the amygdala to create memories that are contextual and emotional at the same time. In addition, because its functions involve conscious reasoning, the working memory allows the individual to consciously reason and decide a course of action based on the information made available by that unified contextual and emotional memory.

The WMM of SAFEL works in a similar manner by associating the temporal patterns of situation instances formed in the Hippocampus Module (HM) with their emotional meaning given by the Amygdala Module (AM), which may be 'safe' or (conditioned) 'aversive'. The main function of the WMM is to retrieve fear memories based on the current context of the robot to predict the occurrence of imminent unpleasant events. To do that, the WMM compares the current state of affairs of the robot with previously experienced situations that preceded an unpleasant event in the past. Whenever the WMM detects a current situation that is similar to a situation that preceded an aversive stimulus in a past experience, it will retrieve the same state of fear triggered at that time and warn the robot controller that an undesirable situation is likely to happen in the near future. By doing so, SAFEL provides the robot with an opportunity to act in advance and maybe prevent the occurrence of that aversive stimulus.

5.2 Underlying Technology

As previously discussed, the main task of the WMM is to associate the contextual memories formed in the HM with their emotional meaning given by the AM. This associative learning is implemented in the WMM using a *binary classification tree* (Breiman et al. 1984), which is used to classify situation instances generated in the HM into safe or (conditioned) aversive. In a binary classification tree, exemplified in Fig. 5.2, each node corresponds to a binary predicate on one attribute, where one branch from the node represents positive instances of the predicate and the other branch represents negative instances. Each leaf node is labelled by a class, which in our case depicts an emotional category (safe or aversive). For example, to predict the emotional category of a situation instance, a path from the root node

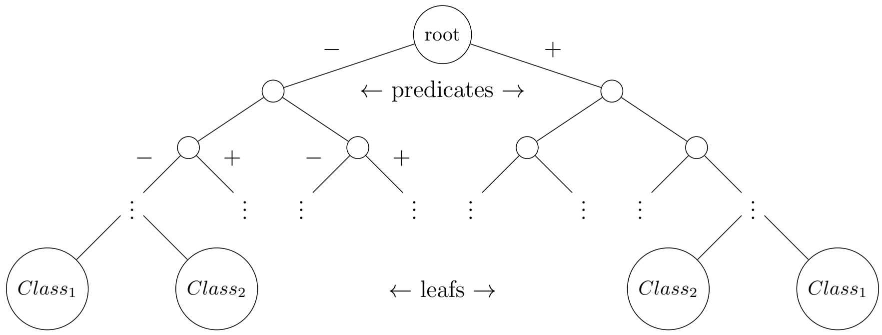


Figure 5.2: Example of a binary classification tree.

to a leaf must be found. This path depends on the values of the visited nodes in the tree, each of which is depicted by a predicate on a particular temporal property of situation information.

Classification trees are generated by first analysing the training dataset and determining a hierarchy of binary splits so that the data in each of the descendant nodes are 'purer' than the data in their parent node. The concept of 'purity' here is related to the class homogeneity of a node, where the smaller the number of classes related to a node, the 'purer' this node is.

Different metrics can be used to measure node impurity. We use the *Gini Index* (GI) as impurity metric to generate the classification tree of the WMM, which is given by Eq. 5.1.

$$GI(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2, \tag{5.1}$$

where c is the number of classes and $p(i|t)$ is the proportion of cases belonging to class i at a given node t. A node with just one class is said to be a pure node and has GI equals zero. Nodes with more than one class have positive GI, where the more classes it has, the higher the GI. The generation of the classification tree consists of splitting nodes so to minimize their impurity index. This process is recursively repeated for the child nodes, stopping when a pure node is found or when a stopping criterion is reached, such as a maximum number of splits or maximum tree depth.

The following design reasons led us to adopt the binary classification tree for implementing the associative learning of the WMM:

- *Interpretable*: classification trees are white-box algorithms, thus allowing one to easily interpret the logic behind the robot's learning and emotional response to stimuli.

- *Implicit feature selection*: classification trees are built by dynamically selecting the most informative features, and ignoring information that is irrelevant for the predictions. This is essential for the WMM because in most cases only a subset of the robot's sensors will provide valuable information about the pattern of a specific situation. For instance, a robot may require a camera, face recognition algorithms and sonar sensors to detect that a person is nearby, but many other sensor information (e.g., internal temperature, accelerometer and battery level) would not give valuable information for this particular task. Analogously, some predictions of aversive situation could heavily rely on information from some sensors while disregarding information from another sensors. If predictions are based on the wrong kind of stimulus information then the predictive performance could be potentially compromised. Therefore, the WMM must be capable to detect and ignore stimulus information that does not contribute to characterizing and predicting aversive situations.

- *Fast training and classification*: the classification tree is an algorithm well known by its fast training and classification processes (Lim, Loh and Shih 2000). This is important because SAFEL's emotional learning greatly relies on constantly retraining the classifier of the WMM at runtime. The slower the retraining and classification processes are, the more time the robot would take to present an emotional reaction to the current state of affairs.

- *Non-parametric*: classification trees are non-parametric algorithms, which means that they do not require the specification of parameters that depend on the distribution of data. One of SAFEL's goals is to be of general purpose. To be applicable to a variety of environmental characteristics, SAFEL's learning must be independent of data shape.

5.3 Model

The WMM is the module of SAFEL where the association between context and "fear" takes place. In the Working Memory Module (WMM), the temporal patterns of situations are memorised and associated with their respective labels (safe or (conditioned) aversive). Two processes take place in the WMM. First, a feature extraction is performed in order to generate compacted versions of situational information containing only the most relevant characteristics of situations' temporal patterns. Later, these compacted situations are delivered to a binary classification tree for associative learning and prediction. These two processes are respectively addressed in Section 5.3.1 and Section 5.3.2.

5.3.1 Unitary Representation of Context

In the WMM, situation instances coming from the HM pass through a feature extraction process in order to generate compacted versions of situational information. This phase consists of extracting relevant information that characterizes the pattern of stimuli's variation over time during the active time of the respective situations.

Chapter 4 has largely discussed the meaning of *event* and *situation* from a conceptual and computational point of view. A mathematical representation has been given to these concepts in Def. 1 and Def. 2, respectively, which are relevant for the explanation given in this section. For convenience, we paraphrase below the definitions of Chapter 4. Def. 1 states that:

> An event $\mathbf{e_t}$ is a collection of all stimuli detected by the robot's sensors at time t, so that $\mathbf{e_t} = [s_1^t, s_2^t, ..., s_n^t]$, where s_i^t is a normalized real value $s_i^t \in [0, 1]$ representing the intensity of stimulus of type i detected at time t.

While Def. 2 states that:

> A situation S is composed of the sequence of events occurring during its active period, so that $S_j = [\mathbf{e_{a_j}}, \mathbf{e_{a_j+1}}, ..., \mathbf{e_{d_j}}]^T$, where a_j and d_j are, respectively, the times of activation and deactivation of situation j. In the case S_j is a current situation, then d_j is the current time.

From Def. 1 and Def. 2, and supposing that $a_j = 1$ and $d_j = m$, and that the robot has n sensory inputs, we have that:

$$S_j = \begin{pmatrix} \mathbf{e_1} \\ \mathbf{e_2} \\ \vdots \\ \mathbf{e_m} \end{pmatrix} = \begin{pmatrix} s_1^1 & s_2^1 & \cdots & s_n^1 \\ s_1^2 & s_2^2 & \cdots & s_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ s_1^m & s_2^m & \cdots & s_n^m \end{pmatrix}. \tag{5.2}$$

From Eq. 5.2 we can say that

$$S_j = [\mathbf{s_1}, \mathbf{s_2}, ..., \mathbf{s_n}], \tag{5.3}$$

where $\mathbf{s_i} = [s_i^1, ..., s_i^m]^T$ and represents the behaviour of stimuli s_i from time $t = 1$ to time $t = m$, which are the activation and deactivation moments of situation S_j. The feature extraction process taking place in the WMM consists of generating from S_j a new piece of situation information S'_j, given by Eq. 5.4:

$$S'_j = [\overline{s}_1, ..., \overline{s}_n, \gamma_1, ..., \gamma_n, \eta_1, ..., \eta_n], \tag{5.4}$$

where $\overline{s}_i$, γ_i and η_i are, respectively, the mean, skewness and number of local maxima of $\mathbf{s_i}$ (Eq. 5.3). The mean value of stimulus s_i, given by $\overline{s}_i$, provides the average intensity of stimulus s_i during the active time of situation S_j. On the other hand, the skewness of stimulus s_i, given by γ_i, provides the approximate time interval when stimulus s_i was more intense during the active time of situation S_j. Finally, the number of local maxima of stimulus s_i, given by η_i, provides the detection frequency of stimulus s_i during the active time of situation S_j. Together, $\overline{s}_i$, γ_i and η_i provide three essential information on the behavioural characteristics of stimulus s_i during the active life-cycle of S_j.

We discussed in Section 4.1 the unitary representation of context in the brain, which represents the main distinction between contextual and cued conditioning. While information processed in the amygdala is purely perceptual (e.g., visual, auditory, olfactory), the unitary representation of context formed in the hippocampus binds all these stimuli along with their inter- and temporal relationship. The feature extraction performed in the WMM is analogous to this unitary representation of context in the brain in the sense that it binds the stimuli composing a situation instance in order to extract intrinsic data characteristics depicting their inter- and temporal relationship during the active time of that situation.

The binding of stimuli is, therefore, performed in two steps. The first part of stimuli binding takes place in the HM, which assembles stimuli information within particular time intervals and attaches to them an emotional meaning, based on the emotional feedback from the AM. At this phase, it is still unknown to SAFEL how stimuli composing situations relate to each other. The second part of stimuli binding takes place in the WMM, which consolidates situation information coming from the HM into a unified representation that expresses the temporal relationship of stimuli composing individual situation instances. After both stimuli-binding phases are completed, in the HM and WMM respectively, SAFEL becomes aware not only of the emotional meaning of situations but also of the temporal relationship between stimuli composing them.

S'_j can also be seen as an approximated representation of S_j, in the sense that it does not comprise all the information contained in S_j, but describes S_j with sufficient accuracy. This is important especially because the resulting situation instances will later compose a dataset for training the classification tree, as explained in Section 5.3.2. By using approximate representations of situation instances instead of the actual situation instances themselves we prevent poor generalization of situations' temporal properties derived from overfitting the training data.

This feature extraction phase is also useful for data compression, by reducing dimensionality and potential data redundancies. The volume of information about situation j is reduced from a matrix S_j of size $n \times m$ to a vector S'_j of size $3n$ after the feature extraction. This is especially effective when $m \gg n$, which is, in fact, the most common case, as the number m of time steps within a situation instance is usually much larger than the number n of sensory inputs a robot may offer.

5.3.2 Associative Learning

The associative learning of the working memory module is implemented using a *binary classification tree* (Breiman et al. 1984), whose basic functioning and main advantages has been briefly explained in Section 5.2. The situation information resulting from the feature-extraction process described in Section 5.3.1 composes the input set delivered to the classification tree for both training and prediction. The classes, i.e., the tree's resulting predictions, represent the emotional meaning of the respective situations, which may be either safe or (conditioned) aversive. In other words, the tree learns the temporal patterns of situations instances stored in S'_j and associates them with their respective emotional label (safe or aversive).

As mentioned in Section 4.4, every single situation instance is relayed in two distinct moments by the HM to the WMM: first when it is a neutral situation, and later when the HM can ensure that it is either safe or aversive (a conditioned aversive (CA) situation in this case). Emotionally categorised situations (i.e., which are either safe or aversive) are used to train the classification tree, which then learns the temporal patterns that characterise safe and aversive situations in the robot's current environment. On the other hand, the tree uses emotionally uncategorised situations (i.e., neutral situations) to try and predict whether an unconditioned aversive situation will occur in the near future, by matching its temporal patterns against those of previously learned situations.

Therefore, at time d_j (i.e., when situation S_j has just been deactivated), the HM will send S_j as a neutral situation to the WMM, where it is transformed into S'_j and submitted to the binary tree for classification. The tree will classify that situation into safe or aversive based on past situation experiences of the robot. Then, at time t_n, where $t_n > d_j$, situation information S_j will be sent to the WMM once again, but this time labelled as either safe of aversive. The generated situation pattern S'_j and its type (safe or aversive) will now be used for retraining the classification tree, providing it with one more situation experience where to base its future predictions.

Fig. 5.3 exemplifies how the learning and prediction processes take place in the WMM. At time t_0, situation S_1 is sent for the first time from the HM to the WMM, when it is still a neutral situation. The classification tree in the WMM

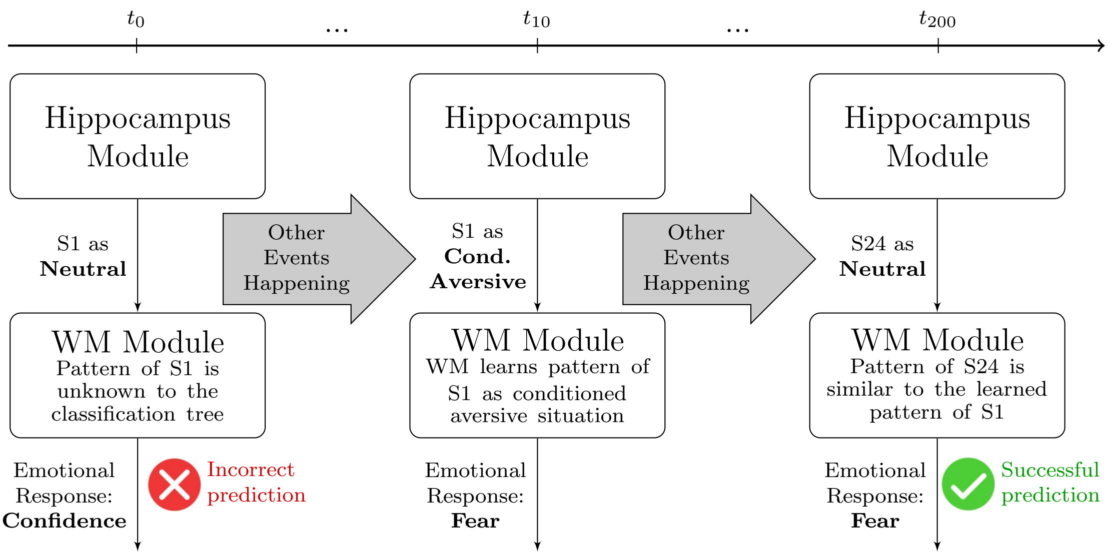


Figure 5.3: Progression of learning and prediction in the Working Memory Module (WMM).

will attempt to predict the emotional meaning of S_1 based on learned patterns of previously experienced situations. Because the classification tree has not previously experienced any situation that is similar to S_1, it may output an incorrect emotional response.

At time t_{10}, the HM is able to ensure that S_1 was actually a conditioned aversive (CA) situation because after its deactivation moment S_1 was observed to precede an unconditioned aversive (UA) situation. Situation S_1 is sent for the second time from the HM to the WMM, but this time labelled as a CA situation. This information is used in the WMM to retrain and update the classification tree, which will then become able to recognise and associate situations similar to S_1 with fear.

At time t_{200}, situation S_{24}, which is similar to situation S_1 in regards to stimuli temporal pattern, is sent for the first time from the HM to the WMM as a neutral situation. When arriving in the WMM, the neutral situation S_{24} will pass through the feature extraction process and then be compared by the classification tree with previously experienced situation instances. The classification tree is likely to recognise the similarity between situations S_1 and S_{24} and return the emotional label that has been assigned to situation S_1 at time t_{10}, which is conditioned aversive. Consequently, the WMM will be predicting that situation S_{24} is aversive before the HM can provide evidence for that. If correct, this prediction would warn the robot with antecedence about a potential imminent threat, thus giving the robot a chance to act towards avoiding this threat before its occurrence.

Note that pre-training SAFEL prior to environmental exploration is optional.

The dataset used to train the decision tree can start empty, with no knowledge about the current environment. As the robot explores the environment and experiences new aversive situations, the dataset grows and the tree is retrained. Therefore, the robot's capability to predict imminent aversive events improves with experience, as it explores the environment.

In addition, the tree is constantly retrained with the latest pre-defined number of observed situation instances. This means that the WMM is capable of gradually forgetting previously learned associations that are no longer consistent with the current state of the robot's environment. For instance, if a particular situation that was safe in a previous environment is now aversive in the current environment, then the classification tree will gradually forget the previous association of that situation with safety and create a new association with fear. This re-learning process occurs gradually as the tree is retrained with information from the most recent observations of the robot in the new environment.

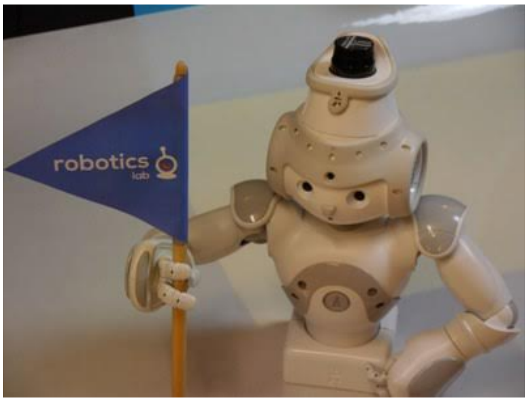
5.4 Preliminary Tests

The experiment performed here aim to evaluate exclusively the Hippocampus Module (HM) and the Working Memory Module (WMM). This means that the Amygdala Module (AM) is not included in the evaluation of the experiments presented in this section. A proper evaluation focused exclusively on the performance of the AM has been discussed in Chapter 3. Also, a full evaluation of all the modules of SAFEL working in conjunction is discussed in depth in Chapter 6.

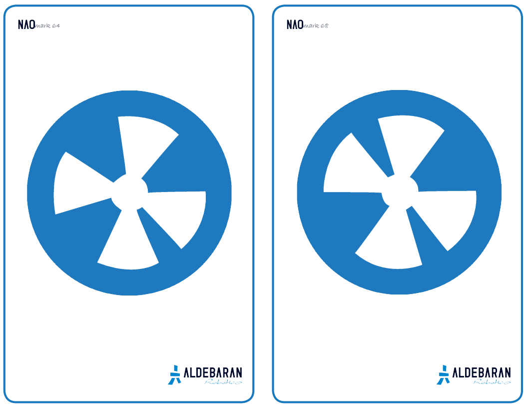
This section presents three experiments. The first experiment, presented in Section 5.4.3, evaluates the first version of the HM and the WMM in terms of predictive performance and robustness. The second experiment, presented in Section 5.4.4, proposes a method for improving the predictive performance of these two modules of SAFEL while reducing user parameter setting. Finally, the third experiment, presented in Section 5.4.5, briefly discusses a comparative analysis with the BEL model, which has been previously discussed in Chapter 2.

All the experiments presented in this section have been conducted using a NAO humanoid robot, model T14 (Fig. 5.4a). NAO is one of the most widely used robots in the Human-Robot Interaction (HRI) field of research (Weiss and Bartneck 2015). By using NAO, we hope to facilitate the reproduction of our work, as well as the implementation of future comparative studies.

In addition, by using a physical robot in these experiments, we aim at exposing SAFEL to noises and reading failures characteristic of real robot sensors. In a virtually simulated environment, the quality of sensor reading could be greatly



(a) The NAO robot, model T14.



(b) Examples of NAOmark.

Figure 5.4: Gadgets used in the experiments.

improved in comparison to real sensors, providing smoother data and possibly facilitating SAFEL's predictions. All sensor noises and detection failures were preserved during the experiments, so to analyse how it would affect SAFEL's prediction performance. We have used four types of sensor readings to represent NAO's perception of environmental stimuli, which are:

- s_1: light level,
- s_2: number of human faces detected,
- s_3: identification of *NAOmarks*, which are landmark images with specific patterns that NAO robots can recognise and identify (Fig. 5.4b),
- s_4: sound detection confidence, which is a number in the range [0,1] depicting NAO's confidence that a particular detected sound is real.

In the experiments presented here, the aversive stimulus is represented by darkness, which is an analogy to the natural fear and stress that most animals experience when they become unable to see. Hence, before running the experiments, the HM of SAFEL was configured to increase adrenaline levels whenever NAO detected low light levels. Remember that the AM is not part of this experiment and, therefore, the adrenaline signal must be fixed in the HM according to a given condition, which in this case is the light level. The remaining environmental stimuli (i.e., human faces, NAOmarks and sound detection) were initially neutral.

In order to create a controlled test environment, where we could analyse the influence of the same set of situations under different parameter settings, we have separated the experiments into three phases. First, we collected data by repeatedly presenting the above-listed stimuli to NAO and then storing NAO's sensor readings. In the second phase, we assembled the collected data in a specific time line, creating a dataset that was reproduced for different parameters and configurations. Lastly, we ran SAFEL on each dataset independently, during which the instances of the datasets were presented sequentially to SAFEL as if it was being executed in the robot at real time. The first and second phases are explained next in Section 5.4.1 and Section 5.4.2. The third phase is explained in the results description of the respective experiments (Section 5.4.3, Section 5.4.4 and Section 5.4.5).

5.4.1 Data Collection

We have collected data respecting six distinct situation patterns. A *situation pattern* is the set of main temporal aspects (such as average time delay and temporal sequence of stimuli) that characterizes a given situation. Hence, a situation instance is in a sense an instantiation of a situation pattern and must have all the temporal properties that characterise that pattern (e.g., a specific order of stimulus detection). This is not to be confused with how a situation instance instantiates a situation type, as discussed in Chapter 4. Note that a situation instance instantiates a situation type in terms of its properties and participants, while the same situation instance can instantiate a situation pattern in terms of the temporal organization of its composing stimuli.

Fig. 5.5 shows examples of NAO's sensor readings for each of the six situation patterns induced in the experiment. For example, the pattern of the situation observed in Fig. 5.5b is characterised by the detection of a human face followed by the detection of a NAOmark. To collect data for situation instances with this pattern, we first presented a human face to the robot for about five seconds. Afterwards, the human face was hidden from the robot and a NAOmark was presented instead, also for about five seconds. This procedure has been performed at good light conditions, so the robot could easily detect both human faces and NAOmarks. The same procedure was then independently repeated several times in order to collect many different instances of this same situation pattern.

Analogously, to collect instances like the one seen in Fig. 5.5c, we presented the NAOmark and a human face at the same time to the robot at good light conditions for about five seconds and then moved both away from the robot's range of vision. Again, we repeated this procedure several times in order to collect many different instances of this same pattern. The same sequence of steps was performed for collecting instances of the remaining situation patterns in Fig. 5.5.

Fig. 5.5a depicts an example of a conditioned aversive (CA) situation followed

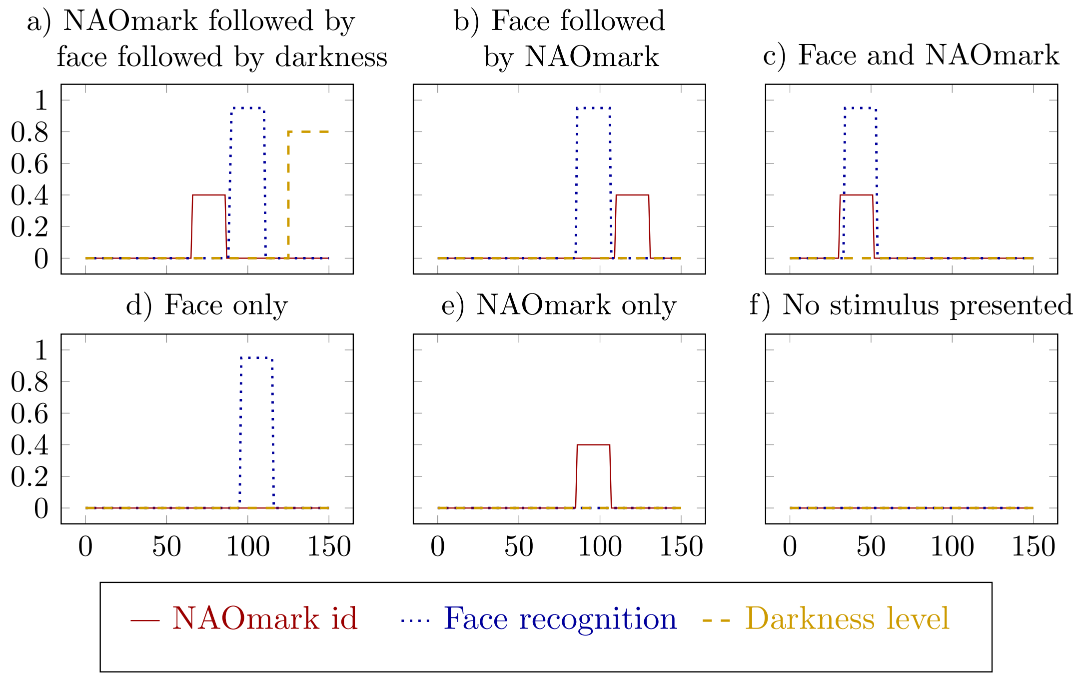


Figure 5.5: Example of situation instances for each of the six situation patterns induced in the experiment. Vertical axis depicts NAO's sensor input after normalization. Horizontal axis depicts the time line counted in numbers of events.

by an aversive stimulus, which in this case is darkness. The CA situation is characterised by the presentation of the NAOmark at good light conditions, followed by the presentation of a human face (demonstrated in Fig. 5.6). Because this pattern is always followed by the presentation of an aversive stimulus, it is then considered to be the pattern of a CA situation. On the other hand, all the other patterns (Fig. 5.5b to Fig. 5.5f) represent safe situations, because they never precede any aversive event.

Observe that some situation patterns, such as the ones in Fig. 5.5b and Fig. 5.5c, are somewhat similar to the pattern of the CA situation in Fig. 5.5a. This has been purposely designed in the experiment, as we desire to verify SAFEL's capability to effectively differentiate safe situations from CA situations, even when the patterns of these situations are similar to a certain extent.

Although duration and delay of stimuli exposition to the robot were similar among data collections, it was not rigorously timed, as it is part of the experiment to evaluate SAFEL's generalisation capability. Besides, in real world cases, situation instances of the same situation pattern may have similar temporal delays, but rarely equal.

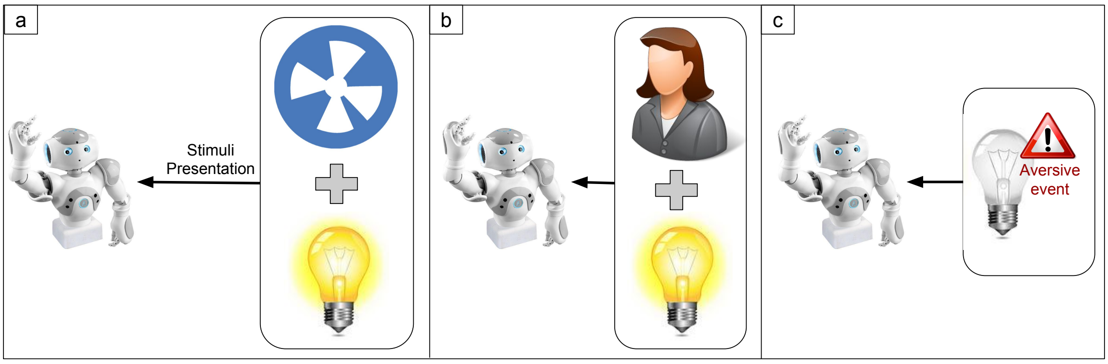


Figure 5.6: Procedure for presenting the aversive event to the robot. (a) Lights are kept on, while a specific NAOmark is presented to NAO for about 5 seconds. (b) With lights still on, the NAOmark is hidden, and then a human face is presented to the robot for about 5 seconds. (c) Both human face and NAOmark are hidden. Light is turned off.

5.4.2 Dataset Generation

We have generated 10 different datasets, which are composed of the situation instances collected through the process explained in Section 5.4.1. The individually collected situation instances were arranged in the datasets according to a specific temporal sequence of situation patterns, which is identical for all the 10 datasets. Fig. 5.7 demonstrates the process for generating the datasets used in this experiment.

To generate a dataset, we randomly selected a situation instance matching the first situation pattern of the chosen temporal sequence and concatenated this situation instance to the dataset. Then we repeated these steps for all the remaining situation patterns in the chosen temporal sequence (Fig. 5.7). Because all sensor noise and failures have been preserved during data collection, a few situation instances may present incomplete or fragmented data. To prevent the temporal positioning of a problematic situation instance from influencing the result, we generated in total 10 datasets base on the same sequence of situation patterns using the above-mentioned method.

Only situation instances with no stimulus presentation (with the pattern of Fig. 5.5f) were reused in the same dataset. Because they are basically the absence of stimulation, situation instances of this pattern are highly similar to each other and, therefore, can be reused without affecting the integrity of the experiment. Situation instances of the remaining patterns (Fig. 5.5a to Fig. 5.5e) were not reused in the same dataset.

