Technical Report 01-2008

A Computational Model of Robotic Surprise.

Alex Juarez

January 2008

Publisher: Dean Prof. Dr. Kurt Ulrich Witt

University of Applied Sciences Bonn-Rhein-Sieg,
Department of Computer Science

Sankt Augustin, Germany

Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

ISSSN 1869-5272

ABSTRACT

The research of autonomous artificial agents that adapt to and survive in changing, possibly hostile environments, has gained momentum in recent years. Many of such agents incorporate mechanisms to learn and acquire new knowledge from its environment, a feature that becomes fundamental to enable the desired adaptation, and account for the challenges that the environment poses. The issue of how to trigger such learning, however, has not been as thoroughly studied as its significance suggest. The solution explored is based on the use of surprise (the reaction to unexpected events), as the mechanism that triggers learning.

This thesis introduces a computational model of surprise that enables the robotic learner to experience surprise and start the acquisition of knowledge to explain it. A measure of surprise that combines elements from information and probability theory, is presented. Such measure offers a response to surprising situations faced by the robot, that is proportional to the degree of unexpectedness of such event.

The concepts of short- and long-term memory are investigated as factors that influence the resulting surprise. Short-term memory enables the robot to habituate to new, repeated surprises, and to "forget" about old ones, allowing them to become surprising again. Long-term memory contains knowledge that is known *a priori* or that has been previously learned by the robot. Such knowledge influences the surprise mechanism, by applying a subsumption principle: if the available knowledge is able to explain the surprising event, suppress any trigger of surprise.

The computational model of robotic surprise has been successfully applied to the domain of a robotic learner, specifically one that learns by experimentation. A brief introduction to the context of such application is provided, as well as a discussion on related issues like the relationship of the surprise mechanism with other components of the robot conceptual architecture, the challenges presented by the specific learning paradigm used, and other components of the motivational structure of the agent.

ACKNOWLEDGMENTS

I would like to thank Prof. Dr. Erwin A. Prassler for his constant support, motivation, dedication and advice that inspire not only me, but the team that has the fortune to work with him. Many thanks also to Prof. Dr. Paul G. Plöger for co-supervising the thesis, and for taking the time to discuss and review it. Special thanks to Timo Henne for all the after-hours scientific discussions and arguments that inspired, challenged and improved many of the ideas now realized in this work.

To my parents Alba and Nelson Juarez, for teaching me that any dream is possible, when there is enough courage to go all the way to the end. *"Never stand on the shoulders of giants, but be a giant yourself"*.

To my fianceé Anna Stawiarska, for showing me how to be so close, being so far. You have been an endless source of support, motivation, inspiration and love all this time, specially when things seemed to totally go the wrong way. *"After all this time... "*.

Finally, any success achieved in life is never the work of a single person, but of a multitude that contributed in one way or another. This thesis is not an exception, and several more pages could be filled in with acknowledgments. My gratitudes go then, to all those that have not been named, but constantly helped me to realize this goal.

The work described in this thesis has been partially funded by the European Commissions Sixth Framework Programme under contract no. 029427 as part of the Specific Targeted Research Project XPERO (Robotic Learning by Experimentation)

CONTENTS

Chapter 1

INTRODUCTION

1.1 Motivation

The research of autonomous artificial agents that adapt to and survive in changing, possibly hostile environments, has gained momentum in recent years. Most of such agents incorporate mechanisms to learn and acquire new knowledge from its environment, a feature that becomes fundamental to enable the desired adaptation, and account for the challenges that the environment poses. The issue of how to trigger such learning, however, has not been as thoroughly studied as its significance suggests.

For a truly autonomous artificial agent that learns (a *robotic learner*), the trigger of learning should originate from the agent itself, either from an internal drive or from the agent's response to a stimulus in its environment. Nature has solved this problem by providing living agents with such mechanisms, in the form of motivations, emotions and drives, therefore it seems appropriate to draw inspiration from them, while designing artificial ones.

Surprise is considered by many psychologists as one of the basic emotions that a human being possesses. Drive theorists consider it as either a primary (automatic) or secondary drive for exploration, while other approaches tend to view it as a source of conflict, a rupture of the internal stability that initiates a search for an explanation to the surprise, a curiosity that is satiated when stability has been regained [Ber60, AB83]. Despite the different opinions regarding its nature, it is generally accepted that surprise plays a powerful role in the learning process of a human being, affecting the learner's performance, and in some cases, determining the course of learning [CF05].

This strongly suggests that a similar mechanism can be applied to stimulate the experimentation and learning process in a robotic learner. Surprise, however, may not be the only mechanism involved, but a crucial part of a set of cognitive inspired techniques and technologies that deliver the experience of desire for knowledge to the robotic learner.

1.2 Overview of the State of the Art

The study of surprise started with cognitive researchers who wanted to better understand the human mind, by studying its different manifestations. These scientists rapidly developed theories that were translated into *cognitive models of surprise*. Most models assumed an internal knowledge representation that allowed to create beliefs about a specific event. The models also agreed that surprise is the product of a disconfirmed expectation, this is, an unexpected event. The evolution of the field gave rise to many heterogeneous approaches: being surprise an intangible element, the different measures of surprise developed were largely subjective, and based on many different, and sometimes orthogonal criteria.