Each dataset is equivalent to about 4.5 hours testing and contains 28 unconditioned aversive (UA) situations separated by intervals varying from 2 to 25 minutes

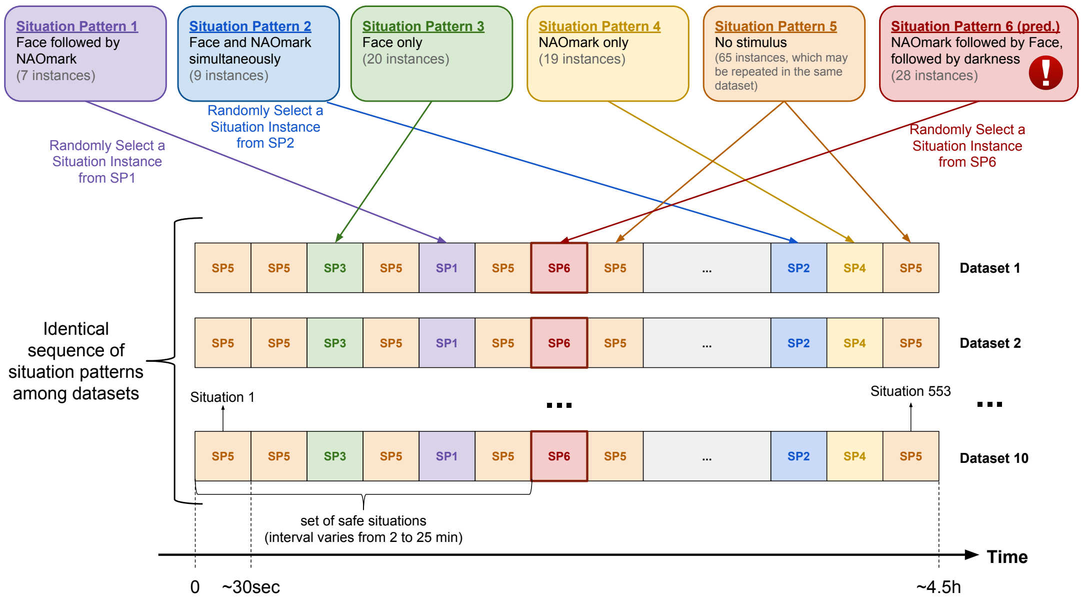


Figure 5.7: Dataset generation process. First, we individually collected a number of situation instances for each of the 6 situation patterns induced in this experiment. Then, for each pattern in the chosen sequence of situation patterns, we randomly select a situation instance of that pattern and concatenate it to the dataset. This procedure was repeated 10 times, so to generate 10 distinct datasets with the same sequence of situation patterns.

representing the set of initially neutral (and potential CA or safe situations), which may comprise any of the situation patterns from Fig. 5.5b to Fig. 5.5f.

5.4.3 Experiment I – Analysing the Hippocampus and Working Memory Modules

This experiment basically evaluates the flexibility of SAFEL's emotional response by analysing SAFEL's capability to generalize similar situation patterns while being able to distinguish situation patterns that are markedly distinct.

We highlight that this experiment focuses on observing the robot's emotional response rather than its behavioural response. In fear conditioning, the behavioural response of an individual is a reflex of its emotional response. The emotional response, in turn, is the most important feedback in order to verify that the individual is under fear, as well as to evaluate the success of fear learning. Thus, in this experiment, we focus on studying the robot's emotional response to different stimulation in order to verify that it can, in fact, learn and predict aversive events based on situational information. A thorough evaluation of SAFEL's influence on the behavioural response of a robot is performed in Chapter 6.

Validation Methodology

The generated datasets have been evaluated according to three factors. The first factor evaluates SAFEL's performance under different pre-defined values of global situation duration (GSD) and situation detection delay (SDD). SAFEL has been analysed for three GSDs: 20 seconds (SDD = 4 sec), 30 seconds (SDD = 6 sec) and 40 seconds (SDD = 8 sec). These GSD values have been selected based on the average time required for the robot to completely observe the induced situation patterns, which is around 30 seconds (± 10 seconds).

The second factor evaluates SAFEL's capability to ignore sensory inputs that are not relevant for predicting the occurrence of aversive stimuli. In this regard, we evaluated SAFEL on two versions of each generated dataset, one with and another without sound sensor input. Since there are no particular patterns in the sound information detected by NAO, it should have a minimal influence in the final prediction. Thus, SAFEL's outcome should be similar for both dataset versions.

Finally, the third factor evaluates the impact of different values of *inter stimulus interval* (ISI) on SAFEL's performance. ISI is the time interval between the offset of the CA situation and the onset of the aversive event. For example, in this experiment, the ISI is the time interval starting right after the presentation of the NAOmark followed by a human face, and ending right before increasing the darkness level of the environment. We have tested three values of ISI: 5, 10 and 15 seconds. The goal of testing different ISIs is to analyse whether the temporal position of relevant events in the CA situation can influence SAFEL's performance.

Considering all dataset generations (10 datasets, 3 ISIs and 2 sets of stimuli input, with and without sound readings) and the 3 GSDs tested, this experiment contains 180 dataset samples in total. All 180 generated datasets were tested independently, and their instances were presented sequentially to SAFEL as if it was being executed in the robot at real time. For each run, we started measuring predictive performance after the classifier had processed the initial 20% of the respective dataset. This decision was made because we assume that the classifier would not have enough samples from each situation type (safe and conditioned aversive (CA)) to create a differentiation among them without learning the initial 20% samples of each dataset.

The generated dataset samples have been divided into groups within each factor that reflect the features under which they are being evaluated. The first factor, which evaluates the influence of different values of the GSD parameter on the classification performance, has been divided into three groups of 60 samples. The first group comprises all dataset samples with GSD = 20 seconds, the second group comprises all samples with GSD = 30 seconds, and the third group comprises all samples with GSD = 40 seconds.

The second factor, which evaluates SAFEL's capability to ignore sensory information that is irrelevant for the prediction, has been divided into two groups of 90 samples. The first group comprises all dataset samples without input from the sound sensor and the second group comprises all dataset samples with input from the sound sensor.

The third factor, which evaluates the influence of different values of ISI on the classification performance, has been divided into three groups of 60 samples. The first group contains all datasets with ISI = 5 seconds, the second group contains all datasets with ISI = 10 seconds, and the third group contains all datasets with ISI = 15 seconds.

We have used the *F-measure* as the performance metric to evaluate SAFEL's efficacy for classifying neutral situations into safe or CA. The F-measure, also known as *F1-score*, is the harmonic mean between precision and recall.

Results

Fig. 5.8 shows SAFEL's performance regarding the three factors previously mentioned, which are GSD (Fig. 5.8a), input set (Fig. 5.8b) and ISI (Fig. 5.8c). In order to study the effects of these three factors on SAFEL's classification performance, we have used the *factorial analysis of variance* (factorial ANOVA), where the null hypothesis states that there is no statistically significant difference in the classification performance among groups within a given factor, and is rejected when $p \leq 0.05$.

Through the ANOVA test, we have analysed the significance of the main effects (i.e., the three factors independently) and of the two-way interactions between factors on the classification performance. The ANOVA test has not found a statistically significant interaction between factors. The test also found no statistically significant difference between groups within the first and second factors, which are GSD and input set, respectively.

This result indicates that there is no significant difference in the classification performance when varying the GSD from 20 to 40 seconds, which reinforces the robustness of SAFEL for situation prediction. It also indicates that there is no significant difference in classification performance between datasets with and without sound sensor input. This demonstrates that SAFEL managed to mostly ignore sound information, as expected. Because sound input had no particular patterns regarding the presentation of aversive stimuli, if SAFEL had significantly considered it for classifying situations into safe or CA, the second group of datasets in Fig. 5.8b could have presented much lower predictive performance.

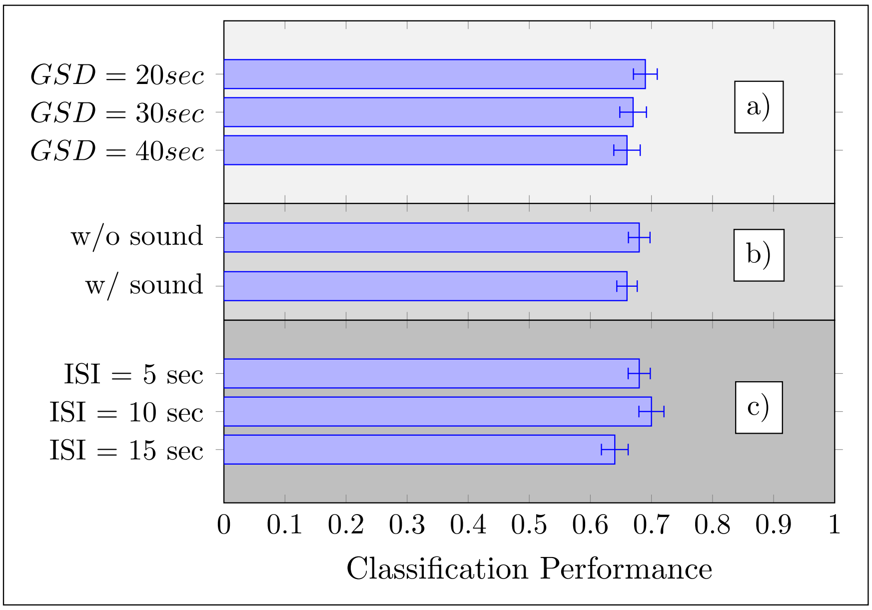


Figure 5.8: Average classification performance (F-measure) among dataset samples per group, where error bars show the 95% confidence interval of the respective group. Each graph shows the results for one of the evaluated factors, which are (a) global situation duration (GSD), (b) input set (with or without sound input) and (c) inter stimulus interval (ISI).

On the other hand, the ANOVA test has found a statistically significant difference in the classification performance among groups within the third factor ($p = 0.0001$), which evaluates the variation of the ISI. However, even though the ANOVA test has found a statistically significant difference among groups, we can observe through the confidence intervals shown in Fig. 5.8c that such difference is minimal. We can assert with 95% confidence level that the (true) performance mean of the three groups in Fig. 5.8c are, respectively, within the intervals [0.66, 0.7], [0.68, 0.72] and [0.62, 0.66]. The closeness of the confidence intervals indicates that, although the ISI can influence the classification performance, such effect is not substantial.

Discussion Part I - Influence of the Events of Interest

In this section, we investigate how the positioning of the events of interest in the CA situation can undermine the classification performance and how it potentially influenced the result observed in Fig. 5.8c. We define *events of interest* in the context of SAFEL's implementation as all events that persistently precede UA situations and are consistently absent in safe situations. Hence, events of interest are the set of events that can provide the most valuable information to differentiate a safe situation from a CA situation. The proper detection and management of

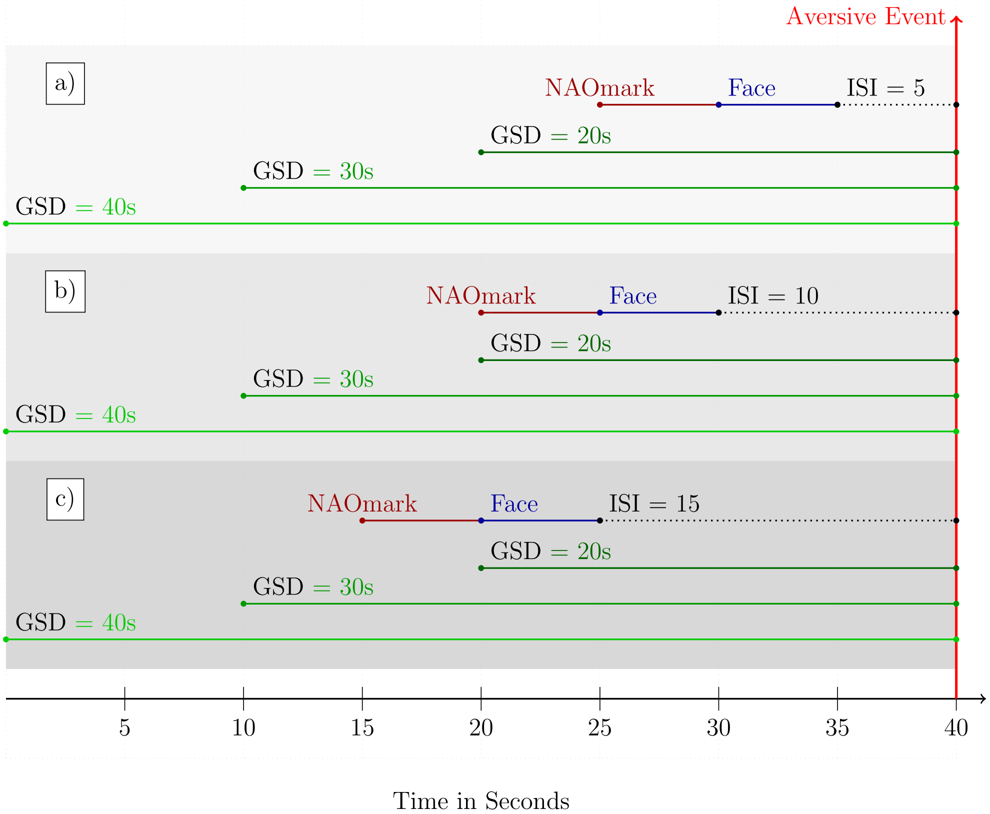


Figure 5.9: Time positioning of the events of interest during CA situations. The diagram shows the possible scenarios considering all combinations of GSD and ISI used in the experiment. Green lines depict the different GSD values (20, 30 and 40 seconds). The different values of ISI are represented by black dotted lines, which are (a) 5 seconds, (b) 10 seconds and (c) 15 seconds. Events of interest are depicted by red and blue lines, which represent the presentation of NAOmark and human face to the robot, respectively.

this information are, therefore, essential for consistently training the classification tree of the WMM.

Fig. 5.9 demonstrates how a particular configuration of ISI and GSD can affect the classification performance. In the performed experiment, the events of interest for predicting the aversive event are the presentation of a NAOmark for about 5 seconds (red lines in Fig. 5.9) followed by the presentation of a human face for about 5 seconds (blue lines in Fig. 5.9). The ISI is represented by dotted black lines, which may have 5, 10 or 15 seconds (Fig. 5.9a, Fig. 5.9b and Fig. 5.9c, respectively). Green lines represent the three tested durations (GSD) of CA situations, which are 20, 30 and 40 seconds.

Observe in Fig. 5.9 that CA situations always contain all events of interest, except when ISI = 15 seconds and the GSD is 20 seconds long (Fig. 5.9c). In this case, the first 5 seconds of the events of interest (i.e., the presentation of

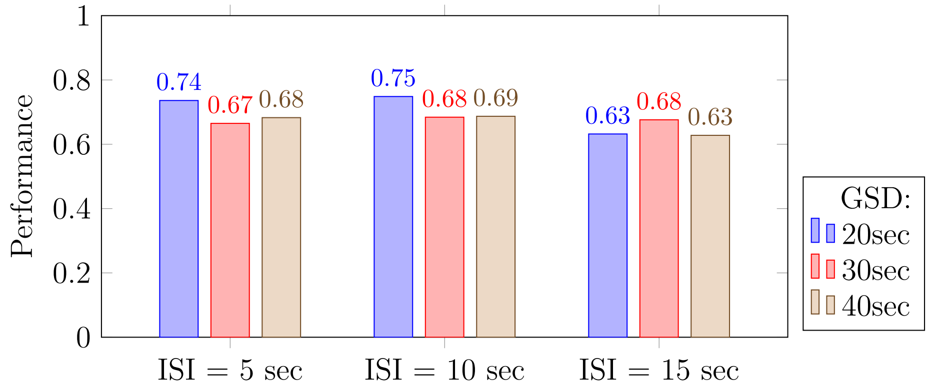


Figure 5.10: Mean classification performance (F-measure) among datasets generated without sound information, grouped by their situation duration and ISI.

the NAOmark) are left out of the CA situation. As consequence, an incorrect pattern of CA situation is used to train the classification tree. Instead of NAOmark followed by face recognition (Fig. 5.5a), the tree is trained to recognized situations with face recognition only (Fig. 5.5d) as CA situations. The problem is aggravated by the fact that some safe situations have the same pattern. As consequence, the tree is trained with inconsistent information, in which the same situation pattern is sometimes presented as safe and sometimes presented as CA. This could explain the difference in classification performance observed in Fig. 5.8c.

Fig. 5.10 shows the average performance for all evaluated datasets without sound input. Note that SAFEL has consistently demonstrated better performance for datasets where GSD = 20 seconds, except when ISI = 15 seconds, case in which we can observe the largest performance decay of the graph. The result of Fig. 5.10 supports the explanation given above, indicating that the problem demonstrated by Fig. 5.9c is indeed the main reason for the discrepancy observed in Fig. 5.8c.

In addition, the higher performance obtained when GSD = 20 seconds (in comparison with the other GSDs values tested) shows that keeping the length of the GSD as close as possible to the length of the events of interest leads to better results (as long as it covers all the events of interest). One can speculate that if the GSD is too large, the classifier may start considering noise from other events (having happened long before the aversive event) that are not part of the events of interest.

In conclusion, the GSD should not be too short, nor too large. The ideal scenario is to have the GSD just large enough to cover the events of interest. A way of tackling this problem is to create a mechanism that allows SAFEL to automatically adjust the duration of situations, which is an improvement that we indicate as future work.

Discussion Part II - Performance Over Time

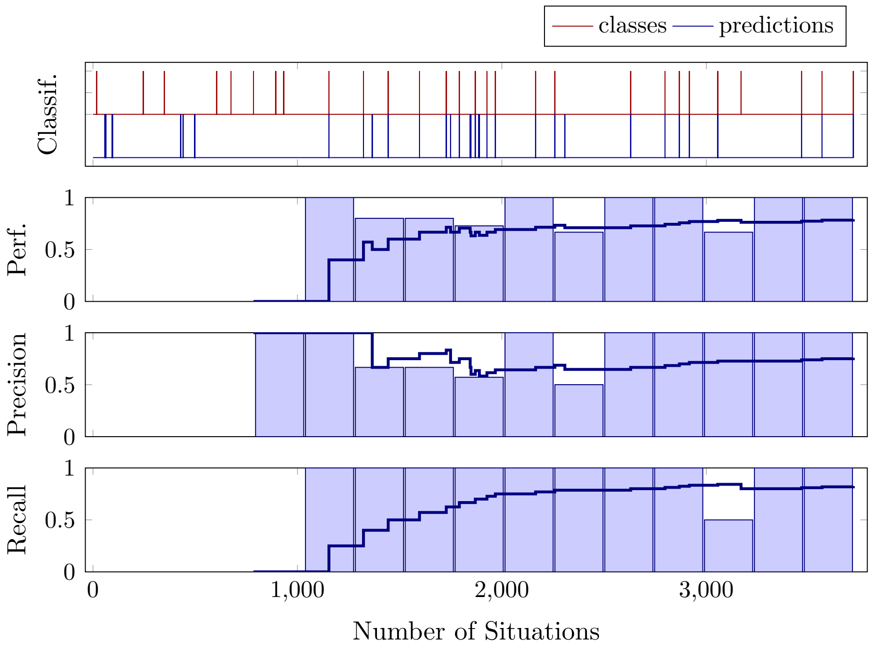
Through SAFEL, the robot learns continuously during its life cycle, thus improving its predictive capabilities with each newly detected stimulus. Fig. 5.11 shows the classification outcome and its performance over time for two of the 180 datasets tested with SAFEL. Fig. 5.11a depicts the most common result among the evaluated datasets and Fig. 5.11b depicts the worst-case scenario. We have generated similar graphs for each of the 180 datasets evaluated, which are available online[1].

Observe in Fig. 5.11a that performance increases as the number of processed situations increases. Classification recall is low for the first third of the detected situations because SAFEL did not predict any of the aversive events happening during that period. Recall improved for the second third of the detected situations, but precision was affected because SAFEL misclassified a few safe situations during that period. However, towards the end, both precision and recall improved as a result of SAFEL correctly classifying most situations in the final third of the dataset.

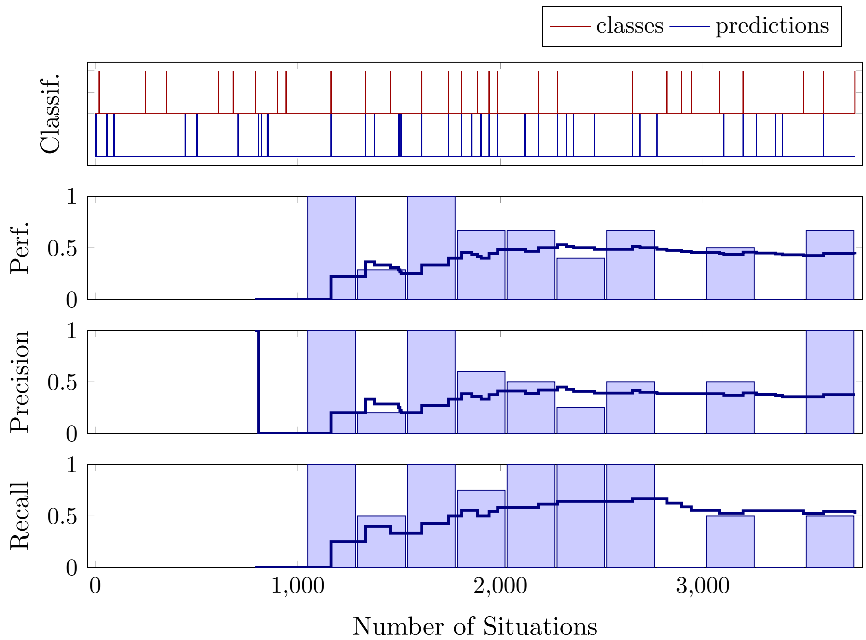
This demonstrates that SAFEL's predictions get more accurate over time. The classification tree starts empty, with no knowledge about the current environment, which explains the low predictive performance in the beginning of the dataset. As the robot experiences different situations, the classification tree is fed with information about the environment and becomes able to provide better predictions. The more experience the robot gains about the environment, the higher the accuracy of SAFEL's predictions.

The learning process described above is ubiquitous in nature. For example, infant animals that have never seen or touched fire before could, by curiosity, naively try to interact with it. After touching it for the first or second time and getting painful burn sensations, they would become afraid of fire and stay away from it in the future. However, note that 'being afraid' of fire is only possible after the animal acquires the knowledge that fire can be harmful through a negative feedback, which in this case is pain. Something unpleasant such as pain, which in this example plays the role of aversive US, must first occur for the learning process to take place; and the more painful experiences the animal has with fire, the bigger its confidence that fire is indeed dangerous. This is an interesting example to demonstrate the two sides of pain. Although unpleasant and potentially debilitating if too intense, pain is an essential negative feedback that helps animals to identify and memorize environmental threats. This learning pattern, in which prediction accuracy improves over time, is reflected in the majority of the experiments that

[1]Other graphs generated in this experiment are available at `https://www.cs.kent.ac.uk/people/rpg/cr519/safel`.



(a) Dataset without sound input, GSD = 30 s and ISI = 10 s.



(b) Dataset without sound input, GSD = 20 s and ISI = 15 s.

Figure 5.11: SAFEL's performance over time for two of the 180 datasets. Figure (a) and (b) show four graphs each. The first graph presents the result of SAFEL's classification: red-line peaks indicate the occurrence of aversive events over time and blue-line peaks indicate SAFEL's predictions for aversive events. The last three graphs show the F-measure, precision and recall of SAFEL's classification over time, respectively. These graphs show two types of over-time measurement: the blue line depicts the cumulative performance over the integral test; the bars depict an 'instantaneous' over-time measurement, where the performance is cumulative only in the interval comprised by the respective bar.

we have performed with SAFEL, and the speed with which performance improves varies among datasets.

Fig. 5.11b shows an example of the performance over time when CA situations happen to miss part of the events of interest. In our case, this happens when GSD = 20 seconds and the ISI = 15 seconds, as previously explained and demonstrated with Fig. 5.9. Fig. 5.11b shows a slow and modest performance improvement over time, which decays after 2000 situations. In addition, classification precision is poor from the beginning to the end of the experiment due to the large number of safe situations classified as CA situations. As previously mentioned (see Fig. 5.9), this is because the tree is being trained with inconsistent information, where the same situation pattern is sometimes presented as safe and sometimes presented as CA. Therefore, in this case, the classification tree has no basis for providing an accurate prediction.

The experiments have demonstrated that, as long as all events of interest are captured by the CA situations, the actual duration of these situations (i.e., the GSD), as well as their ISI, do not meaningfully influence the classification performance. This means that SAFEL is capable of adapting to different temporal characteristics without performance decay. In addition, Fig. 5.8 shows that, although all sensor noises and detection failures have been preserved, SAFEL was capable of predicting aversive events based on situational information with 67% of classification performance (F-measure) on average.

5.4.4 Experiment II – Improving the Predictive Performance of SAFEL

Among others, the situation detection delay (SDD) and global situation duration (GSD) are parameters of SAFEL that must be predefined by the user, as explained in Section 4.4. The GSD defines a fixed duration for neutral situations, whereas the SDD defines the period of time between the activation of a given situation and the activation of its predecessor situation.

The value of the GSD and the SDD can highly influence the performance of the classification tree in the WMM. For example, suppose two subsequent situations S_1 and S_2. If $SDD > GSD$ (Fig. 5.12a), which implies that $d_1 < a_2$, then the stimuli information in between the time stamps d_1 and a_2 will not be collected by the HM and, consequently, will not be sent to the WMM for learning and prediction. Learning this piece of information could be important for the robot to accomplish its task, and thus should not be ignored.

Now, suppose $SDD = GSD$, which implies that $d_1 = a_2$, as seen in Fig. 5.12b.

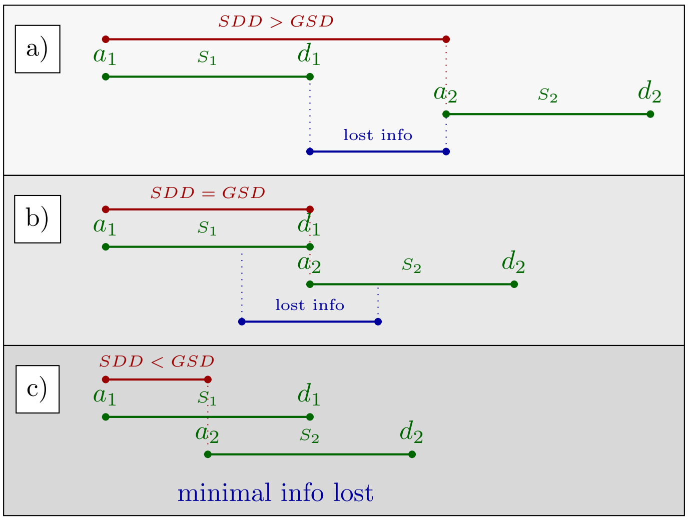


Figure 5.12: Effect of different sizes of situation detection delay (SDD) on the loss of situation information.

Even in this case, there is still some information being ignored. The WMM will be able to learn the pattern of situations S_1 and S_2, but any sequence of events starting after a_1 and finishing before d_2 will not be learned. Sequences of events in between these two time-stamps could be forming the pattern of a CA situation. Therefore ignoring this information could undermine the robot's ability to predict aversive situations.

Hence, it is important that $SDD < GSD$, as shown in Fig. 5.12c, so to minimize the loss of potentially essential information. However, defining how small the SDD should be in relation to the GSD is still an open issue. Essential information could be lost if the SDD is too large. If it is too small, then unnecessary redundancy could be introduced to the system, possibly reducing its response time.

In the first versions of SAFEL, both the GSD and the SDD were pre-defined parameters of SAFEL. While defining the GSD value is fairly intuitive and can be easily induced from the problem the robot has to solve, finding an ideal SDD value is a complex task. For example, suppose an elderly-care robot. One can induce that the set of events relevant for the robot to predict that an elder person sitting in a chair is going to stand and walk may occur between 10 to 30 seconds before that action actually takes place. On the other hand, the set of events relevant for predicting which room of the house that person intends to visit may occur between one to two minutes before the person actually reaches that room. In this case, one can estimate the GSD based on fairly stable factors that can be easily observable during a few trials, such as the person's walking speed, the design of the house, in which room the person is and to which room he/she is moving to. However, one cannot construct a similar reasoning nor make simple observations to estimate a satisfactory value for the SDD.

The goal of this experiment is to turn the SDD into an internal parameter of SAFEL, which shall be calculated based on the value of the GSD. This calculus shall take into consideration the result from an empirical study to find the best quantitative relation between GSD and SDD in order to achieve the highest predictive performance possible from SAFEL. By doing so, we aim at reducing the complexity of SAFEL's pre-configuration while increasing its predictive performance.

Validation Methodology

The habit of overestimating danger is ubiquitous and essential in nature, as the cost of underestimating a danger is usually much higher than that of overestimating it (LeDoux 1999). The same rule may apply to robots, as they inhabit our physical world and may face similar threats. Thus, it is of our interest to take into consideration SAFEL's capability to mimic natures tendency to overestimate danger. For this reason, we have used the *F2-score* as the performance metric to evaluate SAFEL's efficacy for classifying neutral situations into safe or conditioned aversive (CA).

The F2-score is a modified version of the *F1-score* (or *F-measure*) that gives more importance to recall (i.e., the fraction of aversive events that were predicted) than precision (i.e., the fraction of correctly predicted aversive events). While the F1-score is defined as the harmonic mean between precision and recall, the F2-score gives twice the weight to recall in comparison to precision.

We have tested three different values of ISI. Consequently, three different versions of dataset have been created for each dataset generated according to the methodology explained in Section 5.4.2. This has been done by varying the ISI between 5, 10 and 15 seconds. Lastly, we ran SAFEL on each dataset independently, during which the instances of the datasets were presented sequentially to SAFEL as if it was being executed in the robot at real time.

Every dataset has been tested with three different values of GSD: 20, 30 and 40 seconds. As explained in Section 5.4.3, these GSD values have been selected based on the average time required for the robot to completely observe the induced situation patterns, which is around 30 seconds (± 10 seconds). For each dataset, predictive performance started to be measured after SAFEL had processed the initial 20% of its instances, which we assume to be the minimum amount of samples necessary for the classification tree to create a distinction between each situation type (safe and CA).

We have tested nine SDD values, which were defined as a percentage of the corresponding GSD. For each combination of dataset configuration (3 ISIs and 3 GSDs), we tested an SDD equals 10% of the GSD, an SDD equals 20% of the GSD, and so on, until 90% of the GSD.

Results

Fig. 5.13, shows the median and percentiles of predictive performance for each SDD tested. It is clear from Fig. 5.13 that a higher performance is obtained when the SDD is in between 10% and 30% of the GSD.

We used the *factorial analysis of variance* (ANOVA) to study the effects of different SDDs in SAFEL's predictive performance, where the null hypothesis states that there is no statistically significant difference in the predictive performance among different SDDs, and is rejected when $p \leq 0.05$. The ANOVA test found statistically significant difference in performance ($p \approx 0$) when comparing SDDs smaller and bigger than 30%. This result can be observed in Fig. 5.14, which compares the means of predictive performance by SDD. Fig. 5.14 shows that better predictive performance is obtained when the SDD is 20% of the GSD, and the difference in performance is statistically significant when compared with SDDs ranging from 40% to 90% of the respective GSD. However, the difference is not statistically significant when comparing SDDs between 10% to 30% of the GSD.

When observing the interactions between the different values of SDD and ISI

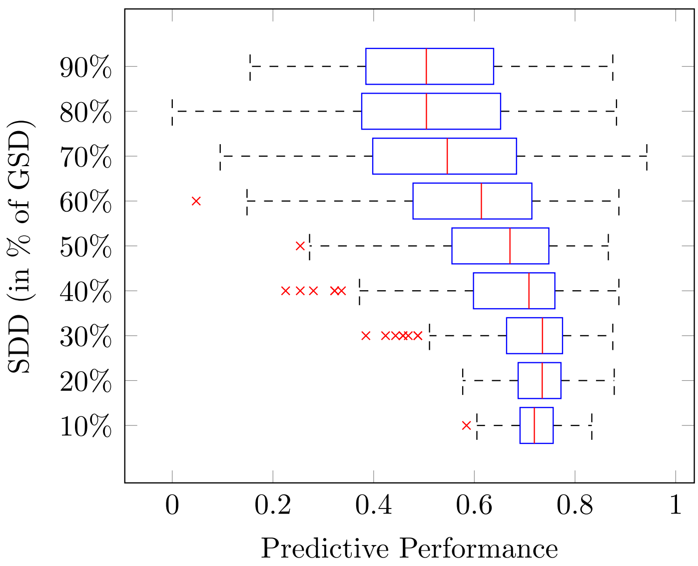


Figure 5.13: Boxplot of the predictive performance of SAFEL by SDD. On each box, the central red mark is the median, the edges of the box are the 25^{th} and 75^{th} percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted as red marks.

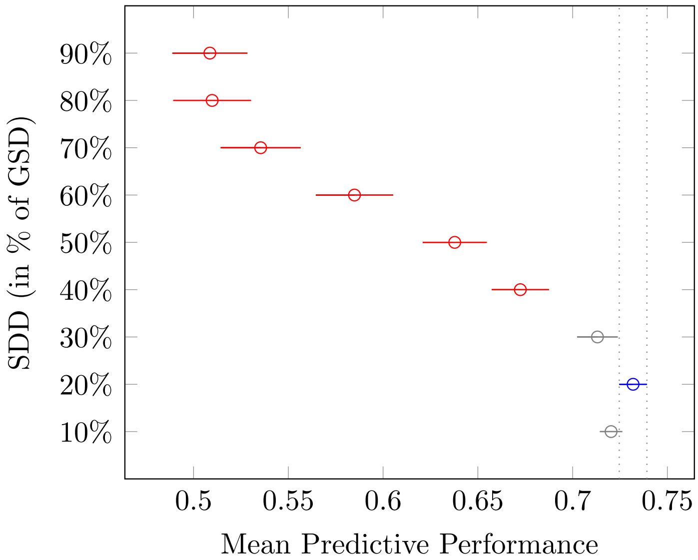


Figure 5.14: Comparison of performance means by SDD. Each performance mean is represented by a mark, and their respective 95% confidence intervals are represented by error bars. Two performance means are significantly different if their intervals are disjoint. The highest performance mean is highlighted in blue. For the remaining performance means, it is represented in red if there is a statistically significant difference from the highest performance, otherwise they are represented in grey.

(Table 5.1a), we found a statistically significant difference indicating that, regardless of the ISI value (5, 10 or 15 seconds), best performance is still mostly obtained when SDD ranges from 10% to 30% of the respective GSD. On the other hand, when analysing the interactions between the different values of SDD and GSD (Table 5.1b), we found some influence of the GSD in the performance. The ANOVA test indicated that GSD = 20 seconds yields statistically significantly better performance if compared with the other tested values of GSD (30 and 40 seconds). However, in our experiment, this effect occurred only when SDD is between 10% and 50% of the respective GSD. Also, the performance is consistently better when SDD = 30 seconds in comparison to SDD = 40 seconds, though the difference is not statistically significant. This result reinforces the hypothesis discussed in Section 5.4.3 that keeping the length of the GSD as close as possible to the length of the events of interest leads to slightly better results, thought the difference in performance is not substantial.

In conclusion, this result clearly indicates that it is worth fixing the SDD value as 20% of the GSD in terms of predictive performance.

Table 5.1: Analysis of the effect of variables' interaction in the predictive performance. The underlined number indicates the highest mean predictive performance. Bold numbers indicate values of predictive performance that have no statistically significant difference from the highest performance.

(a) SDD × ISI.

		ISI (sec)		
		5	10	15
SDD (% of GSD)	10%	**0.711**	**0.720**	**0.730**
	20%	**0.716**	**0.733**	**0.746**
	30%	0.657	**0.728**	<u>**0.754**</u>
	40%	0.571	0.706	**0.741**
	50%	0.521	0.665	**0.727**
	60%	0.431	0.625	0.699
	70%	0.375	0.566	0.665
	80%	0.366	0.525	0.638
	90%	0.371	0.527	0.628

(b) SDD × GSD.

		GSD (sec)		
		20	30	40
SDD (% of GSD)	10%	**0.753**	0.719	0.689
	20%	<u>**0.775**</u>	0.721	0.700
	30%	**0.773**	0.704	0.662
	40%	**0.756**	0.665	0.597
	50%	**0.728**	0.605	0.580
	60%	0.704	0.530	0.521
	70%	0.658	0.474	0.474
	80%	0.616	0.466	0.448
	90%	0.601	0.440	0.485

5.4.5 Experiment III – A Comparative Analysis with BEL

As discussed in Chapter 2, the Brain Emotional Learning (BEL) model (Morén and Balkenius 2001) is among the most cited models of artificial fear conditioning in the literature, especially regarding its applications in real-world robotics and engineering tasks. The BEL model consists of interconnected modules of ANNs that, similarly to SAFEL, simulate the role of neural circuitries involved in fear learning. It receives two types of inputs: environmental neutral stimuli and a reward signal; and outputs an emotional response.

In terms of predictive performance, we understand that comparing BEL (Morén and Balkenius 2001) and SAFEL with a focus on temporal reasoning would be unfair, because unlike SAFEL, BEL is not designed to process temporal sequences of events. Although BEL is considered a related work and has similarities with SAFEL, these are mostly conceptual, such as the fact that both models are inspired by real brain mechanisms.

Unlike SAFEL, BEL would be unable to successfully predict the occurrence of aversive events with the experiment configuration proposed in this section. As evidence for this argument, we have compared the predictive performance of BEL on our dataset, generated as explained in Section 5.4.1 and Section 5.4.2, and on a simpler dataset where the prediction of the aversive stimulus does not depend on the temporal relationship of environmental stimuli. We have obtained the implementation of the BEL algorithm from Lotfi and Akbarzadeh-T. (2014b)[2].

[2]The source code of the BEL algorithm implemented by Lotfi and Akbarzadeh-T Lotfi and Akbarzadeh-T. (2014b) is available at `http://bitools.ir/projects.html`

Table 5.2: Comparison of the result from running the algorithm[2] proposed by Lotfi and Akbarzadeh-T. (2014b) with the dataset that they have provided as example[3] and one of our datasets generated as explained in Section 5.4.1 and Section 5.4.2. The second and third columns show the fraction of true positives for the first and second classes of the corresponding datasets, which in our dataset depict 'CA situation' and 'safe situation' respectively.

Dataset	TP Class 1	TP Class 2
Authors' dataset	69.7%	62.5%
Our dataset	0.0%	100%

The performance of the BEL algorithm has been evaluated on our dataset and on the dataset[3] used by Lotfi and Akbarzadeh-T. (2014b), whose classes indicate the presence or absence of heart disease in a patient based on 13 attributes describing that patient's condition (age, sex, chest pain type, blood pressure, etc.). Table 5.2 compares the results.

Table 5.2 clearly shows a good performance when running the BEL-based algorithm with their own example dataset, which correctly predicts 69.7% of the instances in the first class, and 62.5% of the instances in the second class. However, the performance of the BEL-based algorithm is drastically poor when it is executed with one of our datasets, as none of the situation instances have been classified as aversive. By 'predicting' that all instances were safe situations, the BEL algorithm managed to correctly predict 100% of the safe situations. This is a clear indication that the BEL classifier was not capable to learn with data samples representing aversive situations. As a consequence, the BEL classifier failed the main goal of this experiment, which is to warn the robot of imminent environmental threats. If the classifier predicts only safe situations, then no aversive situations will be ever predicted and the robot will never be given the chance to act in advance to prevent them.

This result is a consequence of the different ways in which SAFEL and BEL observe sensory data in order to output predictions, as seen in Fig. 5.15. SAFEL looks back in the temporal line and observes the pattern of data for a period of time before the detection of the aversive stimulus. The BEL algorithm, in turn, considers only the pattern of data that co-occurs with the aversive stimulus, which is analogous to the task performed in the Amygdala Module (AM) of SAFEL. However, in our dataset, the pattern of neutral stimuli that co-occurs with the aversive stimuli is identical to the pattern of stimuli occurring in many other events where the aversive stimulus is absent. Thus, a dataset containing sensor

[3]This dataset and its complete description are available at `http://archive.ics.uci.edu/ml/datasets/heart+Disease`.

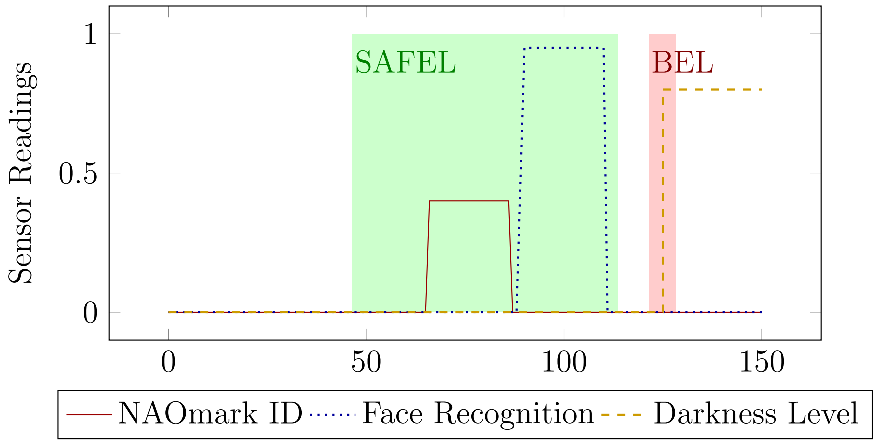


Figure 5.15: Comparison of stimuli analysis by SAFEL and BEL. The vertical axis depicts NAO's sensor input after normalization. Horizontal axis depicts the time step. The green and red areas indicate the portion of sensory information considered by SAFEL and BEL, respectively, at the moment that the robot detects an aversive stimulus (which is darkness in this case).

readings like the one seen in Fig. 5.15 is inconsistent for BEL and leads to poor performance. On the other hand, the same dataset leads to high performance for SAFEL, as demonstrated in the experiments of Section 5.4.3 and Section 5.4.4.

One way to deal with BEL's poor performance is to pre-process our dataset using the HM and part of the WMM of SAFEL, and then deliver to BEL the compacted version of situation instances created in the feature extraction phase performed in the WMM. We believe that this would greatly improve the accuracy of BEL's prediction, but by doing so we would have more than half of data processing performed by SAFEL. It would be analogous to comparing SAFEL with itself, however replacing the classification tree in the WMM with BEL, therefore not configuring a proper comparative study.

In addition, further differences between the two models also include:

1. An optional pre-training phase may be conducted with SAFEL. However, this is not mandatory, as SAFEL is able to learn environmental threats at runtime without any prior knowledge besides the set of predefined aversive unconditioned stimulus (US). On the other hand, BEL is a supervised learning algorithm and requires a training step with many epochs and then a test phase. This also reflects in the learning time, which required more than an hour for BEL with our dataset, while requiring only about 1.5 minutes in average for SAFEL.

2. BEL is mostly used as a controller for industrial and engineering purposes, being usually compared in terms of predictive performance with traditional controllers in the industry such as PID. SAFEL, on the other hand, aims at

complex goals and tasks, such as providing robots with human-like emotional responses, adaptive capabilities and flexible decision making at execution time.

5.5 Final Considerations

This chapter presented the design and implementation of the Working Memory Module (WMM), which is the module of SAFEL responsible for associating the contextual memories formed in the Hippocampus Module (HM) with their emotional meaning given by the Amygdala Module (AM). We have discussed the two processes taking place in the WMM for creating associations between context and 'fear'. The first process is a feature extraction, which selects only the most relevant characteristics of situations' temporal patterns. This feature extraction step helps reducing redundancy and dimensionality of data, which consequently contributes to preventing overfitting of data. It also allows us to generate a unitary representation of context that is similar to its biological counterpart concept discussed in Section 4.1. This process is based on extracting three features from each stimulus, which compose the set of features that we consider to better capture the main temporal characteristics of stimuli variation over time. However, a proper study to increase the number of extracted features is needed, as well as to investigate which features better represent the temporal behaviour of stimuli. We indicate such investigation as future work.

The second process taking place in the WMM is the actual association between context and emotion, which is performed using a binary classification tree. A number of reasons led us to use a classification tree for this task, including that it is easy to interpret, fast to train, nonparametric and implicitly performs feature selection. The compacted versions of situation information generated through the feature extraction process are delivered along with their emotional category to the classification tree for learning and prediction. Safe and conditioned aversive (CA) situations are used to retrain the classification tree, while neutral situations are used for predicting the occurrence of imminent threat. The classification tree does such predictions by matching the temporal pattern of neutral situations with the patterns of previously learned CA situations.

In this chapter, we have also presented three preliminary experiments performed with the Hippocampus and Working Memory modules of SAFEL. The first experiment, presented in Section 5.4.3, aimed at evaluating the predictive performance of the first version of the HM and WMM together. This experiment demonstrated that SAFEL is capable to warn the robot about the imminent occurrence of aversive events based on information that is contextual and emotional at the same time. In addition, these predictions are flexible in the sense that the robot is capable to recognize similarities in the patterns of different situations while being capable to distinguish situation patterns that are markedly distinct.

The key difference between SAFEL and other models of fear learning (see Chapter 2) is that these predictions, which warn the robot controller of imminent threats, is not only based on the relevance of individual stimuli (which is processed in the AM), but also on complex temporal and contextual information. Other models of fear learning have been proposed to date that can predict the imminent occurrence of threats, as discussed in Chapter 2, but these usually either ignore the relationship between multiple stimuli or the temporal behaviour of stimuli. Therefore, they have restricted applicability, as they would only work in scenarios where the aversive stimulus accompanies very abrupt and punctual changes of one single stimulus in the environment. In most cases, those models that take into consideration the temporal behaviour of stimuli do not allow the customization of the time interval comprising the temporal analysis (which in SAFEL is made possible by the GSD parameter), limiting the model's usability and solution scope.

The second experiment, discussed in Section 5.4.4, aimed at improving SAFEL by investigating the best quantitative relation between two parameters of SAFEL, the global situation duration (GSD) and the situation detection delay (SDD), in order to achieve the highest predictive performance possible. This study was later used to fixate the SDD value based on the GSD parameter value, consequently turning the SDD into an internal parameter of SAFEL. By doing so, we managed to reduce the complexity of configuring SAFEL's pre-defined parameters while increasing its predictive performance.

Finally, the third experiment, presented in Section 5.4.5, briefly compared the outcomes of SAFEL and BEL with the datasets used in the experiments of Section 5.4.3 and Section 5.4.4. This is not a thorough comparative study, as its sole goal is to prove that BEL is unable to successfully predict the occurrence of aversive events with the experiment configuration proposed in Section 5.4. Unlike SAFEL, BEL is not designed to process temporal sequences of events. Therefore, we focused on demonstrating that BEL succeeds well with simpler datasets that do not involve complex temporal relationships between stimuli while performing poorly with our experimental setup, where SAFEL demonstrated high predictive performance.

Chapter 6

Case Study: Robot Soccer

This chapter explores the application of SAFEL in the robot soccer context. The analysis conducted in this case study evaluates SAFEL under three different perspectives: the predictive performance, the robot's adaptation performance and how learning evolves at runtime inside SAFEL's modules. Additionally, this chapter also contributes to answering all the three research questions formulated in Section 1.4 by evaluating SAFEL as a whole (and consequently all the three approaches comprising its hybrid architecture) under all the requirements of a situation-aware intelligence (Section 1.3.1) and of an emotional intelligence (Section 1.3.2) in a practical and highly dynamic robotic application, which is the RoboCup competition.

We briefly introduce the RoboCup competition in Section 6.1 and describe the scenario of the case study in Section 6.2. Section 6.3 describes the case study in terms of goals, scope and experimental setup. Finally, results are presented in Section 6.4 and further discussed in Section 6.5.

6.1 The Robot World Cup

Robot World Cup (RoboCup) is an important international scientific initiative with the goal to advance the state of the art of artificial intelligence for autonomous robots by proposing an ambitious challenge. The official challenge of the RoboCup initiative, established in 1997, states that "by the middle of the 21st century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup." (Ferrein and Steinbauer 2016).

The relevance of the RoboCup competition is not on the challenge itself, but on the intrinsic gains from the journey to accomplish such a goal. RoboCup's initiative poses a challenge of high complexity that requires a significant body of research in the areas of artificial intelligence, sensor fusion, real-time planning and navigation, cooperation in multiagent robotics, context recognition, image processing, motor control, among others (Kitano et al. 1998).

For this reason, RoboCup has been considered to be both a *landmark project* and a *standard problem* (Kitano et al. 1998). A *landmark project* is any project aiming to accomplish a highly appealing and exciting goal, capable of capturing the interest of a broad and varied public. Perhaps, the best-known example of a successful landmark project is the Apollo space program, which aimed at the goal of landing a man on the moon and returning him safely to earth. The appeal of such a goal captured the interest of industries, researchers, as well as the general public, leading to a historical technological breakthrough.

On the other hand, *standard problems* are those in which the development of a solution implies on the evaluation of a number of theories, algorithms and architectures, thus contributing to advance the state of the art in various domains. Human-computer chess matches are well-known examples of a standard problem, having greatly contributed to the state of the art of search algorithms. Attempts at programming computers to play chess started in the early 1970's and had its most famous success with the victory of IBM's Deep Blue, a chess-specialised computer, over then world's chess champion Garry Kasparov, in 1997 (Newborn 2012). Nowadays, chess-playing programs running in general purpose mobile phones have been able to win over strong human players in international competitions.

When comparing with the human-machine chess challenge, RoboCup's initiative poses a challenge of higher complexity, involving significant advancements in the areas of artificial intelligence, sensor fusion, real-time planning and navigation, cooperation in multiagent robotics, context recognition, image processing, motor control, among others (Kitano et al. 1998). The RoboCup initiative has hosted annual competitions for more than 20 years now, a period over which significant advancements have been achieved towards autonomous robotics. Such advancements, in turn, led a considerable number of rules in the competition to be reviewed and rigidified over the years, making the competition environment more realistic.

The RoboCup competition is split into five leagues, each focused on advancing different aspects of robot soccer. These are the *Soccer Simulation League*, the *Small-Size League*, the *Middle-Size League*, the *Standard Platform League* and the *Humanoid League*. In this case study, we are interested in the advancements of the *Standard Platform League* (SPL), which relies on equal robot hardware for all teams so that these can solely focus on developing control algorithms for humanoid robots. The current standard platform used in the SPL competition is the humanoid robot NAO, developed by SoftBank Robotics (2017).

Despite RoboCup's many achievements towards a number of research fields related to autonomous robotics, the development of contextual perception and flexible decision making has made modest progress. Ferrein and Steinbauer (2016) mention that stable solutions for the robots' behaviour exist, though fundamental perception and decisional problems have not yet been solved. They also emphasise that according to Itsuki Noda, current president of the RoboCup Federation, future advances in the RoboCup will move towards the flexible interaction between robots and humans, through the development of AI able to understand and react to 'intentions'. The ability to handle intentions in a flexible way is essential mainly when considering the ultimate goal of the RoboCup initiative, in which a team of robots shall play against a team of humans.

This case study investigates whether SAFEL can contribute towards filling the current gap in context awareness and flexible decision making in RoboCup's SPL competition. SAFEL has been tested and evaluated in a particular potential scenario within an SPL match where the goalkeeper is required to understand and adapt to the different behaviour profiles of distinct opponent teams in order to take custom and advantageous decisions.

6.2 Scenario of the Case Study

Teamwork, pre-coordination and collaborative behaviour are the main focus of most research related to robot soccer (Nitschke 2005; Genter et al. 2016; Whiteson et al. 2003). These are undoubtedly crucial in soccer, but the effectiveness of teamwork strategies is limited to the skill level of individuals in a team. A good cooperation strategy is of little aid if the team members are unqualified. In addition, despite the intrinsic team-work nature of soccer, there are many situations in which individual players need to rely on their own skills and decision-making capabilities, often when they find themselves isolated from the rest of their teams. For this reason, we argue that mechanisms to improve the adaptation skills and flexible decision-making of individual players are also essential for robot soccer. This is, however, a neglected area of study in RoboCup (Rizzi, Johnson and Vargas 2017).

We propose the following scenario in RoboCup's SPL (Fig. 6.1): suppose a match between team T_1 and team T_2, where team T_1 is currently attacking. Now suppose that a defender from team T_2 manages to take possession of the ball and switch fields (i.e., pass the ball from one side of the field to the other in one shot). Because team T_1 was fully engaged in the attack, all members of team T_1 are in team T_2's side of the field, except the goalkeeper. Also, second by the goalkeeper from team T_1, the striker from team T_2 is the closest player to the ball at this

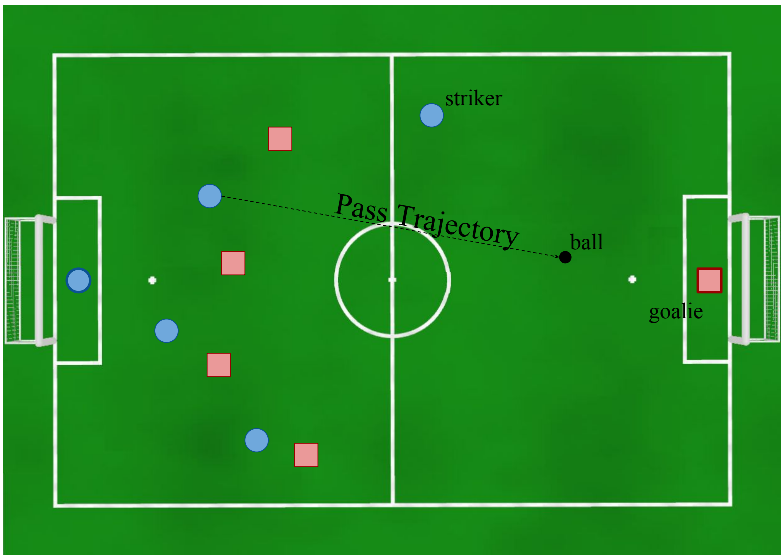


Figure 6.1: Scenario of the case study. Red squares represent players from team T_1, blue circles represent players from team T_2 and the black circle represents the ball. The circle and square with thicker border line represent the respective team's goalkeeper.

moment. The ball stops closer to the goalkeeper than to the striker, but far enough so that the goalkeeper would have to leave the goal area vulnerable in order to pursue the ball.

In this hypothetical scenario, the goalkeeper is isolated from the rest of its team and is forced to rely only on its own judgement and skills. The striker from team T_2 will certainly reach the ball before the other players of team T_1 unless the goalkeeper intervenes. The decision to be taken by the goalkeeper is, therefore, whether to intervene or not. Intervening requires leaving the goal area unattended and, consequently, vulnerable. On the other hand, not intervening would give an obvious advantage to the opponent striker for a clear shot to goal.

The answer to this question is not straightforward, as it depends on the playing profile of the opponent team. If the opponent striker has a weak shot, for instance, then it will likely need more than one kick to attempt a goal, giving team T_1 time to retreat and aid in the defence. Also, if the striker's first kick is weak, then the ball will consequently be even closer to the goalkeeper, making it possible to reach the ball without completely abandoning the goal area. Thus, in this case, remaining in the goal area and waiting for help is a wiser decision.

On the other hand, if the striker has a strong shot and good aim, it may be worth risking the goal by trying to reach the ball first, since staying in the goal area would not reduce the likelihood of team T_2 scoring a goal. At first, it may seem a trivial problem that depends only on the strength of the striker's shot. However, there are many factors involved, including the distances between the elements of interest in this situation (i.e., the striker, the goalkeeper, the goal and the ball).

The kick that is weak from a particular position in the field, may be enough to score a goal from another position if we consider the angle and distance between the ball, the goal and the goalkeeper. The problem can become even more complex by increasing the number of undesirable outcomes to be avoided by the goalkeeper. For instance, in the above example, goals are the only outcome to be avoided. However, if we add collisions as another undesirable outcome, the behaviour profile of opponent teams will diverge even more, and the sequence of events leading to the undesired outcomes (goals and collisions) will become even more complex.

The example described above composes the scenario used for all experiments in this case study. The scenario and its implementation process are formally described next, in Section 6.3.

6.3 Experiments

To implement the proposed scenario, we used the B-Human's robot soccer controller and simulation tool (Röfer et al. 2015). The B-Human team is currently among the best teams in the RoboCup SPL, having won the world championship five times, the RoboCup German Open seven consecutive times and the RoboCup European Open once. We evaluate and compare the outcome of the scenario described in Section 6.2 for two goalkeeper behaviours: the first is the default behaviour implemented by the B-Human team and the second is a similar behaviour, but taking into consideration SAFEL's emotional responses.

In the following sections, we describe the goals and scope of this experiment, as well as the experimental setup. Videos of the simulations and complementary material for the experiments described in this chapter are available online[1].

6.3.1 Goals of the Case Study

In the scenario described in Section 6.2, the default behaviour of the goalkeeper (as implemented by the B-Human team) is to leave the goal area to pursue the ball whenever it is within a particular distance from the goal area. In any other cases, the goalkeeper would stay and guard the goal. Therefore, when deciding whether to leave the goal area, the goalkeeper does not take into consideration any other factors besides the ball position in relation to the goal area.

By using SAFEL, we intend to expand the area in which the goalkeeper is willing to leave the goal area to pursue the ball. However, instead of creating a simple

[1] Complementary material for the experiments discussed in this chapter are available at `https://www.cs.kent.ac.uk/people/rpg/cr519/casestudy`.

rule based on the position of the ball in the field, we expect the keeper to learn with experience when it is safe or not to leave the goal unattended. Such decision would take into consideration the distances between the elements of interest in the field (i.e., the goalkeeper itself, the opponent striker, the defending goal and the ball), the identity of the opponent team and how aggressive and skilled is its striker.

More importantly, we expect the final decision in regards to leaving or not the goal area to be based not only on the stimuli information but also on their inter- and temporal relationship. This is due to SAFEL's capability of integrating stimuli information and temporal relationship in a unified representation of context.

With this case study, we intend to demonstrate:

- That SAFEL is capable of providing fear learning capabilities at stimulus and situational level.
- That SAFEL's emotional feedback can be used to aid in environmental adaptation and flexible decision-making.
- That SAFEL's emotional feedback can improve the intelligent behaviour of robots in the robot soccer context.
- The importance and influence of all modules of SAFEL in its final emotional response.

It is out of this case study's scope to define or investigate which behaviours should be triggered in response to SAFEL's emotional feedback to maximise the goalkeeper's positive outcomes. Although we select some behaviours for our experiments' purposes, we consider that investigating the ideal behaviours for a particular scenario is a task better undertaken by the designers of the robot controller, who are familiar with the specifics of their robots and controllers, as well as with the particular features of their robots' environments and tasks.

It is also out of this case study's scope to evaluate SAFEL's performance in regards to computation time. During the first versions of SAFEL's implementation, we focused solely on ensuring SAFEL's efficacy for predicting future aversive events and responding accordingly, mainly for practical usage in robotics. However, we recognise that optimising computation time is crucial for SAFEL, since it is aimed to be executed at runtime, and we indicate improvements in this direction as a future work.

6.3.2 Experimental Setup

For this case study, we have defined four opponent teams (A, B, C and D) to play the role of team T_2 described in Section 6.2, each exhibiting a distinct striker behaviour. The behaviour of each team's striker is summarised in Table 6.1, whose characteristics are described as follows:

- *Aggressive Ball Pursuit*: indicates whether the striker is inclined to prioritise the ball pursuit so to neglect external factors, such as other players in its way, increasing the occurrence of collisions. Strikers that are not aggressive in their ball pursuit behaviour will take into consideration other players' position in the field, hesitating in its chase for the ball whenever it may lead to collisions.

- *Opportunistic Rebound*: indicates whether the striker is inclined to approach the ball for rebound (after a failed goal attempt), regardless of how close the ball is to the goalkeeper, possibly increasing the occurrence of collisions and goals. Strikers that are not inclined to opportunistic rebound will hesitate in approaching the ball for rebound if it is too close to the goalkeeper.

- *Strong Shot to Goal*: indicates whether the striker has a stronger than average kick when attempting to score a goal, likely increasing its chance of actually scoring. Strikers with average kick strength may still score goals, but less frequently than strikers with a strong shot.

Fig. 6.2 shows the areas of interest in the soccer field for the purposes of this case study. Area A1 shows the goal area, where the goalkeeper usually stands under normal circumstances. Area A2 shows the distance that the goalkeeper is willing to move away from the goal in order to reach the ball. By its default behaviour, the goalkeeper will not leave area A1 if the ball stops anywhere outside area A2. Area A3 shows the new extended area of the goalkeeper's willingness to reach the ball when using SAFEL. A3 also represents the area of the field dedicated for positioning the ball at the beginning of each test execution. Finally, area A4 shows the part of the field dedicated for positioning the opponent striker

Table 6.1: Strikers' behavioural characteristics per team.

Attacking Teams	A	B	C	D
Aggressive Ball Pursuit	✓	✓		
Opportunistic Rebound	✓		✓	
Strong Shot to Goal				✓

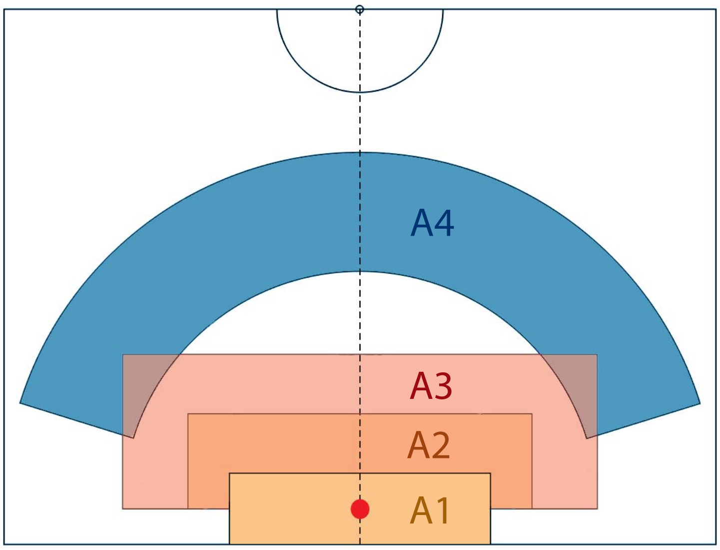


Figure 6.2: Areas of interest in the soccer field. This image shows only the side of the defending team's field (i.e., the team of the goalkeeper), as it is where the scenario takes place. The red circle represents the initial position of the goalkeeper in this case study.

in the beginning of each test execution. Outside this area, the striker may be too far so that it cannot be seen by the goalkeeper, or too close so that it is not advantageous in any way for the goalkeeper to attempt reaching the ball. The red circle in Fig. 6.2 represents the initial position of the goalkeeper in the beginning of each test execution.

In this experiment, we simulate a particular situation (as described in Section 6.2) occurring during an ongoing match. To ensure that our scenario is coherent with possible events occurring in a real ongoing match, we have defined the following restrictions for generating the initial positions of the ball and striker in the field:

- The ball must start inside the defined area of interest A3.
- The striker must start inside area A1, which has been predefined according to (1) a maximum distance from the goalkeeper position so that it can be seen by the goalkeeper and (2) a minimum distance from the goalkeeper so that its position is coherent with that of a real ongoing match.
- The distance of the striker to the ball must be larger than the distance from the goalkeeper to the ball. The contrary would imply higher chances for the striker to reach the ball first, a situation in which leaving the goal area would not be an advantageous action for the goalkeeper.

- The striker must be on the same side of the field as the ball. This is to ensure that the initial position of the striker would be coherent with that of a real ongoing match and to prevent the scenario solution from becoming trivial.

450 unique combinations of initial striker position and initial ball position in the field were randomly generated taking into consideration the above-listed restrictions. The goalkeeper always starts in the same position, which is in the middle of its own goal line (red circle in Fig. 6.2).

We performed three experiments in this case study, each evaluating a different aspect of SAFEL's performance. For the first and second experiments, the B-Human default controller was independently executed in the simulation for all the 450 generated combinations of initial position and for each team, totalling 1,800 unique simulation executions. For the second experiment, the extended B-Human controller using SAFEL was independently executed in the simulation for 135 combinations of initial position and for each team, totalling 540 unique simulation executions using SAFEL. In addition, for the third experiment, 110 out of the 450 combinations of initial positions were selected to evaluate SAFEL's learning evolution over time for each of the four teams, totalling 440 unique simulations executed for the third experiment.

During each simulation execution, the goalkeeper continually collects information on the current state of eight stimuli, two of which were aversive unconditioned stimulus (US) and six were initially neutral stimuli, which could later become conditioned stimulus (CS). The eight stimuli $\vec{s} = [s_1, ..., s_8]$ collected by the goalkeeper, along with their type (US or CS) are:

- s_1 (US u_1): Opponent scoring. This stimulus assumes the value 1 if the opponent has scored a goal and 0 otherwise;
- s_2 (US u_2): Collision exposure. This stimulus indicates whether there was a prolonged collision with an opponent robot. The longer the goalkeeper is continuously exposed to collision, the closer this value is from 1, otherwise the closer it is from 0. The absence of collisions is indicated by 0;
- s_3 (CS c_1): Falling state of the goalkeeper itself. This stimulus assumes the value 0.4 if the goalkeeper is staggering, 0.7 if the goalkeeper is falling, 0.8 if the goalkeeper is getting up and 1 if the goalkeeper is fallen;
- s_4 (CS c_2): Normalized distance from the ball to the goalkeeper;
- s_5 (CS c_3): Normalized distance from the closest opponent to the goalkeeper;
- s_6 (CS c_4): Normalized distance from the closest opponent to the ball;

Table 6.2: Sensitivity matrix used in the experiments of the proposed case study.

	Goal (u_1)	Collision (u_2)
Falling state (c_1)	0.6	0.0
Dist. ball to goalie (c_2)	0.0	0.2
Dist. opponent to goalie (c_3)	0.0	0.4
Dist. opponent to ball (c_4)	0.0	0.2
Dist. ball to goal (c_5)	0.0	0.1
Dist. opponent to goal (c_6)	0.4	0.1

- s_7 (CS c_5): Normalized distance from the ball to the goal; and
- s_8 (CS c_6): Normalized distance from the closest opponent to the goal.

Table 6.2 depicts the sensitivity matrix used in all experiments of this case study. According to our sensitivity matrix, the US depicting an opponent's goal has a higher level of association with two CS's only, which are the falling state of the goalkeeper and the distance from the opponent striker to the goal. This is because goal scoring recurrently occurring while the goalkeeper is on the ground is a strong indication of failed catching ball attempts, showing that the goalkeeper may be ineffective against that specific team's striker. On the other hand, certain strikers may be able to score goals from farther distances than others, and associating the occurrence of goals with the distance between the striker and the goal may help identify which team has a skilled striker with strong kicks.

Finally, all the other CS's have been linked with the US depicting the occurrence of collisions. This is because collision incidence is, of course, highly related with proximity between moving entities, and all CS's except by c_1 describe distances between the elements of interest in the soccer field. The remaining parameters of SAFEL have been defined as follows:

- Global situation duration (GSD): 15 seconds;
- Adrenaline threshold: 0.5 (in the range [0,1]);
- Association rate (AR): 0.4 (in the range [0,1]) for all conditioned stimuli;

Given the above-listed set of stimuli and parameters, as well as Def. 1 and Def. 2 given in Section 4.4, we have that in this case study an event $\mathbf{e_t}$ depicts the state of the eight stimuli detected by the goalkeeper at time step t, so that $\mathbf{e_t} = [s_1^t, s_2^t, ..., s_8^t]$, where s_i^t is a value representing the value of stimulus s_i detected at time t. SAFEL's input set is composed by the vector $\vec{u} = [u_1, u_2] = [s_1, s_2]$, representing the group of US's, and the vector $\vec{c} = [c_1, ..., c_6] = [s_3, ..., s_8]$, representing the group of CS's.

A situation S in this scenario is composed of the sequence of events occurring during its active time in the simulation, so that $S_j = [\mathbf{e_{a_j}}, ..., \mathbf{e_{d_j}}]^T$, where a_j and d_j are, respectively, the times of activation and deactivation of situation j. According to the specified GSD, d_j occurs 15 seconds after the occurrence of a_j for neutral, safe and conditioned aversive (CA) situations (unconditioned aversive (UA) situations have flexible GSD as discussed in Section 4.4, which depends on the current value of the adrenaline signal). The first event e_1 is recorded at the beginning of the simulation, and the last event e_n is recorded when one of the following stop criteria is met in the simulation:

1. A goal has been scored by the opponent striker;
2. The ball has been kicked out of the field boundaries; or
3. The number of recorded events has reached 300, which we consider to be enough time for the rest of the defending team (initially on the other side of the field) to retreat and help the goalkeeper in defending the goal. In this case, the proposed scenario is over.

6.4 Results

This case study consists of three experiments, each evaluating a different aspect of SAFEL's performance in the proposed scenario. The first experiment, discussed in Section 6.4.2, evaluates SAFEL's predictive performance by analysing its capability of predicting the imminent occurrence of aversive stimuli. The second experiment, discussed in Section 6.4.3, compares the goalkeeper's playing performance with and without SAFEL's emotional feedback. The outcome of the simulations that generated the datasets used for the experiments of Section 6.4.2 and Section 6.4.3 is analysed in Section 6.4.1.

Finally, the third experiment, discussed in Section 6.4.4, explores how SAFEL's learning and environmental adaptation evolves over time. Unlike the first two experiments, SAFEL starts with an empty dataset in the third experiment, thus having no previous knowledge about the robot's environment.

6.4.1 Dataset Analysis

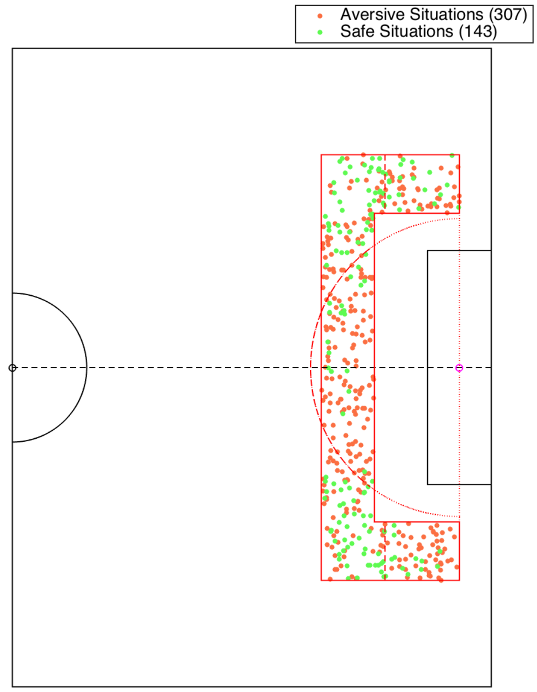
The data presented in this section is used in the experiments of Section 6.4.2 and Section 6.4.3 and have been generated via simulation with the default behaviour of the goalkeeper, as implemented by the B-Human team. For each of the four defined team behaviours, 450 independent simulations have been executed with distinct combinations of initial positions for the striker and ball, as described in Section 6.3.2.

Fig. 6.3 shows the outcome of the 450 soccer simulations in terms of situation type per initial position of the ball in the field. In other words, for each simulation, we observed the generated initial position of the ball as well as whether the result was aversive (goal and/or collision) or safe (no goals or collisions) for the goalkeeper in the end of the simulation. Fig. 6.3 depicts the integration of these two pieces of information, where the dots depict the 450 initial positions of the ball in the field and their colours indicate whether the resulting simulation was aversive (red dot) or safe (green dot). The area delimited by a solid red line depicts the area of interest A3 (defined in Section 6.3.2), which represents the new extended area of the goalkeeper's willingness to reach the ball when using SAFEL, as well as the area dedicated for positioning the ball at the beginning of each test execution. Areas delimited by dotted red lines depict the areas of the field where the ball was positioned that we observed to concentrate the majority of aversive situation outcomes, especially goal scoring. Fig. 6.4 is analogous to Fig. 6.3, however, it shows the portion of aversive situations caused by goals and by collisions.

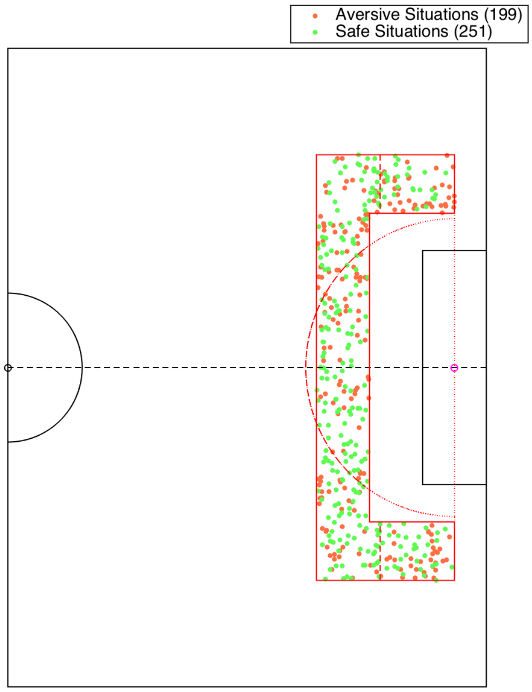
Teams A and D show a clear inclination to causing aversive situations (Fig. 6.3a and Fig. 6.3d, respectively). However, the type of outcomes that lead to aversive situations differs between these two teams. For team D, aversive situations are caused by goals in 97% of the cases (Fig. 6.4d), while the number of aversive situations for team A is well balanced between collisions (54%) and goals (46%) (Fig. 6.4a). Another difference is the areas in the field from where team A and D manage to score goals. Most goals scored by team A are inside the areas delimited by dotted red lines while team D managed to score goals from almost all regions inside area A3. This is likely because the striker from team D has a stronger kick than strikers from the other teams, as discussed in Section 6.3.2, thus being capable of scoring from farther distances with one single kick.

Different from teams A and D, team C shows a clear inclination towards safe situations for the goalkeeper (Fig. 6.3c). Nonetheless, 30% of simulations with team C resulted in aversive situations, whose majority (74%) was caused by goals (Fig. 6.4c). Like team A, most goals scored by team C occurred when the ball started within the areas delimited by dotted red lines.

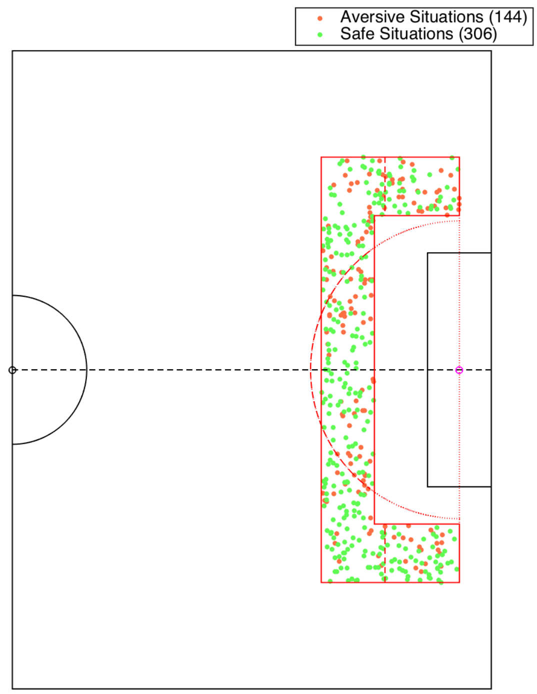
Finally, team B presents the most homogeneous result of all teams. The number of aversive (44%) and safe (56%) situations is well balanced (Fig. 6.3b), as well as the number of goals and collisions, which represent 47% and 53% of aversive situations, respectively (Fig. 6.4b). Unlike teams A and C, whose majority of aversive situations occurred when the ball started within the areas delimited by



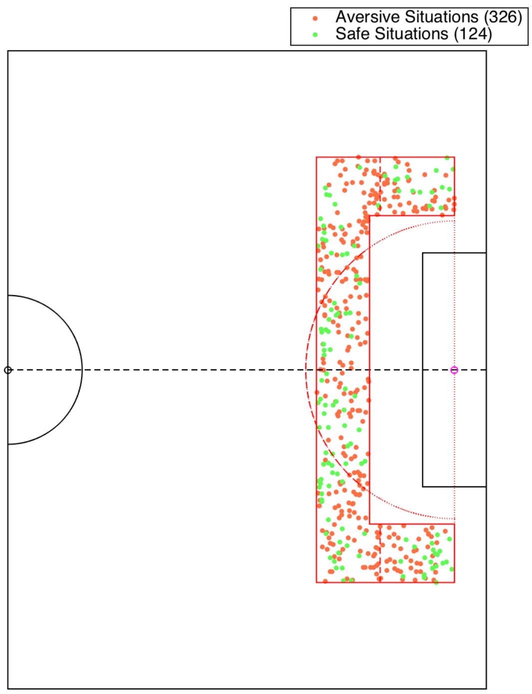
(a) Team A



(b) Team B

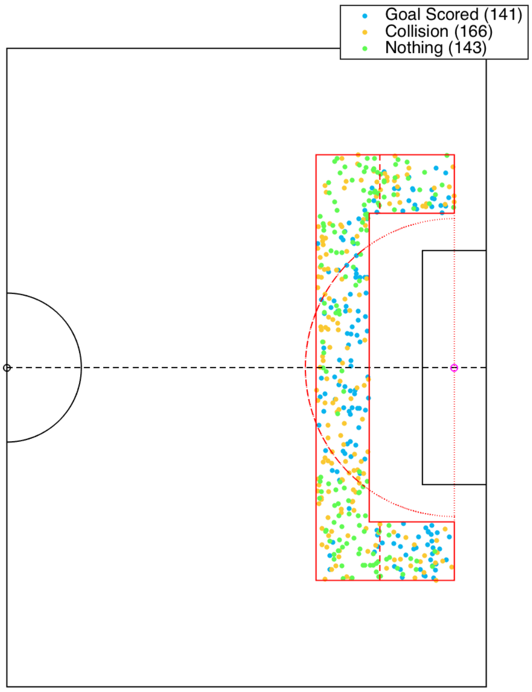


(c) Team C

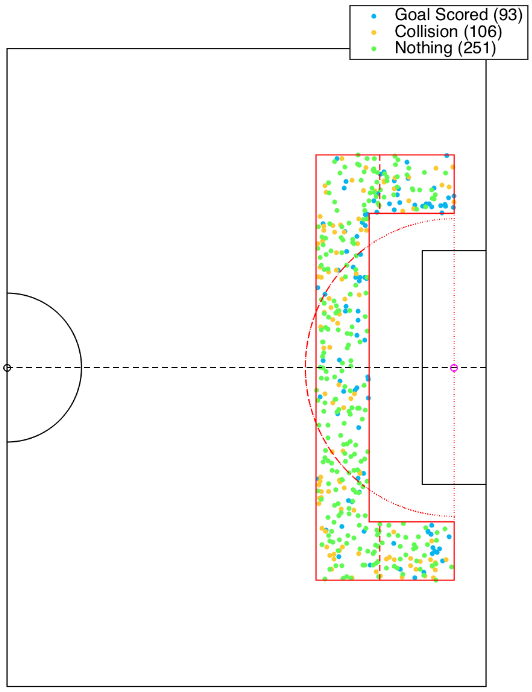


(d) Team D

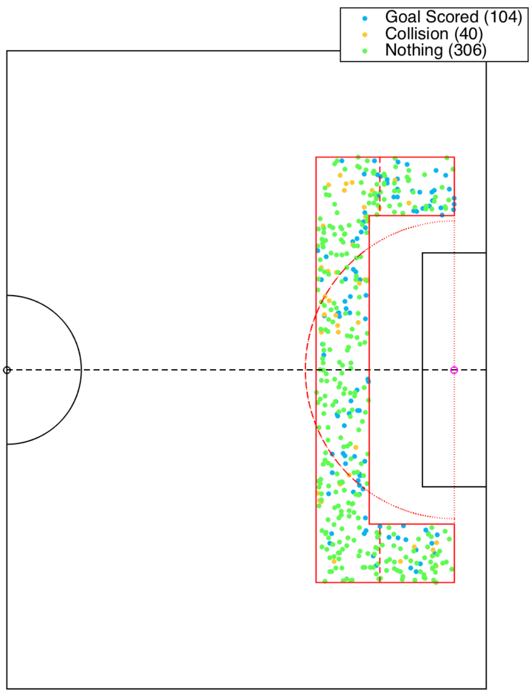
Figure 6.3: Resulting situation types (safe or aversive) per initial ball position (450 in total) for all teams.



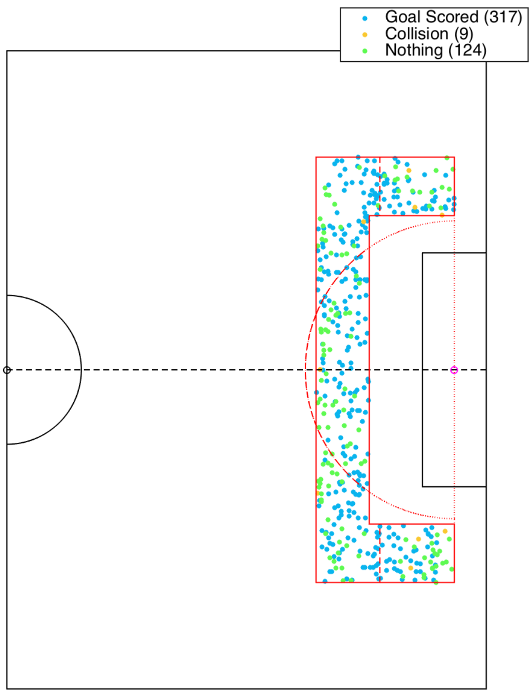
(a) Team A



(b) Team B



(c) Team C



(d) Team D

Figure 6.4: Resulting outcomes (goal, collision or nothing) per initial ball position (450 in total) for all teams.

Table 6.3: Outcome of simulations selected for training. The percentage of goals and collisions out of 315 simulations per team are show in the 'Goals' and 'Collisions' rows, respectively. The 'Aversive Tests' row shows the percentage of simulations with at least one goal or one collision. The 'Safe Tests' row shows the percentage of simulations in which neither goals nor collisions occurred.

Attacking Team	A	B	C	D
Goals (%)	31	21	23	70
Collisions (%)	48	26	10	2
Aversive Tests (%)	68	44	32	72
Safe Tests (%)	32	56	68	28

dotted red lines, the initial ball positions leading to aversive situations for team B are spread throughout the whole area A3.

From the 450 simulation executions for each team, 315 (70%) was selected for training, while 135 (30%) was selected for testing. Table 6.3 depicts the outcome of simulation executions selected for the training phase. Table 6.3 shows a clear advantage for team D in the number of goals if compared with the other teams, in virtue of its striker's stronger than average shot to goal (see Table 6.1). Despite the great number of scored goals, team D caused very few collisions because its striker has neither the aggressive ball pursuit behaviour nor opportunistic rebound inclination. Second in the ranking of goals is team A, which is the most aggressive of all teams by exhibiting both the aggressive ball pursuit and opportunistic rebound behaviours. In addition to the high number of goals, team A also caused the highest number of collisions of all teams, which is higher than the number of collisions caused by all the other three teams together.

Teams B and C are less threatening than teams A and D. The striker from team B exhibits only the aggressive ball pursuit behaviour, while the striker from team C exhibits only the opportunistic rebound behaviour. Consistent with their respective behaviours, Table 6.3 shows that team B has caused more collisions than team C, which in turn scored more goals than team B. Nonetheless, the difference between the number of goals and the number of collisions is more evident for team C (23 goals versus 10 collisions) than for team B (21 goals versus 26 collisions).

6.4.2 SAFEL Predictive Performance

This experiment uses the data generated via simulation presented in Section 6.4.1 for both training and evaluation of SAFEL's predictive performance. This experiment focuses on analysing SAFEL's predictive performance and, therefore, will not take into consideration the behaviour and playing performance of the goalkeeper

Table 6.4: Predictive Performance. SAFEL's predictive performance for each of the four teams is compared against a baseline classifier under two performance measures (F_2-score and edit distance) and two statistical tests (t-test and K-S test). $p \leq 10^{-15}$ for both t-test and K-S test in all the evaluated cases.

	Team A		Team B		Team C		Team D	
	SAFEL	Baseline	SAFEL	Baseline	SAFEL	Baseline	SAFEL	Baseline
F_2-score	0.5013	0.1198	0.5456	0.0853	0.6817	0.1440	0.5188	0.1604
Edit Dist.	0.1453	0.3044	0.1391	0.2130	0.0995	0.1520	0.1693	0.3396

(which is properly addressed in Section 6.4.3). Here, we evaluate SAFEL's success in correctly predicting future occurrences of aversive stimuli.

Predictive performance is measured at the level of the Working Memory Module (WMM). This is because at this stage within SAFEL's architecture, stimuli information has already been processed and transformed into compacted pieces of situational information. Before this phase, data instances are still complex and large sets of information organized in matrices, without well-defined classes necessary for analysing classification performance. On the other hand, data processed by the WMM conforms with the conventional dataset format, in which instances are represented by arrays of data (instead of matrices), each of which has well-established classes: safe situation or conditioned aversive situation.

We have tested SAFEL's performance against a baseline classifier that generates random predictions by respecting the class distribution of the training set. Table 6.4 compares the predictive performance of SAFEL with the baseline classifier under two distinct performance measures: the *F_2-score* and the *edit distance*. Basically, the edit distance calculates the cost for transforming an array into another. In this case, the edit distance calculates the cost to transform the array of predicted classes into the array of correct classes. Consequently, the smaller the edit distance value, the closer the predictions are from the real classes, and the better the predictive performance.

The F_2-score, on the other hand, is a modified version of the *F-score* (also known as *F-measure*) that gives more importance to recall (i.e., the fraction of actual conditioned aversive situations that were predicted) than to precision (i.e., the fraction of predicted conditioned aversive situations that are correct). The conventional F-score is defined as the harmonic mean between precision and recall, whereas the F_2-score weighs recall twice as high as precision. We opted for the F_2-score instead of the conventional F-score because it is of our interest that SAFEL demonstrates capability to mimic nature's tendency to overestimate danger (LeDoux 1999). For an animal surviving in the wild, for instance, the cost of underestimating danger is much higher (e.g., injuries, death) than the cost of overestimating it (e.g., spending energy escaping from harmless animals or objects). It follows that SAFEL would be better evaluated by a metric that rewards danger overestimation to some extent, which is the reason why we opted for using the F_2-score.

The difference in predictive performance between SAFEL and the baseline classifier has been tested under the *null hypothesis* that there is no statistically significant difference between their predictive performance given a particular performance measure (F_2-score or edit distance), which is rejected when $p \leq 0.05$. The null hypothesis has been firstly tested using the *t-test*, which is widely known as a robust test for evaluating whether two sets of data are significantly different from each other. However, the t-test may give misleading results if the data being tested are substantially non-normal. We tested the normality of our data distribution using the *one-sample Kolmogorov-Smirnov test*, also known as the *one-sample K-S test*, and observed that some of our data samples are non-normal. To ensure the reliability of our results, we have also compared the results using the *two-sample K-S test*, which evaluates the difference between the cumulative distribution functions of the sample data and, therefore, is suitable for comparing non-normal distribution data. The resulting p-value was equal to or smaller than 10^{-15} for both the t-test and the K-S test in all the evaluated cases (i.e., for all teams and performance measures).

SAFEL showed remarkably higher F_2-score and lower edit distance in comparison to the baseline classifier for all teams, which has been shown to be statistically significant by both the t-test and the K-S test. This result demonstrates that the classification tree in the WMM makes sense of the data assembled by the other modules of SAFEL and is capable of finding recurrent patterns in it. That is a clear indication of the high quality of data collection and manipulation performed in both in the Amygdala and Hippocampus modules. It also demonstrates that SAFEL is effective in the proposed scenario, by successfully finding patterns in the robot's environment and providing an adequate emotional response, which is clearly distinguished from lucky guesses.

6.4.3 Goalkeeper Playing Performance

The focus of this experiment is to verify that SAFEL can improve the playing performance of the goalkeeper by comparing the outcome of simulations executed with and without SAFEL's emotional feedback. As previously mentioned, the goalkeeper's default behaviour in the proposed scenario is to pursue the ball only if it is within a particular distance from the goal area. We argue that in certain situations, it may be more advantageous for the goalkeeper to leave the goal area, even if the ball is farther than that particular pre-defined distance.

SAFEL's task in this experiment is to learn with experience (i.e., with the training data collected with the B-Human simulation) in order to define when it is or not more advantageous for the goalkeeper to pursue the ball. By doing so, SAFEL provides the goalkeeper with more flexible and adaptable decisions, which are taken according to the playing profile of each particular opponent team.

For this experiment, SAFEL has been trained with the same training dataset described in Section 6.4.1, whose statistics has been presented in Table 6.3. When using SAFEL, the goalkeeper's behaviour is to leave the goal area whenever SAFEL predicts the imminent occurrence of an aversive stimulus (i.e., whenever SAFEL recognises the current neutral situation as being similar to a previously experienced conditioned aversive situation), and keep guard of the goal area otherwise.

This behaviour has been selected according to common-sense knowledge over the soccer scenario and has not been exhaustively tested before performing this experiment. Therefore, there may exist other options of behaviour to be expressed by the goalkeeper in response to SAFEL's emotional feedback that lead to even better playing performance. As we have mentioned in Section 6.3.1, the task of selecting the best possible behaviour to be triggered in response to SAFEL's feedback is out of our work's scope, as we understand that this task would be better undertaken by the designers of the robot's controller. Nonetheless, showing clear improvements in the goalkeeper's playing performance even thought there was no effort to optimize its response only emphasizes SAFEL's capability to improve robots' environmental adaptation.

1080 independent simulations have been executed for this experiment: for each of the four teams, there were 135 simulation executions testing the goalkeeper's default behaviour and 135 simulation executions testing the goalkeeper's behaviour under SAFEL's influence. Table 6.5 presents the outcome of the simulated scenario with and without SAFEL's influence for each team, whereas Table 6.6 shows the statistics over SAFEL's emotional responses for each opponent team.

Table 6.5 shows a reduction in the number of goals and an increase in the number of collisions for all opponent teams in the proposed scenario. This effect is explained by the fact that both striker and goalkeeper attempt to reach the ball at the same time whenever the goalkeeper decides to pursue the ball, which puts them in collision course. The best chance to prevent a goal when dealing with a particular opponent may be to leave the goal area and try to reach the ball first. However, this may also increase the chances of collision depending on the

Table 6.5: Simulations' outcome. For each combination of opponent team and goalkeeper behaviour tested (depicted by the table's columns) there were 135 independent simulation executions. The first three rows show, respectively, the percentage of simulations with at least one occurrence of (1) goal, (2) collision and (3) balls kicked out of the field line. The last row shows the summed collision exposure (see definition of stimulus s_2 in Section 6.3.2) over all simulations for a particular team and goalkeeper behaviour.

	Team A		Team B		Team C		Team D	
	Default	SAFEL	Default	SAFEL	Default	SAFEL	Default	SAFEL
Goals (%)	27	0	17	4	29	14	67	53
Collisions (%)	55	79	25	56	7	11	1	2
Ball Out (%)	35	56	44	51	19	55	16	33
Coll. Exp. Sum	2245	2540	700	1614	433	587	25	60

Table 6.6: Statistics of SAFEL's emotional response. The 'Aversive Simulations' row shows the percentage of simulations (out of 135 for each team) with aversive predictions by SAFEL. The 'Prediction Timing' row shows the average time (in percentage of events per simulation) that SAFEL takes to predict the first aversive situation. Finally, the 'Aversive Events' row shows the average percentage of aversive events per simulation.

Attacking Team	A	B	C	D
Aversive Simulations (%)	100	94	12	21
Prediction Timing (%)	26	34	76	18
Aversive Events (%)	8.9	5.6	1.9	0.6

opponent's ball pursuit behaviour.

Table 6.5 shows a clear reduction in the percentage of goals from team A when using SAFEL, which goes from 27% to no goals at all. This result is mostly a consequence of SAFEL predicting averseness in all the 135 simulations executed against team A, as seen in Table 6.6. This extreme emotional reaction shows that SAFEL successfully detected in the training dataset that team A is the most aggressive team and causes a high incidence of aversive stimuli, both goals and collisions.

Although SAFEL increased the percentage of simulations with collisions from 55% to 79% with team A, collision exposure (see definition in description of stimulus s_2 in Section 6.3.2) increased by 13% only (from 2245 to 2540). This indicates that, despite the increase in the incidence of simulations with collisions, the duration of these collisions reduced with the use of SAFEL. In addition, SAFEL also led to a considerable increase in the number of balls out, which indicates that the goalkeeper managed to maintain the ball away from the goal more frequently when under the influence of SAFEL.

The usage of SAFEL led to an impressive reduction in the percentage of goals when playing against team B. However, such good result came at the cost of a large increase in the incidence of simulations with collisions, which was aggravated by an even larger increase in the collision exposure and a modest increase in the number of balls out. This result can be explained by the similarity in the number of goals and collision in the training dataset for team B, as well as the similarity in the number of aversive and safe situations. Note in Table 6.3 that the other teams, especially C and D, have a clear inclination towards either causing more collisions than scoring goals or scoring more goals than causing collisions. Team B, however, shows the most homogeneous result among all teams in terms of goals × collisions. Additionally, team B has also the most homogeneous result among all teams in terms of safe × aversive situations.

Finally, another compromising factor with team B is the mixed outcomes for the same regions of initial ball position. Aversive situations induced by teams A and C, for example, mostly occurred from within areas of the field delimited by the red dotted lines shown in Fig. 6.3. This facilitates SAFEL in finding a pattern in the behaviour of these teams that leads to the occurrence of aversive situations. However, aversive situations caused by team B occurred from almost all regions within area A3; and the same occurred for safe situations.

Therefore, the behaviour of team B was homogeneous in three important aspects: (1) number of goals versus collisions, (2) number of aversive situations versus safe situations and (3) the initial ball positions in the field that led to safe situations and aversive situations. While teams A and D presented a moderately homogeneous result in one of these aspects only, team B presented a significantly homogeneous result in all of the three aspects, making it more difficult for SAFEL to find a coherent pattern in the behaviour of team B. This is likely the compromising factor that affected the accuracy of SAFEL's emotional response when playing against team B. Regardless of the lower accuracy, SAFEL still managed to improve the goalkeeper's decision-making during the match by considerably reducing the number of goals from the opponent.

The most impressive result in terms of flexible and adaptable decision making shown by SAFEL was with team C. By accusing only 12% of the simulations as aversive (Table 6.6), SAFEL managed to reduce the number of goals by half and more than double the number of balls out, at the cost of a slight increase in the number of collisions (Table 6.5). This result is a clear indication that SAFEL learned with high accuracy the behavioural pattern of team C, by needing a small number of correct aversive predictions to create an effective response that leads to better playing performance for the goalkeeper.

Finally, the goalkeeper's playing performance against team D had an improvement similar to that against team C when using SAFEL. With only 21% of the simulations against team D accused as aversive (Table 6.6), SAFEL managed to reduce the incidence of goals from 67% to 53%, double the number of balls out, at the cost of an insignificant increase in the percentage of collisions (from 1% to 2%), as seen in Table 6.5.

Overall, SAFEL reduced the number of goals and increased the number of balls out against all the four teams, at the cost of slight increases in the number of collisions and collision exposure. The only exception is team B, case in which there was a significant increase in collision incidence. Nevertheless, SAFEL still led to a considerable improvement in the goalkeeper's capability of defending the goal.

Most importantly, Table 6.6 is evidence that SAFEL learned the different behaviour profiles of each team by exhibiting a distinct emotional reaction to each of them, which is consistent with the behaviours described in Section 6.3.2 and the training outcome observed in Table 6.3. Table 6.6 shows that the more aggressive the team, the more simulations against this team are accused as aversive by SAFEL (first row of Table 6.6), the earlier SAFEL's detects threat (second row of Table 6.6) and the more events per simulation are recognized as threatening by SAFEL (third row of Table 6.6). Although team D scored a large number of goals, it caused very few collisions, which is the reason why SAFEL has predicted a smaller than expected number of aversive simulations and events for this team.

6.4.4 Learning Over Time

For this experiment, we focused on observing how SAFEL gradually learns with experience and the progress of its predictions' quality over time. Unlike the experiments performed in Section 6.4.2 and Section 6.4.3, where SAFEL was pre-trained with a large dataset, here SAFEL starts without any knowledge about the environment.

This experiment consists of training phases, each of which is evaluated under ten independent simulation executions. The training phases are successive and learning is cumulative. For instance, the first training phase consists of training SAFEL at runtime inside the simulation, which is later tested under another ten independent simulation executions. The goal of the test simulations is to analyse how the knowledge acquired during the training phase influenced the goalkeeper's decisions afterwards.

The second training phase follows the same procedure, but in addition, the simulation is executed with the same SAFEL instance obtained from the first training phase. This means that the learning acquired in the second training phase also takes into consideration what has been learned in the first training phase. Again, for the second training phase, ten more simulations are independently executed for evaluation. This procedure is repeated for all subsequent training phases. This experiment comprises ten training phases for each of the four teams. Therefore, the proposed scenario has been simulated 440 times for performing this experiment (4 teams, 10 training phases and 10 tests for each training phase).

The results of this experiment are presented in Table 6.7, which notably shows that, when the robot had no initial knowledge of the environment, it demonstrated intense fearful reactions to any newly occurring aversive stimuli. For instance, the first training phase resulted in either goal or collision for all opponent teams except team C (highlighted in Table 6.7). As a consequence, SAFEL predicted threat for teams A, B and D in all the 10 simulations of the first testing phase. For team C, however, there was no threat prediction in the first testing phase because SAFEL learned during the first training phase that playing against team C does not result in the occurrence of aversive stimuli.

The initial overreaction of the goalkeeper against teams A, B and D may seem exaggerated (i.e., predicting threats in all first-phase tests). However, this behaviour is consistent with that of many animals, including our own, when dealing with unfamiliar environments (Likhachev and Arkin 2000). Under unknown situations or environments, we tend to be more alert and cautious, exhibiting fiercer than normal fight-or-flight responses at any minor sign of undesirable or harmful stimulus. If over time no negative experience occurs in that new environment, we tend to feel more comfortable and become less vigilant and reactive. On the other hand, the recurrent occurrence of negative experiences in a novel environment tends to make us more uncomfortable and stressed.

This behaviour is visible along the training and testing phases. After the first two training phases against team B, the goalkeeper showed a very negative expectation of the environment by predicting threat in all 10 executions of both first- and second-phase tests. However, after observing over time that team B's striker is not as dangerous as initially assumed, the goalkeeper reduced its predictions of threat from 10 to 4 per test phase in average. We can also observe that the opposite habituation response occurs with teams A and D. Because playing against these two teams recurrently leads to either goals or collisions, instead of becoming comfortable over time, the goalkeeper maintained its initial intense behaviour by predominantly predicting threat during the tests.

Table 6.7: Statistics of SAFEL's learning over time. The 'training result' row indicates the collision exposure during the respective training phase, where asterisks indicate that a goal has been scored. Values shown for test results are the totals over the 10 test executed for each training phase.

		Training Phase	1	2	3	4	5	6	7	8	9	10
Team A		Training Result	$\mathbf{0^*}$	0	0	0^*	10	15	34	35	20	94
	Test	Goals	0	5	4	6	4	1	2	3	3	0
		Collisions	6	4	4	4	5	5	6	8	4	10
		Coll. Exp.	101	63	134	151	25	56	169	201	87	308
		Balls Out	2	2	2	3	4	2	3	3	3	5
		Predictions	**10**	9	10	7	10	8	10	10	10	10
Team B		Training Result	**4**	0	0	0	0^*	0	0	1	0^*	5
	Test	Goals	0	1	2	2	2	0	1	0	3	3
		Collisions	2	0	3	4	2	1	1	2	1	3
		Coll. Exp.	63	0	25	137	42	5	4	11	2	18
		Balls Out	2	5	2	5	4	3	3	6	4	4
		Predictions	**10**	10	4	5	4	2	4	4	4	3
Team C		Training Result	**0**	0	0	0^*	41	0	3	0	0	0
	Test	Goals	1	1	4	3	5	1	0	3	2	4
		Collisions	0	1	0	0	0	0	0	0	1	0
		Coll. Exp.	0	40	0	0	0	0	0	0	0	0
		Balls Out	5	4	3	4	4	5	3	7	3	3
		Predictions	**0**	1	0	4	4	5	2	0	3	3
Team D		Training Result	$\mathbf{0^*}$	0	45	52	35	0^*	0^*	0^*	0	16
	Test	Goals	0	4	1	5	9	1	3	6	3	3
		Collisions	0	0	6	1	0	4	2	2	4	4
		Coll. Exp.	0	0	48	131	0	40	45	44	89	110
		Balls Out	0	3	0	1	1	3	0	1	1	2
		Predictions	**10**	7	10	10	2	10	10	10	10	10

A completely different behaviour has been observed against team C. Although the goalkeeper started with a very positive expectation of the environment by predicting no threat after the first training phase, we can observe that its negative expectations gradually increase and decrease along the training phases. Also, note that these variations in expectations are consistent with the degree of challenges faced against team C during the training phases.

Interestingly, the maximum number of threat predictions seen per testing phase against team C is 5 out of 10, which is half the maximum number of threat predictions seen for the other teams. This demonstrates that SAFEL was capable of recognising that team C's striker is not as dangerous as the other teams' strikers and, consequently, provide adequate emotional feedback.

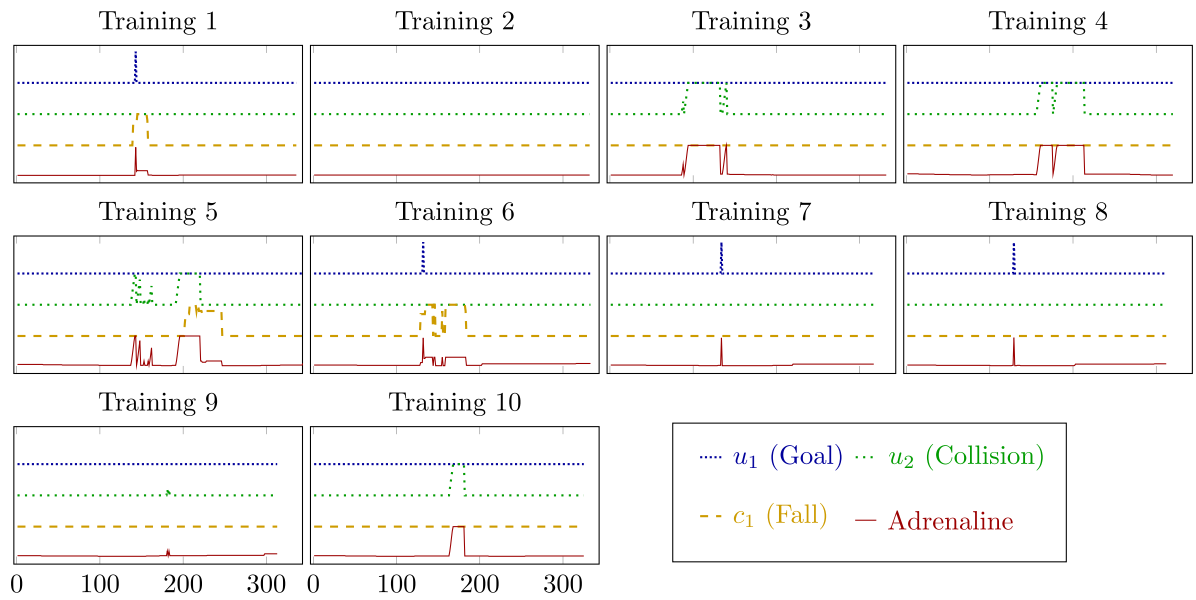
Learning in the Amygdala Module

Fig. 6.5 shows the learning progress in the AM for associating the CS $c_1 = s_3$ (the falling state of the goalkeeper) with the US $u_1 = s_1$ (opponent goal scoring), when playing against team D. Because the striker from team D has a strong shot at goal, it has better chances than the other teams to score a goal with one single kick, regardless of the farther distance. By its default behaviour, the goalkeeper tends to jump and try to catch the ball whenever it observes the ball is arriving at high speed and probably scoring a goal. If the ball is in fact too fast, the goalkeeper is likely to fail even when jumping. This is one of the reasons for the impressively high number of goals from team D seen in Table 6.3.

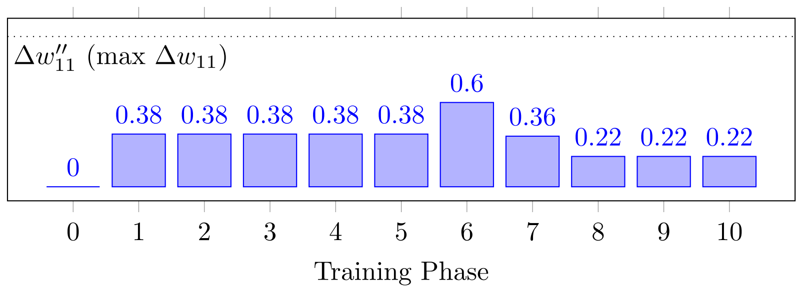
Fig. 6.5a shows the values of stimuli s_1, s_2 and s_3 (goal (US), collisions (US) and falling status (CS), respectively) and the AM's adrenaline signal varying over time for each training phase executed against team D. Fig. 6.5b shows the variation of the first-layer weight w_{11} of the ANN in the AM, which links the first-layer neuron representing stimulus c_1 (goalkeeper falling state) with the first neuron of the second (hidden) layer. In other words, Fig. 6.5b demonstrates the progression of synaptic plasticity process in the AM for neuron c_1, which is the neuron describing the goalkeeper's falling state.

We can observe in Fig. 6.5a that, already in the first training phase, there is a co-occurrence of stimuli u_1 and c_1, which leads to an update in the value of w_{11} (Fig. 6.5b). This initial association is enough to slightly increase the adrenaline signal in the first training phase right after the goal occurrence (i.e., when u_1 has stopped occurring), while the goalkeeper is still on the ground (i.e., when c_1 is still occurring). This means that, at this point, c_1 is already able to slightly raise the adrenaline signal by itself, even in the absence of the US u_1. The initially neutral stimulus c_1 has become a conditioned stimulus, though still weak (the adrenaline does not rise above the defined threshold) and unable to affect the behaviour of the robot.

There is no occurrence of falls from the second to the fourth training phases. In the fifth training phase, though, the goalkeeper falls while a collision was already happening, meaning that at some point c_1 and u_2 co-occurred. However, as specified by the sensitivity matrix depicted in Table 6.2, c_1 has sensitivity to u_1 (goal) only. Since association between c_1 and u_2 (collision) is obstructed by the



(a) Values of stimuli u_1, u_2 and c_1 (goal, collision and goalkeeper falling state, respectively) and the AM's adrenaline signal overtime for each training phase.



(b) Plasticity mechanism taking place along the training phases for weight Δw_{11} linking the first-layer neuron representing stimulus c_1 (goalkeeper falling state) with the first neuron of the second (hidden) layer.

Figure 6.5: Association process in the Amygdala Module over time when playing against team D.

sensitivity matrix, no association arises from the co-occurrence between c_1 and u_2. The absence of association can be confirmed in Fig. 6.5b, which shows that w_{11} suffered no alteration in the fifth training phase as a result of such stimuli co-occurrence. This demonstrates that the sensitivity matrix, in fact, works as expected by preventing unwanted associations.

On the sixth training phase, a co-occurrence between c_1 and u_1 happens again (Fig. 6.5a) and the value of w_{11} is once again updated (Fig. 6.5b), increasing the influence of c_1 on the value of the adrenaline signal. Nevertheless, Fig. 6.5a shows that a different event consecutively occurs on the seventh and eight training phases. A goal is scored by the opponent at some point in time in both training phases, but no falls occur. The presence of the US in the absence of the CS, as explained in Section 3.1, leads to dissociation (i.e., the forgetting of a created association), caused by the LTD phenomenon. The simulation of this phenomenon is observable in Fig. 6.5b, which shows that a decrease in the value of w_{11} occurs in the seventh training phase, and then again in the eight training phase. No meaningful event occurs in regards to the AM associative learning process in the remaining training phases.

The result seen in Fig. 6.5 is a clear evidence that the AM of SAFEL is capable of inducing associative learning that is analogous to the cued fear conditioning by successfully simulating the LTP and LTD phenomena in the brain. This result also demonstrates that such learning technique has practical application in robotics by allowing robots to successfully learn with environmental exploration and autonomously adapt.

Learning in the Hippocampus and Working Memory Modules

Interestingly, a clear distinction in how SAFEL perceived the different opponent teams is also observable in the resulting classification tree of the WMM after the training phases, which reflects the contextual learning taking place in both the HM and WMM. Fig. 6.6 shows the classification tree of the WMM after the second, sixth and tenth training phases for each of the four opponent teams.

For the sake of readability, descriptions of stimuli and extracted feature have been abbreviated in the presentation of the trees. Each node of the tree is described by an abbreviation of the format STM_F, where STM describes the stimulus and F describes the extracted feature. Stimuli are abbreviated as follows:

- **Fall:** stimulus s_3 – Falling state of the goalkeeper itself. This stimulus assumes the value 0.4 if the goalkeeper is staggering, 0.7 if the goalkeeper is falling, 0.8 if the goalkeeper is getting up and 1 if the goalkeeper is fallen;
- **DBK:** stimulus s_4 – Normalized distance from the ball to the goalkeeper;
- **DOK:** stimulus s_5 – Normalized distance from the closest opponent to the goalkeeper;
- **DOB:** stimulus s_6 – Normalized distance from the closest opponent to the ball;
- **DBG:** stimulus s_7 – Normalized distance from the ball to the goal; and
- **DOG:** stimulus s_8 – Normalized distance from the closest opponent to the goal.

While the extracted feature, which characterizes a temporal property of a particular stimulus during the life cycle of a particular situation, is abbreviated as follows:

- **M**: the mean value of that stimulus during the life cycle of the respective situation.
- **S**: the skewness of that stimulus' distribution during the life cycle of the respective situation.
- **P**: the number of peaks (or local maxima) in the distribution of that stimulus during the life cycle of the respective situation.

For instance, a tree node described as DBK_M will evaluate the mean value of the distance between the ball and the goalkeeper during the life cycle of each situation instance that visits that node of the tree.

The first obvious difference is in the size of the classification trees, which varies not only between different training phases but also between different teams. For all teams, increasing the number of training phases consistently leads to larger classification trees. As discussed in Section 5.3, the more the robot explores the environment and experiences new aversive situations, the larger is the dataset depicting the robot's environmental conditions and, consequently, the more knowledge the classification tree is able to absorb. Fig. 6.6 is a clear evidence that the knowledge base of the tree (and consequently the robot's capability to predict imminent aversive events) increases with experience, as the robot explores the environment.

However, in terms of flexible decision making and adaptive learning, the most important difference to be observed is how the trees have grown for each different team. Interestingly, all teams but team C have started with the exact same pattern. These tree structures is coherent with the result seen in Table 6.7, which shows that the first training phase resulted in either goal or collision for all opponent teams except for team C. Since there was no aversive situation to learn, the classification tree assumed that any situation instance arriving for prediction regarding team C would belong to the 'safe situation' category. Also, the decision of the trees for teams A, B and D after the second training phase is solely based on the distance from the ball to the goal. Curiously, this is the same criteria for leaving the goal area implemented by the B-Human team in their default goalkeeper behaviour.

Despite the initial similarity, the classification trees started to grow differently for each different opponent over the training phases, as the game style of each team's striker gradually became distinct. Observe, for example, that the classification tree after the sixth training phase is much larger for team D (Fig. 6.6d) than for the other teams. Different from the goalkeeper's default behaviour, which is based only on the distance from the ball to the goal area, the decision of the tree for team D is based on the distances from the ball and from the opponent to all the other elements of interest in the field, as well as all the ways that the temporal properties of these variables can be expressed in SAFEL (i.e., mean, skewness and number of peaks). In comparison with the default behaviour of the goalkeeper as implemented by the B-Human team, the decision of leaving or not the goal area with SAFEL is based on a much richer net of factors, which in turn is constructed based on events taking place at real time while the goalkeeper is in action.

The tree for team C (Fig. 6.6c) after the sixth training phase is, perhaps, the easiest to interpret and greatly matches the profile of team C, as well as the events taking place between the second and sixth training phase. This tree can be interpreted as follows:

1. If the opponent has been close to the goalkeeper for a while, then warn imminent threat (This is likely to lead to collisions);

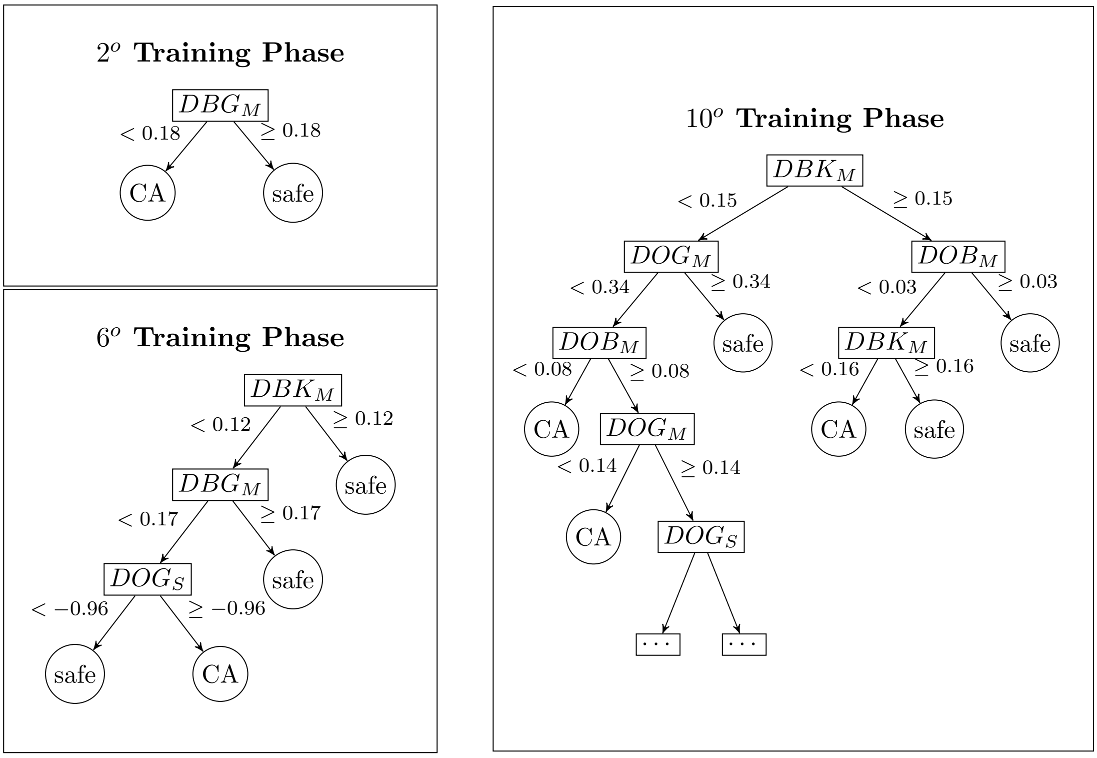
2. If the opponent has been relatively far from the goalkeeper, but the goalkeeper has been fallen for a long time, then warn imminent threat (This is likely to lead to goals);

3. If the opponent has been relatively far from the goalkeeper and the goalkeeper has been fallen but not for long, then warn imminent threat only if the opponent is relatively close to the goal (this is likely to lead to goals);

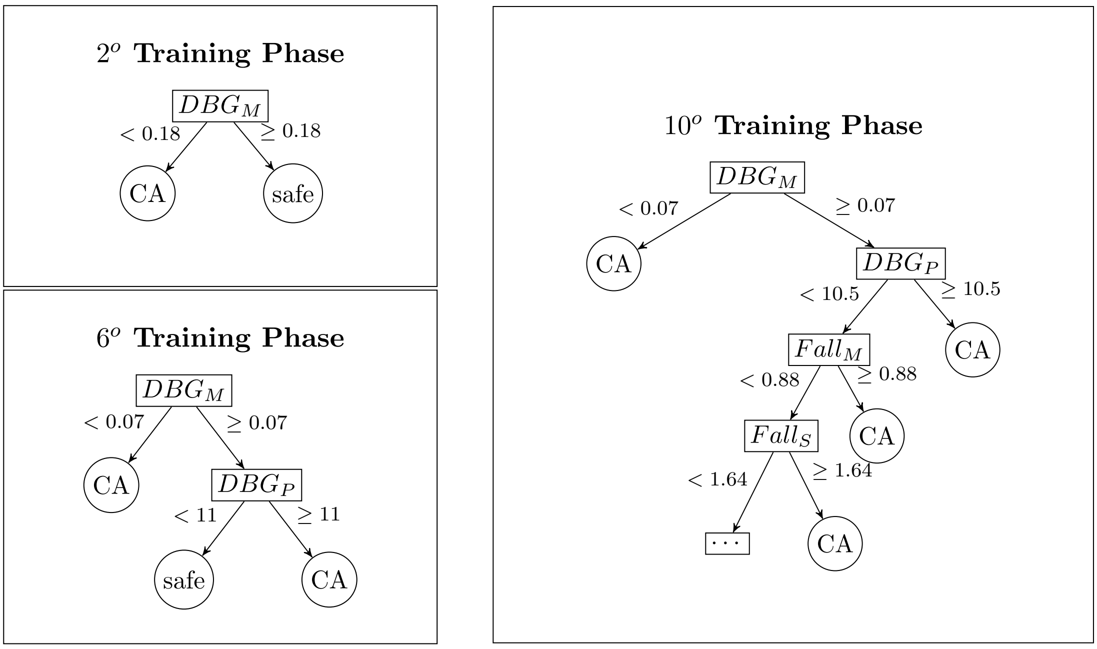
4. Otherwise, consider this a safe situation.

We can conclude from the rules above that, in this specific scenario against team C, SAFEL is particularly concerned with states of the world where a rebound is likely to occur (e.g., the goalkeeper being on the ground after an attempt to block the ball while the opponent is getting closer to the goal). SAFEL's concern with rebound opportunities is an exact match with the profile of the team C described in Table 6.1. According to Table 6.1, the striker from team C has neither a strong shot to goal nor an aggressive ball pursuit behaviour. The only behaviour this team has is, in fact, the opportunistic rebound. Additionally, the situations described above in items 2 and 3 have happened in these exact circumstances during the fourth training phase[2], which is probably the moment when the tree started to change and adapt to the opportunistic rebound behaviour of team C. The reasoning depicted by the rules of this classification tree is strong evidence

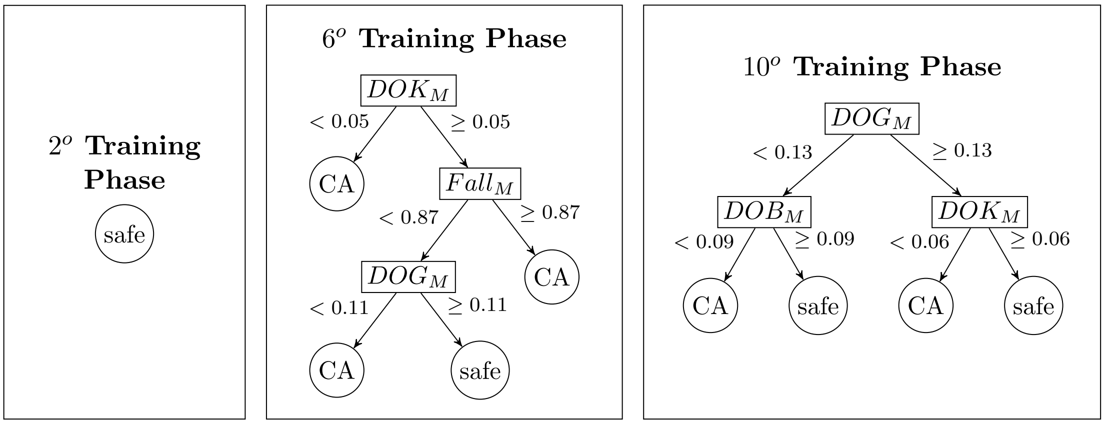
[2]These situations can be observed in the videos of the simulations, which are available at `https://www.cs.kent.ac.uk/people/rpg/cr519/casestudy`.



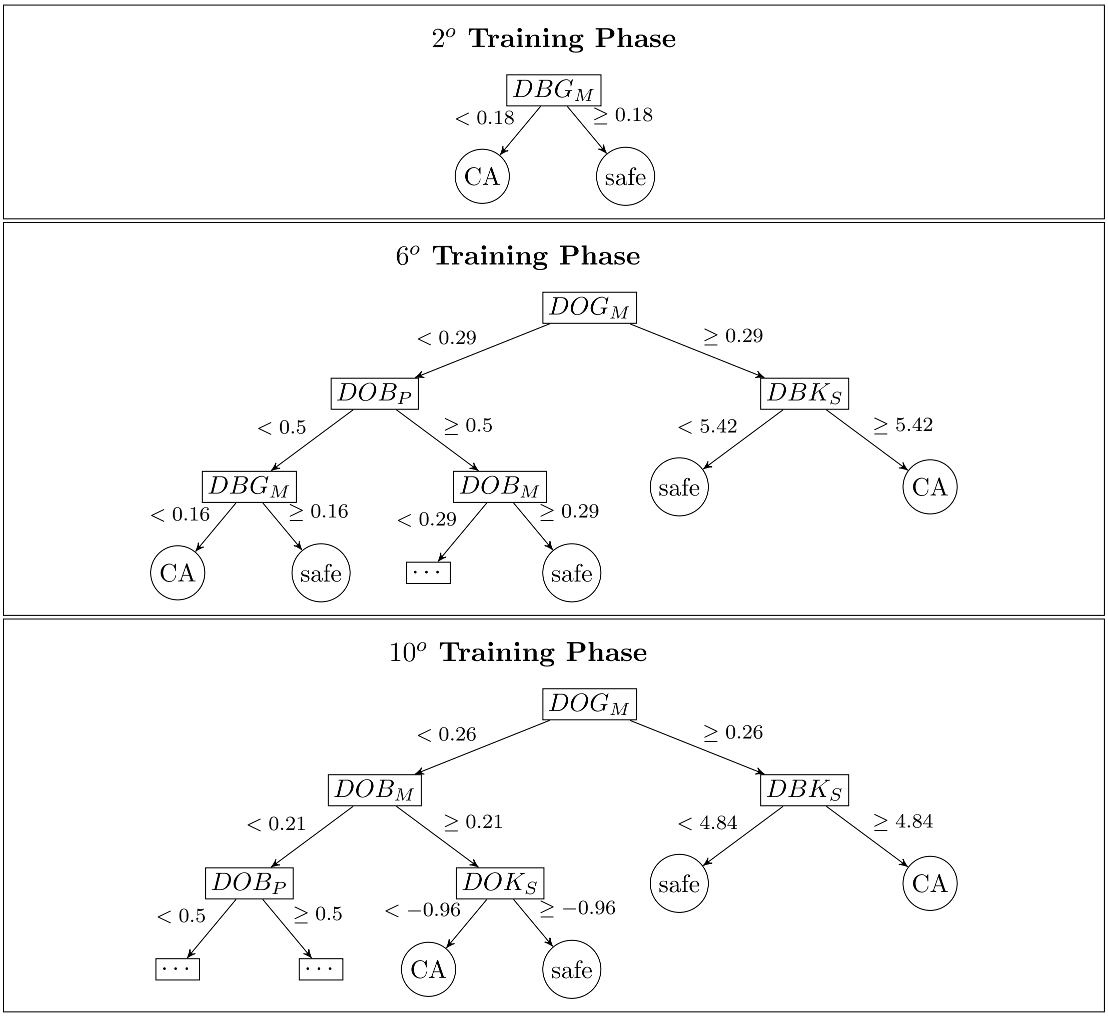
(a) Team A



(b) Team B.



(c) Team C.



(d) Team D

Figure 6.6: Classification tree of the WMM after the second, sixth and tenth training phases for each of the four opponent teams. For the sake of readability, some of the large trees have been truncated in their presentation.

that, together, the HM and WMM of SAFEL are able to identify unique patterns in stimuli interaction over time that characterizes the distinct profile of each team.

6.5 Discussion

Next we discuss how the learning and association processes in each module influences SAFEL's overall learning and emotional feedback, where Section 6.5.1 discusses the role of the Amygdala Module (AM) and Section 6.5.2 discusses the role of the Hippocampus Module (HM) and the Working Memory Module (WMM) in SAFEL's predictions. Section 6.5.3 gives a closure to this discussion by recapitulating the importance of the integrated work and communication between all modules in the SAFEL model. Finally, Section 6.5.4 discusses other possible scenarios in the RoboCup context where SAFEL could also be used to provide adaptation and flexible decision making.

6.5.1 The Role of the Amygdala Module

Although the AM itself is not solely responsible for the positive results presented in Section 6.4, it plays an important role in the overall learning process. While the HM is concerned with creating and associating representations of context with aversive stimuli, the AM is the one module responsible for revealing to the HM what stimulus is or not aversive. More importantly, the AM is responsible for finding new aversive stimuli (in addition to those pre-defined) with which the HM can associate context. This discovery of new aversive stimuli is based on a conditioning-like procedure taking place at runtime. Therefore, the AM learns with experience which stimuli in the environment are threatening enough so that the HM and WMM should be aware of its existence as aversive.

For instance, Section 6.4.4 demonstrated how the adrenaline signal is influenced by the conditioning process taking place at the AM. Because the sensitivity matrix indicates a high relationship between stimuli u_1 (goal scoring) and c_1 (goalkeeper falling state), an association process takes place whenever these two stimuli co-occur. As a consequence, stimulus c_1 becomes more efficient in raising the adrenaline output from the AM. After a few repeated co-occurrence of these two stimuli, c_1 becomes efficient enough to raise the adrenaline signal above the predefined adrenaline threshold, thus leaving the status of neutral stimulus (NS) and becoming a conditioned stimulus (CS).

From this moment on, the HM will start to associate with fear any context preceding the occurrence of c_1, regardless of the state the other aversive stimuli.

In other words, the HM will label as conditioned aversive any situation preceding the occurrence of c_1. Consequently, the WMM will learn the pattern of these situations as well, in addition to those preceding US's. In the future, the WMM's classification tree will be likely to predict averseness whenever it receives from the HM a neutral situation with pattern similar to those preceding c_1. For the robot, this prediction means that something undesirable is about to happen in the near future. However, in a deeper analysis, we can state that the WMM is actually predicting that the goalkeeper is about to jump for catching the ball and is likely to fail.

Unlike aversive US's, which are universally aversive regardless of what is happening in the environment, aversive CS's may return to the status of NS. As explained in Section 3.1, the dissociation process that turns a CS into a NS takes place when the US recurrently occurs in the absence of the CS. This process has also been demonstrated in Section 6.4.4 to be successfully induced in the AM. In our example, dissociation would mean that at some point during a soccer match the goalkeeper is no longer jumping to catch the ball when goals are scored. This could happen for a number of reasons.

For instance, the over-time increasing frequency of fearful responses from SAFEL induces the goalkeeper to more frequently leave the goal area for pursuing the ball. Consequently, there will be times in which the goalkeeper will be disputing the ball with the striker instead of standing in the goal area. Suppose that in some of these cases, the striker manages to dribble the goalkeeper and score a goal. Because the goalkeeper is away from the goal, it has no reasons to jump. Therefore, in this scenario a goal has been scored but the goalkeeper did not fall, case in which the dissociation process takes place.

The likelihood of a goal being scored while the goalkeeper is jumping to catch the ball is now reduced by the fact that the goalkeeper is more often disputing the ball with the striker instead of standing in the goal area. This makes the association between u_1 and c_1 less relevant than it was in the beginning, which justifies the dissociation.

Finally, it is interesting to observe that the AM has created strong associations between u_1 and c_1 only when the goalkeeper was playing against team D. This is in virtue of team D being the only team with a striker capable of frequently scoring goals from farther distances with just one kick. This is a clear evidence that the AM was not only capable to create associations and detect new aversive stimuli, but also to identify aversive stimuli based on the profile of the opponent team.

6.5.2 The Role of the Hippocampus and Working Memory Modules

Although the AM is essential in the SAFEL architecture, as discussed in Section 6.5.1, it is unable to make sense of context or its temporal properties. Like the AM, the HM and WMM together play an essential role in the SAFEL model, which is understanding and attaching emotional labels to the robot's state of affairs. This task involves several subtasks, which are divided between the HM and the WMM.

The robot's perception of the environment starts to be constructed in the AM, of course, but it is in the HM that environmental perception becomes more elaborated and realistic. While the AM detects punctual events capturing unique traits of each team, the HM is able to gather complex sets of information that can describe with more details the unique patterns of action of each team and, therefore, is better able to differentiate one team from another based on their in-game actions.

For instance, the AM may be able to detect that the goalkeeper is usually fallen when team D scores a goal because these two stimuli co-occur. As discussed in Section 6.4.4, this association is important because it indicates that jumping to catch the ball is an ineffective strategy against team D. However, the AM is usually unable to predict that a goal is likely to occur in the near future based on what is occurring at the moment because, in this case, the stimuli relevant for the prediction are not co-occurring. This kind of prediction would require a mechanism that makes sense of how events evolve over time, which is provided by the HM and WMM.

The situation management performed in the HM allows SAFEL to build pieces of information describing how environmental stimuli interact over time. This provides SAFEL with a much richer perception of the environment, allowing the robot to observe and understand a level of detail in the environment that would not be possible with information gathered in the AM only.

For instance, the AM is able to detect that a particular US u_1 occurs whenever a particular CS c_1 has low value and another particular CS c_2 has high value. However, only the WMM is able to detect that some US is likely to occur at time t_{n+100} if the CS c_1 is high at time t_n and gradually decreases its value until time t_{n+100} while the CS c_2 is constantly high from time t_n to time t_{n+100}. Both cases describe the same situation. However, while the AM is detecting the occurrence of the US only around time t_{n+100} (possibly a couple of time steps earlier), the WMM has already predicted it at time t_n. Such prediction from the WMM is only possible, though, due to the assembling of situation instances performed by the HM. Therefore, SAFEL's contextual perception of the environment is not a merit of the HM or the WMM independently but instead of the collaboration between these two modules, whose harmony is highly dependent on the emotional feedback provided by the AM.

In the scenario of this case study, the kind of data processing performed in the HM and WMM allowed constructing logical premises describing specific details of each team's playing profile as discussed in Section 6.4.4. It is interesting to observe that a completely different reasoning has been built by the classification tree of the WMM for each of the four teams, which is in accordance with the behaviours described in Table 6.1. While the AM only managed to build a clear distinction between team D and the other teams, the WMM managed to build distinct logical premises for each of the four teams, thus successfully differentiating all of the teams from each other.

It is worth noting that the difference in the complexity of data processed by the AM and by the HM/WMM also represents the pivotal difference between SAFEL and BEL that led to the result observed in the experiment of Section 5.4.5. The level of details perceived in the environment by BEL would be analogous to the level of environmental perception from SAFEL if it was composed of the AM only.

6.5.3 The Big Picture

Now that the functioning of each of SAFEL's modules has been explained in detail and thoroughly analysed in the case study, we can review the model from a global perspective. During environmental exploration, a number of factors can be relevant for predicting the occurrence of an aversive stimulus, such as:

1. The current state of a particular stimulus,
2. The current state of a particular set of stimuli,
3. The way the state of a particular stimulus changes over time, and
4. The way the state of a particular set of stimuli changes over time.

Some of these factors may be more or less relevant than the others, depending on the robot's context and task, but most importantly, they all may play a role when predicting aversive events. Therefore, none of these factors should be neglected in the fear-learning process. In contrast to previous fear-learning models (see Chapter 2), SAFEL takes all the above-listed factors into consideration in its fear-learning and decision-making processes.

The AM deals with the relevance of the robot's present state-of-affairs, learning which states of particular stimuli are relevant for detecting averseness. The HM, in turn, collects the state of sets of stimuli, and their variations over time. Finally, the WMM makes sense of the information collected by the HM by fusing it with the information coming from the AM. Each module of SAFEL's architecture contributes in a distinct and relevant way to the fear learning and prediction processes, and removing any of them would imply in ignoring at least one of the factors previously listed.

In addition, due to the parallel communication between its modules, SAFEL is capable to simulate two aspects of fear memory in the brain that are broadly explored by LeDoux (2003, 1999), which have already been discussed in Section 3.1 and Section 4.1. The first aspect is based on the idea of a unified representation of context, which considers not only the relevance of individual stimuli per se, but also their inter-relationship and the way it varies over time.

The second aspect is based on the distinction between the *emotional memory* and the *memory of emotion*. The AM creates an *emotional memory*, by indicating through the adrenaline signal when the robot is presently facing one or more aversive stimuli and, therefore, should trigger fearful reactions and behaviours. While the amygdala receives categorized input (NS or aversive US), the HM receives the same input set, but uncategorised, so to generate an emotionally neutral memory. Ultimately, the WMM uses the emotionally neutral memory coming from the HM to retrieve a *memory of emotion* once generated by the AM in previous experiences of the robot.

A final consideration is that some existing models inspired by the real mechanisms of the brain focus on providing a close-to-real emulation of brain functions without addressing its practical usage in artificial intelligence (Gardner and Grüning 2013; Subagdja and Tan 2015). Although SAFEL attempts to mimic certain aspects of the brain's mechanisms of fear learning and memory, it has no intention of achieving a realistic emulation of the brain's functions and is exclusively focused on real-world applications for artificial intelligence and autonomous robotics.

6.5.4 Other RoboCup Scenarios

In addition to the scenario explored in Section 6.2, SAFEL has great potential to be used in other RoboCup scenarios as well, as we further discuss in Rizzi, Johnson and Vargas (2017). Next, we briefly suggest additional scenarios and approaches through which SAFEL could be used for improving collaborative behaviour, post-coordination and *ad-hoc* teamwork.

Anticipated Help Request: In many soccer situations, an undesirable outcome may be unavoidable, regardless of a player success in correctly predicting it and taking the appropriate actions. For instance, in the scenario of the case study discussed in Section 6.2, the best way for the goalkeeper to avoid a goal when dealing with the striker from team A is to leave the goal area and try to reach the ball first. However, this action also increases the number of collisions, since both striker and goalkeeper attempt to reach the ball at the same time. One option for the goalkeeper in this case would be to, in addition to pursuing the ball, message teammates whenever SAFEL predicts an aversive situation. By requesting help before it is actually needed, the goalkeeper allows its teammates to act with antecedence and, perhaps, aid in situations where help would be impracticable without the opportunity to anticipate their actions.

The Coordinator-Robot Approach: Ros et al. (2009) propose an coordination protocol in which one single robot is selected as the "coordinator". The coordinator is responsible for reasoning over the current problem, given the state of the world, and messaging to the remaining teammates the sequence of actions that should be executed by the group in order to solve that problem. This approach provides a simple solution to the post-coordination problem, while preventing decision-making conflicts that could arise from a distributed system, and can easily be applied using SAFEL. For instance, SAFEL could be used only with the selected coordinator robots, which in turn can change the actions and strategies of the whole team whenever SAFEL predicts danger.

Drop-in Competition: The drop-in competition, introduced in 2013 (Genter, Laue and Stone 2015), encourages the creation of agents capable of coordinating and co-operating with other teammates in an *ad-hoc* manner. In this competition, robots of different RoboCup teams collaborate as a single team towards a common goal: win the match with the highest goal difference possible. The biggest challenge of the drop-in competition reside in the the lack of pre-coordination, which affects the players' capability to properly communicate. Because of the limited and possibly misleading communication in the drop-in competition, many RoboCup teams do not completely trust their teammates' messages, completely ignoring them in many cases. This is because misleading communication could lead a player robot to engage in a disadvantageous or non-intelligent behaviour, which would negatively affect its score in the competition. In this sense, SAFEL could improve *ad-hoc* teamwork in the drop-in competition by providing means to learn and predict (1) the reliability of each teammate's communication; (2) the relative skill of each teammate in particular roles and positions; and (3) when it is better to make or receive a pass.

6.6 Final Considerations

This chapter analysed the performance of SAFEL in the robot soccer context. More than a competition, the Robot World Cup (RoboCup) is an initiative with the ambitious challenge of evolving robotics and artificial intelligence technologies to a level where a soccer team of robots can defeat the best soccer team of humans in the world. Such challenge has motivated researchers in a variety of robotics related areas to develop machines, algorithms and models that contribute to the goal of the RoboCup initiative. Nonetheless, artificial intelligence techniques related to flexible decision making, contextual perception and real-time adaptation still have a modest presence in RoboCup's list of accomplishments. This chapter explored how SAFEL can contribute to filling this gap while also evaluating its performance in the robot soccer scenario.

A thorough analysis has been conducted in this chapter, which evaluated SAFEL under three different perspectives: (1) the predictive performance of SAFEL in Section 6.4.2, (2) the robot's adaptation performance in Section 6.4.3 and (3) how learning evolves at runtime inside SAFEL's modules in Section 6.4.4. For such analyses, we set up a scenario where the goalkeeper of one team is alone with the striker of the opponent team and must decide, based on the playing profile of the opponent striker and other factors in the game, whether to leave the goal area unattended to pursue the ball. To construct this scenario, we used the robot soccer controller and simulation tool of the B-Human team (Röfer et al. 2015), which is currently among the best teams in the Standard Platform League (SPL) of RoboCup.

The results of the first experiment, which evaluates the predictive performance of SAFEL, demonstrated that SAFEL is capable of finding recurrent patterns in the behaviour of each team and providing an adequate emotional response by anticipating the occurrence of collisions or goals. The second experiment, which compared the playing performance of the goalkeeper with and without SAFEL, demonstrated that SAFEL reduced the number of goals and increased the number of balls out against all the four teams, at the cost of slight increases in the number of collisions. It is also worth noting that SAFEL's emotional response was different for each team and coherent with the particular behaviours that we have implemented for each of them, which is a strong evidence that SAFEL successfully learned the different behaviour profiles of each team during this experiment.

Finally, the third and most comprehensive experiment aimed at monitoring how SAFEL gradually learns with experience and how the quality of its predictions progresses over time. Unlike the first and second experiments described above, SAFEL started with an empty dataset in this experiment. This means that SAFEL started without any knowledge about the environment and adapted to it solely at runtime.

During this experiment, SAFEL was allowed only 10 independent simulations of the scenario for each team to learn and adapt to the particular strategies and behaviours of each team's striker. Despite the short number of simulations allowed for training, SAFEL showed differentiated emotional responses for each team, which were coherent with the events taking place during the respective simulations.

During the first rounds of simulation, when SAFEL still had insufficient time to observe the environment and build a robust expectation of it, SAFEL reacted with intense fearful responses to any newly occurring aversive stimuli. Interestingly, this behaviour is consistent with that of many animals, including our own, when dealing with unfamiliar environments. After a few more simulation rounds, though, SAFEL could better observe the environmental outcomes with each team and adequate its emotional responses, responding with more 'confidence' to less threatening strikers and maintaining the initial stress against more aggressive or skilled strikers.

Section 6.4.4 also analysed how the learning and prediction processes evolved over time inside each independent module of SAFEL. Ultimately, Section 6.4.4 concluded by discussing how the results observed in this experiment were affected by each independent module of SAFEL and how their integrated work is essential to successfully accomplish all the goals of the SAFEL model.

Together the three experiments presented in Section 6.4.2, Section 6.4.3 and Section 6.4.4 provide strong evidence of SAFEL's effectiveness in:

- Generating emotional responses that are coherent with the robot's state of affairs;
- Perceiving context in a detailed way, including its temporal properties;
- Identifying the specific patterns that make a particular environment different from another;
- Requiring little time of environmental exploration in order to learn relevant environmental patterns and adapting to environmental changes and threats;
- Giving the robot controller a robust basis for generating flexible decision-making and adaptive behaviour by means of its emotional responses.

Chapter 7

Conclusion

This thesis presented a novel hybrid cognitive computational model named SAFEL (Rizzi et al. 2017; Rizzi, Johnson and Vargas 2016, 2017, 2018; Rizzi Raymundo, Johnson and Vargas 2015). Inspired by well-known neuroscience findings on areas of the brain involved in fear learning, SAFEL integrates machine learning algorithms with concepts of situation-awareness from expert systems to simulate both the cued and contextual fear-conditioning phenomena. Ultimately, SAFEL provides autonomous robots with the ability to predict undesirable or threatening situations based on their past experiences and use this information for adaptation and flexible decision-making.

SAFEL consists of a hybrid architecture composed of three modules, each based on a different approach and inspired by a different region (or function) of the brain involved in fear learning. These modules are: the Amygdala Module (AM), the Hippocampus Module (HM) and the Working Memory Module (WMM). Each module was presented in a dedicated chapter, which discussed the biological inspiration, underlying technology, design and preliminary experiments (when applicable) of the respective module of SAFEL. Ultimately, a comprehensive case study was conducted to evaluate the collective work of all modules. It also analysed to which extent the emotional feedback of SAFEL can improve the intelligent behaviour of a robot in a practical real-world situation, where adaptive skills and fast/flexible decision-making are crucial.

The case study consisted of a robot soccer scenario. We focused on simulating a particular situation potentially occurring during a soccer match in the Standard Platform League (SPL) of the Robot World Cup (RoboCup) competition. In the simulated situation, the playing performance of the goalkeeper was compared with and without the influence of SAFEL's emotional responses. The core behaviour of the goalkeeper player used in this case study was implemented by the B-Human team (Röfer et al. 2015), which is currently among the best teams in the

RoboCup competition, having won it for several years. This experiment showed that, in comparison with the default player behaviour, the in-game performance of the goalkeeper was improved when the predictions of SAFEL were affecting the robot's actions. The goalkeeper was capable to distinguish the different profiles and recurrent behaviours of four opponent teams and adapt its actions accordingly during the game. Overall, when using SAFEL, the goalkeeper was able to reduce the number of goals scored by the opponent and increase the number of successful attempts at kicking the ball away from the opponent at the cost of slightly increasing the number of collisions with the opponent robot.

In addition to the player performance, this case study also evaluated the predictive performance of SAFEL in the RoboCup scenario, as well as how learning in SAFEL evolved at execution time. These experiments demonstrated that SAFEL is capable of finding recurrent patterns in the behaviour of each team and providing an adequate emotional response by anticipating the occurrence of collisions or goals. This is true even when the robot started exploring the environment with no prior knowledge about it. In this case, SAFEL showed an impressive efficiency for learning solely at runtime the particularities of each specific environment and displaying emotional responses that were coherent with the profile of each particular opponent team.

Next, we revisit the main contributions of this thesis in Section 7.1, comparing its hypothesis and research questions with the accomplishments of SAFEL as a domain independent model for providing at-runtime adaptation and robust situation-aware fear-learning capabilities to autonomous robots. In Section 7.2, we express our final considerations and opinions on the work presented in this thesis, suggesting future work and improvements to SAFEL. Finally, Section 7.3 presents ideas for future research involving SAFEL.

7.1 Contributions Revisited

This section revisits the main objectives of SAFEL and the research questions that it naturally evokes. We start this section by revisiting the central hypothesis of this thesis, as stated in Section 1.4:

> It is possible to provide robots with online and domain-independent fear learning and memory capabilities at both stimulus and contextual abstraction levels through a robust mechanism for situation awareness that considers multi-stimulus temporal relationships. Such learning mechanism shall allow robots to perceive intricate elements and relationships in their environment, learn with experience through autonomous environmental exploration

and adapt at execution time to environmental changes and threats.

It is our understanding that SAFEL has successfully achieved the aims of this thesis and fulfilled the formulated hypothesis. The SAFEL model provides robots with fear learning and memory skills in a manner analogous to both the cued and contextual fear-conditioning biological phenomena. It is also domain independent, as has been demonstrated by the several successful experiments in different scenarios presented throughout this thesis. Finally, SAFEL also allows the robot to learn about its environment without any prior information apart from the identification of unconditioned and neutral/conditioned stimuli. This learning and adaptation process takes place at runtime, during the robot's environmental exploration, and requires no explicit human or other external intervention.

In summary, SAFEL fulfils several important requirements for modelling situation appraisal for adaptive and autonomous robotics, allowing us to settle our research questions. Next, we paraphrase the research questions specified in Section 1.4 and answer them based on the achievements of SAFEL.

1. **Can a cognitive computational model be designed so to fully meet the requirements of a robust situation-aware fear-learning model of artificial intelligence?** SAFEL successfully fulfils all the requirements of a robust situation-aware fear-learning model of artificial intelligence as specified in Section 1.3. In regards to the requirements related to a situation-aware intelligence (Section 1.3.1), SAFEL allows the robot controller to:

   - Take into consideration the current state of each particular stimulus sensed by the robot at a given time, as well as how it influences the robot's interaction with its environment.
   - Take into consideration the robot's context (i.e., the combined state of all stimuli sensed by the robot at a given time), as well as how it influences the robot's interaction with its environment.
   - Take into consideration the variation of the state of each particular stimulus sensed by the robot over time, as well as how it influences the robot's interaction with its environment.
   - Take into consideration the robot's situation (i.e., how the robot's context varies over time), as well as how it influences the robot's interaction with its environment.

   In regards to the requirements related to an emotional intelligence (Section 1.3.2), SAFEL successfully simulates and/or provides:

- A neuroplasticity-like mechanism, which allows the robot to adapt to new environments by learning new stimuli associations and forgetting those associations that are no longer useful in its current state of affairs.
- Associative learning and memory, by means of the associative learning performed in both the AM and WMM, where stimuli and context are associated with the simulated fear emotion.
- Real-time learning and adaptation, by allowing the robot to learn and understand its environment at execution time and subsequently use the learned information to express adaptive and flexible decision-making behaviours.

2. **Can a hybrid cognitive computational model, depending on the contribution of different approaches and techniques, meet the requirements of a robust situation-aware fear-learning model of artificial intelligence?** SAFEL consists of a hybrid architecture composed of three modules, each based on a different approach. The Amygdala Module (AM) is based on a modified ANN that provides associative learning for cued fear conditioning. The Hippocampus Module (HM) is based on concepts of situation-awareness, which are implemented by means of a powerful rule-based platform for situation management. Finally, the contextual association and memory retrieval processes taking place in the Working Memory Module (WMM) are performed by a binary classification tree. This is evidence that a robust situation-aware fear-learning model of artificial intelligence can be built on a hybrid architecture. Additionally, the experiment discussed in Section 5.4.5 demonstrates that SAFEL displayed remarkably better performance for handling context than the BEL model, which is based on one single technique and relies on the joint work of multiple ANN modules.
3. **Can a robust situation-aware fear-learning model of artificial intelligence be effective in real-world robotics applications?** The experiments of Chapter 6 are strong evidence that SAFEL can be effectively used in real-world applications for improving the robot's task performance and increasing its chances of successfully accomplishing its goals in a dynamic and competitive environment.

7.2 Final Considerations

Together, the preliminary experiments presented in Chapter 3 and Chapter 5, along with the experiments of the case study presented in Chapter 6, provide strong evidence that SAFEL has met all of the requirements and research questions formulated in Chapter 1 and fulfilled the hypothesis stated by this thesis. Overall, SAFEL demonstrated good predictive performance and at-runtime learning capabilities. It was shown in a number of experiments in different scenarios that SAFEL effectively allows robots to use complex temporal and contextual information allied with simulated mechanisms of fear learning to predict the imminent occurrence of threats or undesirable situations. The ability to predict such events allow robots to take actions towards preventing their occurrence, thus becoming better fitted to the features of that environment, which configures the very definition of adaptation.

Notwithstanding the positive outcomes of this thesis, there are a few aspects of SAFEL that require further attention and improvements. In regards to the AM, we suggest the implementation of a better method for robot-controller designers to optionally stipulate the longevity of the AM memory, if desired. Currently, there is only one predefined parameter that allows configuring the learning latency in the AM, as well as the longevity of learned information, which is the association rate (AR). However, defining the value of the AR is not intuitive and can only be fine tuned through trial and error. Furthermore, the predefined AR value impacts both the association and dissociation processes of the AM. This means that the learning latency and the longevity of the learned information in regards to a particular stimulus are governed by the same parameter, even though these are two completely distinct processes.

Despite the discussed downsides, the AM as we designed still presents clear advantages in comparison to other models of artificial synaptic plasticity in the literature. Among the main advantages we highlight:

- The AM does not require predefining parameters to configure the learning process of internal nodes of the ANN. The AM requires parameter settings only for the input nodes, so as to define which inputs are CS and US. An optional parameter setting is also allowed to describe the sensitivity level of particular CS's to particular US's. In both cases, the relationship between the ANN's inputs is the only knowledge required from the robot designer in order to configure such parameters. This relationship, in turn, is intuitive and can be inferred from the robot's task in most cases, as the inputs of the ANN usually reflect the robot's domain of interest in the real world.

- A number of models consider that CS-US pairing occurs only when the input value of these two stimuli are simultaneously high (Morén and Balkenius 2001; Timmis, Neal and Thorniley 2009). However, unlike aversive US's (whose behaviour is known beforehand by the robot's designer and learned by the ANN at the training phase), the behaviour of an NS or CS is only revealed at runtime and, therefore, is unpredictable before that. In the real world, there are many examples of NS that assume their average or highest values in neutral situations. Contrasted to other models of synaptic plasticity, the AM of SAFEL does take this aspect into consideration.

- Although here we use the mechanism of the AM to induce fear learning, it can actually be used for other kinds of conditioning. In fact, the original design of the AM, proposed by Rizzi Raymundo and Johnson (2014), aims at providing classical conditioning between CS's and US's, where the US is not necessarily aversive. This is an important aspect of the AM, as it creates opportunities for us to modify SAFEL so to simulate emotions other than fear.

We are also aware of a few facets of the HM that require further consideration. The first that comes to our attention is the need to define an absolute and pre-fixed number for the adrenaline threshold. Contrary to how it is currently implemented in the HM, the line between fear and confidence tends towards a blurred and soft range rather than a final absolute number. Also, this range should be flexible and adaptable to the current state of affairs. In fact, the existence of such threshold leads us to our second concern in the HM, which is the limitation imposed by this module in the final fear responses of the architecture. Although the AM outputs a continuous value representing the adrenaline signal, the HM transforms it into a binary response: safe or aversive situation. The adrenaline signal of the AM allows for a larger and richer range of emotional interpretations, which are considerably narrowed by the HM. This, in turn, affects the range of affective responses accessible to the WMM, ultimately restricting the range of affective responses of SAFEL as a whole.

A third concern in the HM is the need to predefine a global situation duration (GSD), which is then equally applied to all situation types with the exception of unconditioned aversive (UA) situations. Although such parameter can be easily induced by the robot designer from the robot's domain and tasks, there is no guarantee that the duration of situations will remain the same over time, which could later lead to decay in predictive performance. We have partially addressed this issue in the latest version of SAFEL by increasing the number of existing conditioned aversive (CA) situations preceding the occurrence of UA situations. By doing so, we provide the classification tree of the WMM with a larger set of patterns from where to extract relevant information to predict aversive events, giving the WMM part of the responsibility to identify what is or not a CA situation. We recognize, though, that this is a fragile solution and should be reviewed in future versions of SAFEL.

In regards to the WMM, we recognize the demand for investigating other possible features to be extracted from situation instances delivered by the HM. Currently, a set of three features is extracted from each stimulus composing the situation instances, which are the features that we consider to better capture the main temporal characteristics of stimuli variation over time. However, a proper study to increase the number of extracted features is needed, as well as to investigate which features better represent the temporal behaviour of stimuli. Additionally, once the narrowing of affective responses in the HM (as discussed above) is addressed, we will have a margin for tackling the same issue within the WMM. This could be potentially resolved by replacing the current binary classification tree in the WMM with a regression tree. A more thorough investigation is advised though.

In addition to the discussed issues, which concern to specific modules, there are two other general aspects of the architecture that require attention. The first is that SAFEL is not currently capable to identify that the CS associated with an aversive US originates from the robot's actions. Such capacity would allow the robot to identify which of its own actions are leading to undesirable situations and subsequently adapt or inhibit such actions. This would certainly enhance the adaptive skills and flexible decision-making already generated by SAFEL.

This issue can be potentially addressed by dividing CS inputs into two types: controllable and uncontrollable. In this approach, the robot designer would not only categorize SAFEL's inputs into US's and CS's but also categorize CS inputs into those whose source is the robot (controllable CS's) and those whose source is the environment (uncontrollable CS's). The former depicts consequences of the robot's decisions and actions which are, therefore, under the robot's control. The latter depicts uncontrollable factors in the environment, such as the consequences of other agents' actions. For instance, in the case study presented in Chapter 6, the goalkeeper's decision of jumping to catch the ball would configure a controllable CS, whereas the distance between the opponent striker and the goal would configure an uncontrollable CS.

Internally to the architecture, the fear learning and association processes of SAFEL would be duplicated, so that a separate association process takes place for each type of CS. Ultimately, SAFEL would provide two fear feedbacks, one for controllable CS's and one for uncontrollable CS's. This would allow the robot controller to identify whether the imminent threat is a consequence of the robot's actions, of the environment or a combination of both. Nonetheless, this approach does not indicate which specific action (or sequence of actions) of the robot is inducing the occurrence of the aversive stimulus. To solve that, a potential approach is to simply output together with the fear responses the situation pattern that retrieved that fear response from the classification tree of the WMM.

Despite the robot not being able to recognize with the current version of SAFEL which of its actions may be compromising its own goals, we have demonstrated in Chapter 6 that such actions can be later identified by the robot designers, which may be crucial during the development phase of the robot's behaviours. For instance, the AM revealed in the experiments of Chapter 6 that the goalkeeper's default action of jumping to catch a ball was ineffective against team D while being successful against the other teams.

The second aspect is the evident need for an analysis of SAFEL's performance in terms of computation time. At the initial stages of designing and implementing SAFEL, we opted for firstly ensuring that SAFEL would meet the requirements and goals considered in Chapter 1. However, we recognize that a high performance in terms of computation time is also essential for SAFEL's purposes, as it is expected to be executed at runtime and, therefore, any latencies may compromise the performance of the robot.

An interesting fact that compensates the current lack of a processing time analysis is SAFEL's potential for working in a distributed manner. Although the proper functioning of SAFEL requires the collective work of all modules, each module works independently from the others. This means that robotic tasks demanding extensive computational resources could use SAFEL in a distributed manner by, for instance, dedicating a machine for each module of SAFEL. This would contribute to alleviating the pressure on computational resources by reducing the computational time of SAFEL and consequently reducing the computational time of the robot controller.

7.3 Future Research

In Section 7.2, we reviewed from a critical perspective some aspects of SAFEL that, in our opinion, require further improvements and should be addressed in future work. By contrast, here we explore future work in a more generic and inquisitive way, expressing our aspiring ideas for expanding and further experimenting with SAFEL in future research.

7.3.1 Reinforcement Learning

Reinforcement learning is a computational approach in which, similar to SAFEL, the agent learns by interacting with the environment (Sutton and Barto 1998). Reinforcement learning algorithms interact with the environment in time steps, where the learning agent first observes the current state of the environment at a given time t and uses a *policy* to select an action from a set of possible actions. The environment is altered by the robot's action, thus moving to a new state at time $t + 1$. The outcome of this new state in regards to the robot's goals is calculated by a *reward function*. The cycle is then restarted so that the robot makes a new observation of the environment state at time $t + 1$ and selects a new action. Ultimately, a *value function* estimates the total long-term reward an agent can expect to accumulate in the future. The main goal of a reinforcement learning paradigm is to maximize the agent's total reward in the long run.

Sutton and Barto (1998) describe rewards and values respectively as primary and secondary to a reinforcement learning agent. He states that 'Without rewards there could be no values, and the only purpose of estimating values is to achieve more reward.' Therefore, the reward is a central concept in reinforcement learning algorithms, and so is the method used to estimate it.

Rewards are usually defined by the robot designer according to the robot's task and environment. For instance, when learning to walk, the reward may be calculated proportional to the robot's forward motion. A classic example of reinforcement learning application is teaching an agent how to escape a maze. In this case, a common approach is to reward the robot with 0 or -1 at every time step prior to the actual escape and with +1 when it manages to escape. Although this methodology has been widely and successfully applied in the literature, it is limited by the fact that the reward of each environment state is pre-determined and fixed. Also, note that the source of the reward information is the environment, not the agent.

We hypothesize that SAFEL can contribute to leverage reinforcement learning approaches by providing (1) a reward function that is dynamic and capable to evolve according to environmental changes and (2) a second reward source in addition to the environment that is guided by the agent's internal states of fear. Similar to conventional reward functions, SAFEL also provides feedback about the state of the environment in regards to the robot's goals, however, based on a much richer, more complex and potentially more realistic set of environmental information, which is learned at runtime rather than pre-determined. SAFEL is capable to associate the robot's actions and state of affairs with aversive stimuli and output a fear (or negative) feedback that, if inverted, is analogous to a reward (or positive) feedback. However, different from conventional reward functions, SAFEL's feedback is based on complex temporal and contextual information deriving from multi-stimulus interactions. This reward feedback could be further improved by implementing the modifications suggested in Section 7.2, so that SAFEL could output a continuous value rather than binary.

Alternatively, we hypothesize that SAFEL could also be used to generate robust environment models. In model-based reinforcement-learning approaches, models of environments assist the decision-making process by planning courses of action based on estimations of probable future situations. Sutton and Barto (1998) define models of the environment as 'anything that an agent can use to predict how the environment will respond to its actions'. Since SAFEL can take as input the robot's actions and current environmental state and output a fear response that is equivalent to a negative feedback, we conjecture that SAFEL could be used to construct a robust model of the environment. It is part of our plans for future work to test the two hypotheses discussed above and compare their outcome with the performance of standard reinforcement learning techniques.

7.3.2 Deep Learning

We have designed and implemented the HM on a symbolic and rule-based paradigm because we believe that a standard ANN would not be capable of successfully simulating the intricate LTP process taking place in the hippocampus (Eichenbaum 2004), which is considerably more complex than the LTP process simulated in the AM. However, recent advancements in *deep learning* techniques have changed this scenario, making of *deep neural networks* (DNNs) a promising method for approaching the implementation of both the HM and WMM. This could possibly lead to the fusion of these two modules of SAFEL into a single one.

By contrast with shallow standard ANNs, DNNs can handle a large number of processing layers, consequently being able to better learn and represent data with multiple levels of abstraction. DNNs have remarkably advanced the state of the art in speech recognition, image processing, object detection and many other domains (LeCun, Bengio and Hinton 2015).

Among the many variations of deep architectures, we are mostly interested in the *convolutional neural network* (CNN). CNNs have several features that fulfil the requirements of both the HM and WMM, especially in two important aspects:

1. CNNs are designed to process data in the form of multiple arrays in order to detect local combinations of features from the previous layer (LeCun,

Bengio and Hinton 2015). This is important because situations are initially represented in the HM as matrices $S_{m\times n}$, where n is the number of different stimuli sensed by the robot and m is the number of time steps considered within the duration of a situation.

2. In addition to convolutional layers, CNNs also contain *pooling layers*, which are responsible for merging semantically similar features into one single node. This is analogous to the feature extraction process performed in the WMM.

Another interesting approach that can potentially be embedded into SAFEL's architecture is the deep *recurrent neural network* (RNN). RNNs are commonly used for processing data with intrinsic temporal information, where the sequence with which data is observed is considered to be part of the data itself. This characteristic of RNNs is very attractive for SAFEL because the major requirement of the HM is the capability to comprehend and manage the temporal properties of sensed environmental stimuli.

An RNN processes the elements of an input sequence individually and stores in its hidden layers information that implicitly depicts the history of past elements in that sequence. This powerful mechanism used to be drastically limited by the *vanishing gradient problem* common in standard ANNs trained with the backpropagation method. However, recent advancements in deep RNNs have addressed this issue and demonstrated an increasingly better capacity of predicting the future elements of a sequence (LeCun, Bengio and Hinton 2015).

Our plans for future work include investigating the use of deep learning architectures in the implementation of the HM and the WMM, especially in regards to:

- Which deep architecture (CNN, RNN, etc.) best meets the requirements of SAFEL;
- How the modules of SAFEL can be redesigned so to include deep learning techniques;
- Whether deep learning does, in fact, lead to better predictive performance than the current approach;
- How some design requirements of the WMM that are particularly well-fulfilled by the classification tree (e.g., being easy to interpret and fast to train) could be addressed when using a DNN architecture.

7.3.3 Fear-Derived Behaviours

We have discussed in Section 2.1.4 a number of methods to implement emotion-induced behaviours such as comfort, deception and guilt. Here we contemplate the modelling of these same behaviours as an expression of fear (or the lack of it in the case of comfort), which could be accomplished with SAFEL, perhaps in an even more robust manner.

In Section 2.1.4, we discussed a comfort function based on the idea of 'objects of attachment' that has been proposed by Likhachev and Arkin (2000) to control robots' navigation and exploration behaviours. We claim that, in a sense, SAFEL already simulates the idea of comfort and confidence. It may be used in a manner similar to that suggested by Likhachev and Arkin (2000), however without the need for specifying an object of attachment. For the specific problem of careful navigation and exploration in unfamiliar environments, the robot controller could provide SAFEL with an additional input that is proportional to the time spent by the robot in the current environment, whose inverse value would represent the aversive US. In other words, the more time spent in a given environment, the higher the level of familiarity with that space, the higher the robot's confidence and, consequently, the lower the intensity of the fear response.

We conjecture that SAFEL could also be used to generate deceptive behaviour by indicating when the deception does not work. For instance, the CS inputs to SAFEL could be the robot's actions when trying to deceive along with other environmental variables that may describe the robot's state of affairs. The US inputs, on the other hand, would depict the success of the robot in its attempt of deceiving. For instance, we have discussed in Section 2.1.4 the model proposed by Davis and Arkin (2012) for simulating deceptive behaviour that is based on the mobbing behaviour of a species of birds called Arabian Babbler. In the scenario studied by Davis and Arkin (2012), the CS inputs to SAFEL would represent aspects of the state of affairs observed by the agent such as the fitness of involved agents (as in the work of Davis and Arkin (2012)), the identity of the predator, distances between involved agents, and so on. US inputs, in turn, would depict the outcome of the encounter, i.e., whether the predator engaged in confronting the agents and, if so, the intensity and aggressiveness level of the confrontation.

By observing the outcome of particular confronting situations, each individual agent in the mob would learn which features of the environment are predictors of dangerous and aggressive disputes. In future encounters, SAFEL would indicate with fear levels when it is wiser to abandon the mobbing strategy and flee. The difference between the contemplated method using SAFEL and the method adopted by Davis and Arkin (2012) is that SAFEL allows a larger and more varied set of inputs that better describes the environment, which is customizable and defined according to the robot's task and needs. The decision on whether to engage in the deceiving behaviour would be also influenced by the identity of the predator and the outcome of previous encounters with it, rather than a pre-fixed fitness value only. With SAFEL, this decision would also be based on more complex and rich observations of the environment, as it would include not only the value of stimuli at a given point in time but also their variation over time.

Finally, the feeling of guilt can be interpreted as the realization of being responsible for harming, endangering or simply doing something undesirable to others. However, the awareness of being accountable for such an action is insufficient on its own to induce the feeling of guilt. For instance, a person with a psychopathic personality is unlikely to feel guilty despite being aware of its own actions and their effect on others. Therefore, the feeling of guilt also involves the sense of unpleasantness when causing harm and the desire to avoid that unpleasant sensation, which is equivalent to fear.

For instance, we have discussed in Section 2.1.4 the model of ethical behaviour for military robots proposed by Arkin, Ulam and Wagner (2012). The second module of their model, called the *ethical adaptor*, promotes the expression of guilt by adapting the robot's behaviour to the consequences of its own actions. In other words, the more unjustified damage the robot's actions cause, the higher the level of guilt and the more of its lethal weapons get deactivated. SAFEL could be used in this scenario to predict and prevent the re-occurrence of the offending actions in the specific circumstances that they happened in the first instance. Because the action of preventing past wrong-doings also configures the expression of guilt, SAFEL would be theoretically inducing ethical and guilt behaviours by means of fear simulation.

7.3.4 Beyond Fear

In this thesis, SAFEL has been specifically modelled to simulate the fear emotion. However, SAFEL's design allows the simulation of emotions other than fear, as long as their expression is conditional to the association between environmental stimuli.

For instance, in Section 3.1.2 we explained the difference between aversive and appetitive stimuli. An *aversive US* is any stimulus that naturally elicits fear or anxiety in the animal, whereas an *appetitive US* is any stimulus that naturally elicits contentment or satisfaction in the animal. Analogously, aversive conditioning leads the animal to avoid the CS that signals the presence of the aversive

US, whereas the appetitive conditioning encourages the animal to pursue the CS that signals the availability of the appetitive US. Although leading to contrasting behaviours, the same mechanism underlies these two phenomena. Equivalently, the generation of artificial aversive conditioning (as performed by SAFEL) and artificial appetitive conditioning could, in theory, be carried out by the same computational mechanism.

Appetitive conditioning could be accomplished with SAFEL by simply replacing the pre-definition of aversive US inputs with appetitive US inputs. For instance, suppose a companion robot that autonomously seeks its recharging base and recharges by itself when needed. In this scenario, the set of CS inputs (e.g., room of the house, region of the room, time of the day, etc.) would be associated with a pre-defined set of appetitive (rather than aversive) US inputs (e.g., visual detection of the recharging base, physical attachment to the recharging base, continuous increase of the battery level, etc.). This technique would allow the robot to continuously adapt its seeking behaviour for the recharging base taking into consideration its own location in relation to the location of the base in the house, even if it is moved to another location.

A robot could even be equipped with both the aversive and appetitive conditioning mechanisms to accomplish more realistic and robust adaptive behaviour. Subsequently, the robot's behaviour and action inclinations could be modelled as a function of SAFEL's fear/confidence and satisfaction/dissatisfaction outputs through an approach similar to the affective space model (discussed in Section 2.1). For instance, when dissatisfied and confident (i.e, lacking resources but in a familiar and safe environment) the robot could start seeking resources in a more 'determined' and 'aggressive' way, by moving faster and less concerned with obstacles or threats. On the other hand, when satisfied and confident, the robot would adopt a more 'relaxed' behaviour, analogous to resting, by interrupting the search for resources and lowering energy consumption. When satisfied and afraid, the robot would prioritize the search for a familiar and safe area, since resources are not a concern at that moment. Finally, when dissatisfied and afraid, the robot would perform a more prudent search for resources, by perhaps moving slower or turning on any safety mechanisms that it may have.

Alternatively, such mechanisms could be used to generate even more complex emotional behaviours as a function of fear/confidence and satisfaction/dissatisfaction values varying over time. In this approach, the robot would be constantly seeking to fulfil an ideal internal state corresponding to particular values of confidence and satisfaction. The longer the robot is in that ideal internal state, the 'happier' it is considered to be. However, the longer the robot is in a dissatisfied state, the more 'frustrated' it gets. Analogously, the longer the robot is in an afraid state, the more 'anxious' it becomes. Optionally, one could implement the 'bored' state, so that the longer the robot is in a highly satisfied and confident state, the more 'bored' it gets. This would work as a motivation for the robot to autonomously decide to explore new areas, objects, tasks, strategies, etc.

7.3.5 Applications

We believe that SAFEL can be applied to a number of real-world applications where robots are required to deal with highly dynamic or competitive environments, adapt and behave flexibly. Some examples of such scenarios are scientific exploration robots (planetary, undersea, etc.), search and rescue robots, surveillance robots, robot toys, robotic prosthetics, educational robots and elder-care robotics. Among the many application opportunities, we are especially enthusiastic to testing SAFEL for autonomous vehicles and Human-Robot Interaction (HRI).

Autonomous vehicles usually need to deal with highly dynamic and uncertain environments, in which multiple non-controllable factors may influence or compromise the safety of the passenger. Among these factors are the intentions and emotional state of other drivers. When driving, humans make use of emotional reasoning to detect the emotional state of other drivers (e.g., aggressive, friendly, rushed, apprehensive, etc.) as well as their intention (e.g., enter the roundabout first or give way). We hypothesize that SAFEL could provide autonomous vehicles with an equivalent skill, where the vehicle learns which behaviour patterns of other vehicles indicate that it is safe to advance with a particular action and which patterns indicate that it is safer to wait and give way. Subsequently, the vehicle may even learn which particular areas and roads of everyday driving paths (e.g., from home to work) that are more troublesome and, thus, require more caution.

In the area of HRI we are particularly interested in testing whether SAFEL can effectively predict the user's affective response (e.g., satisfaction or disapproval) to specific states of affair or even to particular sequences of actions of the robot. By predicting which sequence and combination of stimuli or actions precede the contentment of the user, for instance, the robot may be able to better fulfil the user's needs over time and autonomously adapt to each user's preferences.

7.3.6 Prospects

We have demonstrated throughout this thesis that SAFEL is a novel and robust online adaptation mechanism for threat prediction and prevention capable of taking into consideration complex context-temporal information in its internal learning processes. We have also demonstrated that SAFEL is domain independent by successfully employing it on different applications. In summary, we have performed extensive experiments with SAFEL and demonstrated its capabilities and contributions to the state of the art. Nonetheless, we believe and hope that much more can be accomplished with SAFEL and that its contributions to robotics and artificial intelligence can transcend the subjects covered in this thesis. We also hope that SAFEL may serve as a basis or inspiration for future works, to build better artificial mechanisms and to provide autonomous agents with even more robust emotional and adaptive behaviours.

Bibliography

ActivMedia Robotics (1999). *Pioneer 2 Mobile Robot – Operation Manual.* Peterborough.

Ahmadi, M., Keighobadi-Lamjiri, A., Nevisi, M. M., Habibi, J. and Badie, K. (2003). Using a Two-Layered Case-Based Reasoning for Prediction In Soccer Coach. In *Proceedings of the International Conference of Machine Learning; Models, Technologies and Applications (MLMTA)*, pp. 181–185.

Ahn, H. and Picard, R. W. (2005). Affective-Cognitive Learning and Decision Making: A Motivational Reward Framework for Affective Agents. In J. Tao, T. Tan and R. W. Picard, eds., *First International Conference on Affective Computing and Intelligent Interaction (ACII 2005)*, *Lecture Notes in Computer Science*, vol. 3784, Springer, pp. 866–873.

Allen, J. F. (1981). An Interval-Based Representation of Temporal Knowledge. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, San Francisco: Morgan Kaufmann Publishers Inc., IJCAI'81, pp. 221–226.

Allen, J. F. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11), pp. 832–843.

Anderson, J. (1996). *The Architecture of Cognition.* Cognitive Science, New Jersey: Lawrence Erlbaum Associates.

Andreatta, M. and Pauli, P. (2015). Appetitive vs. Aversive conditioning in humans. *Frontiers in Behavioral Neuroscience*, 9(128), pp. 1–6.

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

Arkin, R. C., Ulam, P. and Wagner, A. R. (2012). Moral Decision Making in Autonomous Systems: Enforcement, Moral Emotions, Dignity, Trust, and Deception. *Proceedings of the IEEE*, 100(3), pp. 571–589.

Azizur Rahman, M., Milasi, R., Lucas, C., Araabi, B. and Radwan, T. (2008). Implementation of Emotional Controller for Interior Permanent-Magnet Synchronous Motor Drive. *IEEE Transactions on Industry Applications*, 44(5), pp. 1466–1476.

Babaie, T., Karimizandi, R. and Lucas, C. (2007). Learning based brain emotional intelligence as a new aspect for development of an alarm system. *Soft Computing*, 12(9), pp. 857–873.

Baddeley, A. (1995). Working Memory and Conscious Awareness. In A. F. Collins, S. E. Gathercole, M. A. Conway and P. E. Morris, eds., *Theories of Memory*, Lawrence Erlbaum Associates, chap. 2, pp. 11–28.

Baddeley, A. D. and Hitch, G. (1974). Working Memory. *Psychology of Learning and Motivation*, 8, pp. 47–89.

Bali, M. (2013). *Drools JBoss Rules 5. X Developer's Guide*. Packt Publishing.

Balkenius, C. and Morén, J. (1998). Computational Models of Classical Conditioning: A Comparative Study. In R. Pfeifer, B. Blumberg, J.-A. Meyer and S. W. Wilson, eds., *From Animals to Animats: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, vol. 5, MIT Press, pp. 348–353.

Barad, M., Gean, P. and Lutz, B. (2006). The Role of the Amygdala in the Extinction of Conditioned Fear. *Biological Psychiatry*, 60(4), pp. 322–328.

Beheshti, Z. and Hashim, S. Z. M. (2010). A Review of Emotional Learning and It's Utilization in Control Engineering. *International Journal of Advances in Soft Computing and Its Applications*, 2(2), pp. 191–208.

Bekey, G. (2005). *Autonomous Robots: From Biological Inspiration to Implementation and Control*. MIT Press.

Bemelmans, R., Gelderblom, G. J., Jonker, P. and de Witte, L. (2012). Socially Assistive Robots in Elderly Care: A Systematic Review into Effects and Effectiveness. *Journal of the American Medical Directors Association*, 13(2), pp. 114–120.e1.

Bliss, T. V. P. and Collingridge, G. L. (1993). A synaptic model of memory: long-term potentiation in the hippocampus. *Nature*, 361(6407), p. 31.

Bliss, T. V. P. and Lømo, T. (1973). Long-Lasting Potentiation of Synaptic Transmission in the Dentate Area of the Anaesthetized Rabbit Following Stimulation of the Perforant Path. *Journal of Physiology*, 232(2), pp. 331–356.

Breazeal, C. and Brooks, R. (2005). Robot Emotion: A Functional Perspective. In J.-M. Fellous and M. A. Arbib, eds., *Who Needs Emotions? The Brain Meets the Robot*, Oxford University Press, Affective Science, pp. 271–310.

Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*. Belmont, California, U.S.A.: Wadsworth International Group.

Bringsjord, S. and Clark, M. H. (2012). Red-Pill Robots Only, Please. *IEEE Transactions on Affective Computing*, 3(4), pp. 394–397.

Cañamero, D. (1997). Modeling Motivations and Emotions as a Basis for Intelligent Behavior. In *Proceedings of the First International Conference on Autonomous Agents*, New York, NY, USA: ACM, pp. 148–155.

Cañamero, L. (2005). Emotion Understanding from the Perspective of Autonomous Robots Research. *Neural Networks*, 18(4), pp. 445–455.

Callan, R. (1999). *The Essence of Neural Networks*. The Essence of Computing Series, London: Prentice Hall Europe.

Coeckelbergh, M. (2012). Are Emotional Robots Deceptive? *IEEE Transactions on Affective Computing*, 3(4), pp. 388–393.

Cooke, S. F. and Bliss, T. V. P. (2006). Plasticity in the human central nervous system. *Brain*, 129(7), pp. 1659–1673.

Cugola, G. and Margara, A. (2012). Processing Flows of Information: From Data Stream to Complex Event Processing. *ACM Computing Survey*, 44(3), pp. 1–62.

Damasio, A. (1994). *Descartes' Error: Emotion, Reason and the Human Brain*. New York: Avon Books.

Daryabeigi, E., Abjadi, N. and Arab Markadeh, G. (2014). Automatic speed control of an asymmetrical six-phase induction motor using emotional controller (BELBIC). *Journal of Intelligent and Fuzzy Systems*, 26(4), pp. 1879–1892.

Dautenhahn, K. (1998). The Art of Designing Socially Intelligent Agents - Science, Fiction and the Human in the Loop. *Applied Artificial Intelligence*, 12(7-8), pp. 573–617.

Davachi, L. and Wagner, A. D. (2002). Hippocampal Contributions to Episodic Encoding: Insights From Relational and Item-Based Learning. *Journal of Neurophysiology*, 88(2), pp. 982–990.

Davis, J. and Arkin, R. (2012). Mobbing Behavior and Deceit and Its Role in Bio-inspired Autonomous Robotic Agents. In M. Dorigo, M. Birattari, C. Blum, A. L. Christensen, A. P. Engelbrecht, R. Groß and T. Stützle, eds., *Swarm Intelligence*, *Lecture Notes in Computer Science*, vol. 7461, Springer, pp. 276–283.

Dey, A. K. (2001). Understanding and Using Context. *Personal Ubiquitous Computing*, 5(1), pp. 4–7.

Eichenbaum, H. (2004). Hippocampus: Cognitive Processes and Neural Representations that Underlie Declarative Memory. *Neuron*, 44(1), pp. 109–120.

El-Garhy, A. M. and El-Shimy, M. E. (2015). BELBIC for MRAS with highly non-linear process. *Alexandria Engineering Journal*, 54(1), pp. 7–16.

Enz, S., Diruf, M., Spielhagen, C., Zoll, C. and Vargas, P. A. (2011). The Social Role of Robots in the Future - Explorative Measurement of Hopes and Fears. *International Journal of Social Robotics*, 3(3), pp. 263–271.

Ferrein, A. and Steinbauer, G. (2016). 20 Years of RoboCup. *Künstliche Intelligenz*, 30(3-4), pp. 225–232.

Forgy, C. L. (1982). Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. *Artificial Intelligence*, 19(1), pp. 17–37.

Fortin, N. J., Agster, K. L. and Eichenbaum, H. B. (2002). Critical role of the hippocampus in memory for sequences of events. *Nature Neuroscience*, 5(5), p. 458.

Fragopanagos, N. and Taylor, J. (2006). Modelling the interaction of attention and emotion. *Neurocomputing*, 69(16–18), pp. 1977–1983.

Gardner, B. and Grüning, A. (2013). Learning Temporally Precise Spiking Patterns through Reward Modulated Spike-Timing-Dependent Plasticity. In V. Mladenov, P. Koprinkova-Hristova, G. Palm, A. E. P. Villa, B. Appollini and N. K. and, eds., *Artificial Neural Networks and Machine Learning - ICANN 2013*, *Lecture Notes in Computer Science*, vol. 8131, Berlin, Heidelberg: Springer, pp. 256–263.

Garmsiri, N., Najafi, F. and Saadat, M. (2013). A New Intelligent Approach to Patient-Cooperative Control of Rehabilitation Robots. *International Journal of Engineering*, 27(3), p. 467.

Genter, K., Laue, T. and Stone, P. (2015). Benchmarking Robot Cooperation without Pre-Coordination in the RoboCup Standard Platform League Drop-In Player Competition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3415–3420.

Genter, K., MacAlpine, P., Menashe, J., Hannah, J., Liebman, E., Narvekar, S., Zhang, R. and Stone, P. (2016). UT Austin Villa: Project-Driven Research in AI and Robotics. *IEEE Intelligent Systems*, 31(2), pp. 94–101.

Hansmann, U., Merk, L., Nicklous, M. S. and Stober, T. (2013). *Pervasive Computing Handbook*. Berlin, Heidelberg: Springer-Verlag, 1st edn.

Harrison, L., Duggins, A. and Friston, K. (2006). Encoding uncertainty in the hippocampus. *Neural Networks*, 19(5), pp. 535–546.

Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory.* New York: John Wiley & Sons.

Herry, C. and Johansen, J. P. (2014). Encoding of fear learning and memory in distributed neuronal circuits. *Nature Neuroscience*, 17(12), pp. 1644–1654.

Ho, W. C., Dautenhahn, K., Lim, M. Y., Vargas, P., Aylett, R. and Enz, S. (2009). An Initial Memory Model for Virtual and Robot Companions Supporting Migration and Long-Term Interaction. In *The 18th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 277–284.

Hollinger, G., Georgiev, Y., Manfredi, A., Maxwell, B. A., Pezzementi, Z. and Mitchell, B. (2006). Design of a Social Mobile Robot Using Emotion-Based Decision Mechanisms. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing: IEEE, pp. 3093–3098.

Honey, R. C., Watt, A. and Good, M. (1998). Hippocampal Lesions Disrupt an Associative Mismatch Process. *Journal of Neuroscience*, 18(6), pp. 2226–2230.

Ippolito, C., Pisanich, G. and Young, L. A. (2005). Cognitive Emotion Layer Architecture for Intelligent UAV Planning, Behavior and Control. In *2005 IEEE Aerospace Conference*, pp. 1–16.

Jackson, P. (1998). *Introduction to Expert Systems*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 3rd edn.

Jafari, M., Shahri, A. and Shouraki, S. (2013). Attitude Control of a Quadrotor Using Brain Emotional Learning Based Intelligent Controller. In *2013 13th Iranian Conference on Fuzzy Systems (IFSC)*, pp. 1–5.

Jafarzadeh, S., Mirheidari, R., Motlagh, M. and Barkhordari, M. (2008). Designing PID and BELBIC Controllers in Path Tracking and Collision Problem in Automated Highway Systems. In *10th International Conference on Control, Automation, Robotics and Vision, 2008. ICARCV 2008*, pp. 1562–1566.

Jamali, M. R., Dehyadegari, M., Arami, A., Lucas, C. and Navabi, Z. (2010). Real-time embedded emotional controller. *Neural Computing and Applications*, 19(1), pp. 13–19.

Kim, C. and Langari, R. (2009). Target Tracking Control of a Mobile Robot Using a Brain Limbic System Based Control Strategy. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009*, pp. 5059–5064.

Kim, J., Gu, G. and Heo, P. (2016). Robotics for Healthcare. In H. Jo, H. Jun, J. Shin and S. Lee, eds., *Biomedical Engineering: Frontier Research and Converging Technologies*, *Biosystems & Biorobotics*, vol. 9, Springer, pp. 489–509.

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawai, E. and Matsubara, H. (1998). *RoboCup: A Challenge Problem for AI and Robotics*, Berlin, Heidelberg: Springer, *Lecture Notes in Computer Science*, vol. 1395. pp. 1–19.

Kolb, B. and Whishaw, I. Q. (2004). *An Introduction to Brain and Behavior*. New York: Worth Publishers, 2nd edn.

Kozima, H., Nakagawa, C. and Yasuda, Y. (2007). Children-robot interaction: a pilot study in autism therapy. *Progress in Brain Research*, 164, pp. 385–400.

Kraus, S., Althoff, M., Heissing, B. and Buss, M. (2009). Cognition and Emotion in Autonomous Cars. In *2009 IEEE Intelligent Vehicles Symposium*, pp. 635–640.

Krause-Utz, A., Elzinga, B. M., Oei, N. Y. L., Paret, C., Niedtfeld, I., Spinhoven, P., Bohus, M. and Schmahl, C. (2014). Amygdala and Dorsal Anterior Cingulate Connectivity during an Emotional Working Memory Task in Borderline Personality Disorder Patients with Interpersonal Trauma History. *Frontiers in Human Neuroscience*, 8(848), pp. 1–18.

Kwon, O., Yoo, K. and Suh, E. (2006). ubiES: Applying ubiquitous computing technologies to an expert system for context-aware proactive services. *Electronic Commerce Research and Applications*, 5(3), pp. 209–219.

Lazzeri, N., Mazzei, D., Zaraki, A. and De Rossi, D. (2013). Towards a Believable Social Robot. In N. Lepora, A. Mura, H. Krapp, P. Verschure and T. Prescott, eds., *Biomimetic and Biohybrid Systems*, *Lecture Notes in Computer Science*, vol. 8064, Berlin, Heidelberg: Springer, pp. 393–395.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature*, 521(7553), pp. 436–444.

LeDoux, J. (1999). *The Emotional Brain: The Mysterious Underpinnings of Emotional Life*. London: Phoenix.

LeDoux, J. (2003). The Emotional Brain, Fear, and the Amygdala. *Cellular and Molecular Neurobiology*, 23(4–5), pp. 727–738.

LeDoux, J. E. (1995). Emotion: Clues from the Brain. *Annual Review of Psychology*, 46, pp. 209–235.

Lewis, M., Haviland-Jones, J. M. and Barrett, L. F., eds. (2010). *Handbook of Emotions*. New York: Guilford Press, 3rd edn.

Likhachev, M. and Arkin, R. C. (2000). Robotic Comfort Zones. In *Proceedings of SPIE on Sensor Fusion and Decentralized Control in Robotic Systems III*, vol. 4196, Boston, pp. 27–41.

Lim, T.-S., Loh, W.-Y. and Shih, Y.-S. (2000). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. *Machine Learning*, 40(3), pp. 203–228.

Lin, P., Abney, K. and Bekey, G. A. (2011). *Robot Ethics: The Ethical and Social Implications of Robotics*. Cambridge, Massachusetts: MIT press.

Lotfi, E. and Akbarzadeh-T., M.-R. (2014a). Adaptive brain emotional decayed learning for online prediction of geomagnetic activity indices. *Neurocomputing*, 126, pp. 188–196.

Lotfi, E. and Akbarzadeh-T., M.-R. (2014b). Practical emotional neural networks. *Neural Networks*, 59, pp. 61–72.

Lucas, C., Milasi, R. M. and Araabi, B. N. (2006). Intelligent Modeling and Control of Washing Machine Using Locally Linear Neuro-Fuzzy (LLNF) Modeling and Modified Brain Emotional Learning Based Intelligent Controller (BELBIC). *Asian Journal of Control*, 8(4), pp. 393–400.

Lucas, C., Shahmirzadi, D. and Sheikholeslami, N. (2004). Introducing Belbic: Brain Emotional Learning Based Intelligent Controller. *Intelligent Automation & Soft Computing*, 10(1), pp. 11–21.

Malenka, R. C. and Bear, M. F. (2004). LTP and LTD: An Embarrassment of Riches. *Neuron*, 44(1), pp. 5–21.

Markadeh, G., Daryabeigi, E., Lucas, C. and Azizur Rahman, M. (2011). Speed and Flux Control of Induction Motors Using Emotional Intelligent Controller. *IEEE Transactions on Industry Applications*, 47(3), pp. 1126–1135.

Martin, S. J., Grimwood, P. D. and Morris, R. G. M. (2000). Synaptic Plasticity and Memory: An Evaluation of the Hypothesis. *Annual Review of Neuroscience*, 23(1), pp. 649–711.

Matarić, M., Tapus, A., Winstein, C. and Eriksson, J. (2009). Socially Assistive Robotics for Stroke and Mild TBI Rehabilitation. *Advanced Technologies in Rehabilitation*, 145, pp. 249–262.

MATLAB (2014). *version 8.4.0.150421 (R2014b)*. Natick, Massachusetts: The MathWorks Inc.

Mehrabian, A. R., Lucas, C. and Roshanian, J. (2006). Aerospace launch vehicle control: an intelligent adaptive approach. *Aerospace Science and Technology*, 10(2), pp. 149–155.

Michel, O. (2004). Webots: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems*, 1(1), pp. 39–42.

Miller, R. R., Barnet, R. C. and Grahame, N. J. (1995). Assessment of the Rescorla-Wagner Model. *Psychological Bulletin*, 117(3), pp. 363–386.

Moré, J. J. (1978). *The Levenberg-Marquardt Algorithm: Implementation and Theory*, Berlin, Heidelberg: Springer, *Lecture Notes in Mathematics*, vol. 630. pp. 105–116.

Morén, J. (2002). *Emotion and Learning - A Computational Model of the Amygdala*. dissertation, Lund University.

Morén, J. and Balkenius, C. (2001). Emotional Learning: A Computational Model of the Amygdala. *Cybernetics and Systems*, 32(6), pp. 611–636.

Neal, M. and Timmis, J. (2003). Timidity: A Useful Emotional Mechanism for Robot Control? *Informatica*, 27(2), pp. 197–204.

Newborn, M. (2012). *Kasparov versus Deep Blue: Computer Chess Comes of Age*. New York: Springer.

Nijholt, A., Arkin, R. C., Brault, S., Kulpa, R., Multon, F., Bideau, B., Traum, D., Hung, H., Santos Jr., E., Li, D., Yu, F., Zhou, L. and Zhang, D. (2012). Computational Deception and Noncooperation. *IEEE Intelligent Systems*, 27(6), pp. 60–75.

Nitschke, G. (2005). Emergent Cooperation in RoboCup: A Review. In *RoboCup 2005: Robot Soccer World Cup IX*, *Lecture Notes in Computer Science*, vol. 4020, Berlin, Heidelberg: Springer, pp. 512–520.

O'Reilly, R. C. and Rudy, J. W. (2001). Conjunctive Representations in Learning and Memory: Principles of Cortical and Hippocampal Function. *Psychological Review*, 108(2), pp. 311–345.

Ortony, A., Clore, G. L. and Collins, A. (1990). *The Cognitive Structure of Emotions*. Cambridge: Cambridge University Press.

Oxford Dictionaries (2017a). *Context*. Oxford University Press, Available from `https://en.oxforddictionaries.com/definition/context` [Accessed 30 June 2017].

Oxford Dictionaries (2017b). *Emotion*. Oxford University Press, Available from `https://en.oxforddictionaries.com/definition/emotion` [Accessed 14 August 2017].

Paula, M. D. and Acosta, G. G. (2015). Trajectory tracking algorithm for autonomous vehicles using adaptive reinforcement learning. In *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–8.

Pavlov, I. P. (1927). *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*. Oxford University Press.

Pereira, I., Costa, P. and Almeida, J. (2013). A Rule-Based Platform for Situation Management. In *2013 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, pp. 83–90.

Phillips, R. and LeDoux, J. E. (1992). Differential Contribution of Amygdala and Hippocampus to Cued and Contextual Fear Conditioning. *Behavioral Neuroscience*, 106(2), pp. 274–285.

Phillips, R. G. and LeDoux, J. E. (1994). Lesions of the Dorsal Hippocampal Formation Interfere with Background but Not Foreground Contextual Fear Conditioning. *Learning & Memory*, 1(1), pp. 34–44.

Picard, R. W. (1995). *Affective Computing*. Massachusetts Institute of Technology, Cambridge, Tech. Rep. 321.

Picard, R. W. (2000). *Affective Computing*. Cambridge: MIT Press.

Pioggia, G., Igliozzi, R., Sica, M. L., Ferro, M., Muratori, F., Ahluwalia, A. and De Rossi, D. (2008). Exploring Emotional and Imitational Android-Based Interactions in Autistic Spectrum Disorders. *Journal of CyberTherapy & Rehabilitation*, 1(1), pp. 49–61.

Prekop, P. and Burnett, M. (2003). Activities, context and ubiquitous computing. *Computer Communications*, 26(11), pp. 1168–1176.

Ravi, R. and Mija, S. (2014). Design of Brain Emotional Learning Based Intelligent Controller (BELBIC) for Uncertain Systems. In *2014 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pp. 1089–1093.

Read, M., Andrews, P. S. and Timmis, J. (2012). An Introduction to Artificial Immune Systems. In G. Rozenberg, T. Bäck and J. N. Kok, eds., *Handbook of Natural Computing*, Berlin, Heidelberg: Springer, pp. 1575–1597.

Red Hat (2017a). *Drools - Business Rules Management System (Java, Open Source)*. Red Hat, Inc., Available from `https://www.drools.org/` [Accessed 04 July 2017].

Red Hat (2017b). *Red Hat JBoss BRMS - business rules management*. Red Hat, Inc., Available from `https://www.redhat.com/en/technologies/jboss-middleware/business-rules` [Accessed 04 July 2017].

Reichardt, D. M. (2008). Approaching Driver Models Which Integrate Models of Emotion and Risk. In *2008 IEEE Intelligent Vehicles Symposium*, pp. 234–239.

Riek, L. D. (2015). Robotics technology in mental health care. In D. D. Luxton, ed., *Artificial Intelligence in Behavioral and Mental Health Care*, Elsevier, chap. 8, pp. 185–203.

Rizzi, C., Johnson, C. G. and Vargas, P. A. (2016). Improving the predictive performance of SAFEL: A Situation-Aware FEar Learning model. In *25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 736–742.

Rizzi, C., Johnson, C. G. and Vargas, P. A. (2017). Fear Learning for Flexible Decision Making in RoboCup: A Discussion. In *RoboCup 2017: Robot World Cup XXI*, Springer, Lecture Notes in Artificial Intelligence, in press.

Rizzi, C., Johnson, C. G. and Vargas, P. A. (2018). Online Adaptation and Situation Awareness Using SAFEL with Applications to Robot Soccer. *Artificial Intelligence*, under review.

Rizzi, C., Johnson, C. G., Fabris, F. and Vargas, P. A. (2017). A Situation-Aware Fear Learning (SAFEL) model for robots. *Neurocomputing*, 221, pp. 32–47.

Rizzi Raymundo, C. and Johnson, C. G. (2014). An Artificial Synaptic Plasticity Mechanism for Classical Conditioning with Neural Networks. In Z. Zeng, Y. Li and I. King, eds., *Advances in Neural Networks ISNN 2014*, *Lecture Notes in Computer Science*, vol. 8866, Springer, pp. 213–221.

Rizzi Raymundo, C., Johnson, C. G. and Vargas, P. A. (2015). An Architecture for Emotional and Context-Aware Associative Learning for Robot Companions. In *24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 31–36.

Rizzi Raymundo, C., Dockhorn Costa, P., Almeida, J. and Pereira, I. (2014). An Infrastructure for Distributed Rule-Based Situation Management. In *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, pp. 202–208.

Roberts, A. C. and Glanzman, D. L. (2003). Learning in Aplysia: looking at synaptic plasticity from both sides. *Trends in Neurosciences*, 26(12), pp. 662–670.

Röfer, T., Laue, T., Richter-Klug, J., Schünemann, M., Stiensmeier, J., Stolpmann, A., Stöwing, A. and Thielke, F. (2015). *B-Human Team Report and Code Release 2015*. Available from `http://www.b-human.de/downloads/publications/2015/CodeRelease2015.pdf` [Accessed 15 May 2017].

Ros, R., Arcos, J. L., Lopez de Mantaras, R. and Veloso, M. (2009). A case-based approach for coordinated action selection in robot soccer. *Artificial Intelligence*, 173(910), pp. 1014–1039.

Rosenbloom, P. S., Laird, J. E., Newell, A. and McCarl, R. (1991). A preliminary analysis of the Soar architecture as a basis for general intelligence. *Artificial Intelligence*, 47(1-3), pp. 289–325.

Rudy, J. W., Barrientos, R. M. and O'Reilly, R. C. (2002). Hippocampal Formation Supports Conditioning to Memory of a Context. *Behavioral Neuroscience*, 116(4), pp. 530–538.

Rudy, J. W., Huff, N. C. and Matus-Amat, P. (2004). Understanding contextual fear conditioning: insights from a two-process model. *Neuroscience and Biobehavioral Reviews*, 28(7), pp. 675–685.

Rudy, J. W. and O'Reilly, R. C. (2001). Conjunctive representations, the hippocampus, and contextual fear conditioning. *Cognitive, Affective, & Behavioral Neuroscience*, 1(1), pp. 66–82.

Salichs, M. A. and Malfaz, M. (2012). A New Approach to Modeling Emotions and Their Use on a Decision-Making System for Artificial Agents. *IEEE Transactions on Affective Computing*, 3(1), pp. 56–68.

Sanger, T. D. (1989). Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network. *Neural Networks*, 2(6), pp. 459–473.

Sasikumar, M., Ramani, S., Raman, S. M., Anjaneyulu, K. and Chandrasekar, R. (2007). *A practical introduction to rule based expert systems*. New Delhi: Narosa Publishing House.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, pp. 85–117.

Sharbafi, M., Lucas, C. and Daneshvar, R. (2010). Motion Control of Omni-Directional Three-Wheel Robots by Brain-Emotional-Learning-Based Intelligent Controller. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(6), pp. 630–638.

Sharma, M. K. and Kumar, A. (2015). Performance Comparison of Brain Emotional Learning-Based Intelligent Controller (BELBIC) and PI Controller for Continually Stirred Tank Heater (CSTH). In K. Maharatna, K. G. Dalapati, K. P. Banerjee, K. A. Mallick and M. Mukherjee, eds., *Computational Advancement in Communication Circuits and Systems: Proceedings of ICCACCS 2014*, *Lecture Notes in Electrical Engineering*, vol. 335, New Delhi: Springer India, pp. 293–301.

Shim, J. and Arkin, R. C. (2012). Biologically-Inspired Deceptive Behavior for a Robot. In T. Ziemke, C. Balkenius and J. Hallam, eds., *From Animals to Animats 12*, *Lecture Notes in Computer Science*, vol. 7426, Heidelberg: Springer, pp. 401–411.

SoftBank Robotics (2017). *Discover Nao, the little humanoid robot from Aldebaran, SoftBank Robotics*. SoftBank Robotics, Available from `https://www.ald.softbankrobotics.com/en/cool-robots/nao` [Accessed 05 August 2017].

Spellman, T., Rigotti, M., Ahmari, S. E., Fusi, S., Gogos, J. A. and Gordon, J. A. (2015). Hippocampal-prefrontal input supports spatial encoding in working memory. *Nature*, 522(7556), pp. 309–314.

Steunebrink, B. R., Dastani, M. and Meyer, J.-J. C. (2006). Emotions as Heuristics in Multi-Agent Systems. In *Proceedings of the 1st Workshop on Emotion and Computing*, Bremen, pp. 15–18.

Stuchlik, A. (2014). Dynamic learning and memory, synaptic plasticity and neurogenesis: an update. *Frontiers in Behavioral Neuroscience*, 8(106).

Subagdja, B. and Tan, A.-H. (2015). Neural modeling of sequential inferences and learning over episodic memory. *Neurocomputing*, 161(C), pp. 229–242.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*, vol. 1. MIT press.

Thenius, R., Zahadat, P. and Schmickl, T. (2013). EMANN – a model of emotions in an artificial neural network. In *Proceedings of the 12th European Conference on Artificial Life*, Taormina, pp. 830–837.

Timmis, J., Neal, M. and Thorniley, J. (2009). An Adaptive Neuro-Endocrine System for Robotic Systems. In *IEEE Workshop on Robotic Intelligence in Informationally Structured Space*, Nashville, pp. 129–136.

Twining, R. C., Vantrease, J. E., Love, S., Padival, M. and Rosenkranz, J. A. (2017). An intra-amygdala circuit specifically regulates social fear learning. *Nature Neuroscience*, 20(3), pp. 459–469.

Vargas, P., Moioli, R., Castro, L. N., Timmis, J., Neal, M. and Zuben, F. J. V. (2005). Artificial Homeostatic System: A Novel Approach. In M. S. Capcarrère, A. A. Freitas, P. J. Bentley, C. G. Johnson and J. Timmis, eds., *Advances in Artificial Life*, *Lecture Notes in Computer Science*, vol. 3630, Berlin, Heidelberg: Springer, pp. 754–764.

Vargas, P. A., Fernaeus, Y., Lim, M. Y., Enz, S., Ho, W. C., Jacobsson, M. and Aylett, R. (2011). Advocating an ethical memory model for artificial companions from a human-centred perspective. *AI & Society*, 26(4), pp. 329–337.

Vazquez, J. and Malcolm, C. (2004). Distributed Multirobot Exploration Maintaining a Mobile Network. In *Proceedings of the 2nd International IEEE Conference on Intelligent Systems*, vol. 3, Varna, pp. 113–118.

Wagner, A. R. and Arkin, R. C. (2011). Acting Deceptively: Providing Robots with the Capacity for Deception. *International Journal of Social Robotics*, 3(1), pp. 5–26.

Weiss, A. and Bartneck, C. (2015). Meta Analysis Of The Usage Of The Godspeed Questionnaire Series. In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, IEEE, pp. 381–388.

Whiteson, S., Kohl, N., Miikkulainen, R. and Stone, P. (2003). *Evolving Keepaway Soccer Players through Task Decomposition*, Berlin, Heidelberg: Springer, *Lecture Notes in Computer Science*, vol. 2723. pp. 356–368.

Yang, S. N., Tang, Y. G. and Zucker, R. S. (1999). Selective Induction of LTP and LTD by Postsynaptic [Ca2+]i Elevation. *Journal of Neurophysiology*, 81(2), pp. 781–787.

Annex I

Drools Rules of the Hippocampus Module

```

package uk.ac.kent.cs.knowledge.situation

// ===== IMPORT ============================== //
import uk.ac.kent.cs.Hippocampus;
import uk.ac.kent.cs.knowledge.event.*;
import br.ufes.inf.lprm.scene.base.SituationHelper;
import br.ufes.inf.lprm.situation.SituationType;
import br.ufes.inf.lprm.scene.base.evaluators.*;
import java.util.ArrayList;

// ===== GLOBAL ============================== //
global Hippocampus hippocampus;

// ===== EVENT =============================== //
declare Event
   @role(event)
end

declare Adrenaline
   @role(event)
end

declare Flag
   @role(event)
end

// ===== RULE ================================ //
rule "Neutral Situation Instantiation"
@role(situation)
```

```
@type(NeutralSituation)
when
  $flag : Flag()
  not UnconditionedAversiveSituation(active)
then
  SituationHelper.situationDetected(drools);
end

rule "Neutral Situation Projection"
when
  not UnconditionedAversiveSituation(active)
  $neutral : NeutralSituation(!active, !projected)
  $events : ArrayList (size > 0) from collect (Event(this during
      $neutral))
then
  $neutral.addEvents($events);
  $neutral.projectAs("neutral");
  update($neutral);
end

rule "Unconditioned Aversive Situation Instantiation"
@role(situation)
@type(UnconditionedAversiveSituation)
when
  $adrenaline : Adrenaline (level >= hippocampus.getAdrenalineThreshold())
then
  SituationHelper.situationDetected(drools);
end

rule "Safe Situation Projection"
when
  $neutral : NeutralSituation(projected)
  $seclastneutral : NeutralSituation(projected, this after $neutral)
  $lastneutral : NeutralSituation(projected, this after $seclastneutral)
then
  neutral.projectAs("safe");
  retract($neutral);
end

rule "Conditioned Aversive Situation Projection"
salience 20
when
  $ua : UnconditionedAversiveSituation (active)
  $neutral : NeutralSituation(projected, this before $ua)
then
  $neutral.projectAs("aversive");
```

```
    retract($neutral);
end

rule "Neutral Situation Retraction"
salience 30
when
    $ua : UnconditionedAversiveSituation(active)
    $neutral : NeutralSituation(!active, !projected)
then
    retract($neutral);
end

rule "Unconditioned Aversive Situation Retraction"
when
    $ua : UnconditionedAversiveSituation(!active)
    exists UnconditionedAversiveSituation(active)
then
    retract($ua);
end
```