Cognitive models of surprise were rapidly formalized, in the mathematical sense, by *computational models of surprise*. These models tried to provide a mathematical foundation that allowed to reproduce artificially, the response of biological surprise towards an unexpected situation. While such models also agreed that surprise came from unexpected situations and disconfirmed beliefs, they followed the path of their cognitive predecessors: most computational models of surprise were based on one or a couple of cognitive models, trying to be as faithful as possible to their biological counterparts, also resulting into a largely heterogeneous body of approaches to artificial surprise.

The recent interest of artificial intelligence in alternative motivation and control mechanisms for artificial agents, resulted in researchers paying attention to cognitively inspired techniques, such as artificial surprise. Several approaches to artificial surprise appearing in varied fields such as evolutionary and developmental robotics, social agents, and human-machine interaction, confirm that surprise occupies a place among popular techniques used nowadays.

When it comes to the application of artificial surprise to robotic learners, however, there are **deficiencies** that have not been addressed as thoroughly as they deserve. Most notably, the use of available knowledge acquired by the robot, in order to explain a surprising situation has been largely neglected by most approaches to surprise. Furthermore, the use of surprise as an instigator for learning new concepts has not been yet covered to an appropriate extent, by existing approaches.

The following sections present the problem addressed in this thesis, as well as the solution given. A more detailed presentation of the state of the art can be found in Chapter 3.

1.3 Problem Statement

The problem investigated in this thesis, is that of defining an intrinsic mechanism to be provided to a robotic learner, for it to autonomously initiate the learning and acquisition of new knowledge, a fundamental feature to survive in a constantly changing environment. The solution explored is based on the use of surprise (the reaction to unexpected events), as the mechanism that triggers learning.

A key task to be carried out while achieving such solution, consists of creating a computational model of robotic surprise. The state of the art presented previously, revealed that an appropriate model that suits the requirements of the robotic learner is greatly needed. Such model must not only define an appropriate measure of surprise that denotes the unexpectedness of an event, but also account for the use of knowledge available to the robot, when triggering a new learning process aimed to obtain insights about the surprise.

This thesis gives an answer to burning issues derived from the proposed solution, namely, which measure of surprise to use and how to implement it, what other components must be present in the model, how does the available knowledge influence the output of the surprise mechanism, what are the knowledge representations that will be used/included in the model and its implementation, and how the model integrates in the framework of the robotic learner.

All these issues present important challenges, whose answer lead to the use of probabilistic and information theoretic measures, the use of concepts such as short- and long-term memory, habituation to, and suppression of surprises, and the use of qualitative and first order logic models as knowledge representations.

The solutions to such problem and challenging issues, however, would be meaningless without providing a satisfactory answer to the question of how well the model performs on a real robotic learner. This thesis answers that by implementing the model into a robotic learner, that follows a paradigm of learning by experimentation.

Delimitation

This thesis draws inspiration from cognitive research to propose a computational model of surprise that is applicable to robots, especially for those robots that learn, however, it is not concerned or bounded by cognitive and/or biological plausibility. Additionally, it does not claim to provide an exhaustive search of cognitive models of computational surprise that have been applied to problems not related to robotics or to robotic learners.

The state of the art research in cognitive robotics suggests that surprise should be combined with other kinds of emotions and/or drives such as novelty, anger, fear or curiosity, as is presented in Section 3.3. The combination of different drives and emotions into a coherent motivation system is out of the scope of this thesis, however, a brief discussion is presented, providing ideas and concepts that can be useful in a future conceptualization and implementation of such system.

Chapter 2

APPLICATION OF SURPRISE TO LEARNING BY EXPERIMENTATION

Vast effort has been spent in the past decades to study mechanisms that enable an artificial agent to autonomously learn and acquire new knowledge. This is a fundamental characteristic if such agent has to survive in a changing and possibly hostile environment. Every new piece of knowledge acquired should be, at the same time, a source of information that helps the agent not only to specialize and avoid threatening, dangerous situations; but also to reason about problems/events it faces, and eventually evolve into new, different states of wellness.

On the other hand, providing a robot with the ability to compare its observations with its knowledge, enabling it to react to unexpected (surprising) situations or events in the environment, appears to be a valid way of addressing the problem of autonomously initiating the learning process mentioned before.

In this thesis we are interested not only in artificial surprise applied to a robotic learner, but in its application to a specific kind of learning paradigm, that involves the interaction between the agent and its environment: *learning by experimentation*. The following sections briefly describe the concepts associated to a robot that implements such paradigm, and how a surprise mechanism fits into that framework, as an initiator of learning.

2.1 A Robot that Learns by Experimentation

The computational model of robotic surprise has been applied to an intelligent autonomous robot that learns in an unsupervised manner, by interacting and experimenting with objects it finds in its environment, gaining new insights about the surrounding world and physical phenomena. This learning paradigm is known as *learning by experimentation*, and is a topic investigated by researchers in domains such as open-ended learning [PJH07] and planning [Gil94, CG87].

One fundamental difference between a robot that learns by experimentation, and other robotic agents using different learning approaches, resides in that the process that initiates and triggers learning come from the robot itself, and its interaction with the environment.

As nature has solved that issue by providing living beings with several kinds of stimulus and motivations, it seems advisable to take inspiration from it, when designing artificial motivation mechanisms.

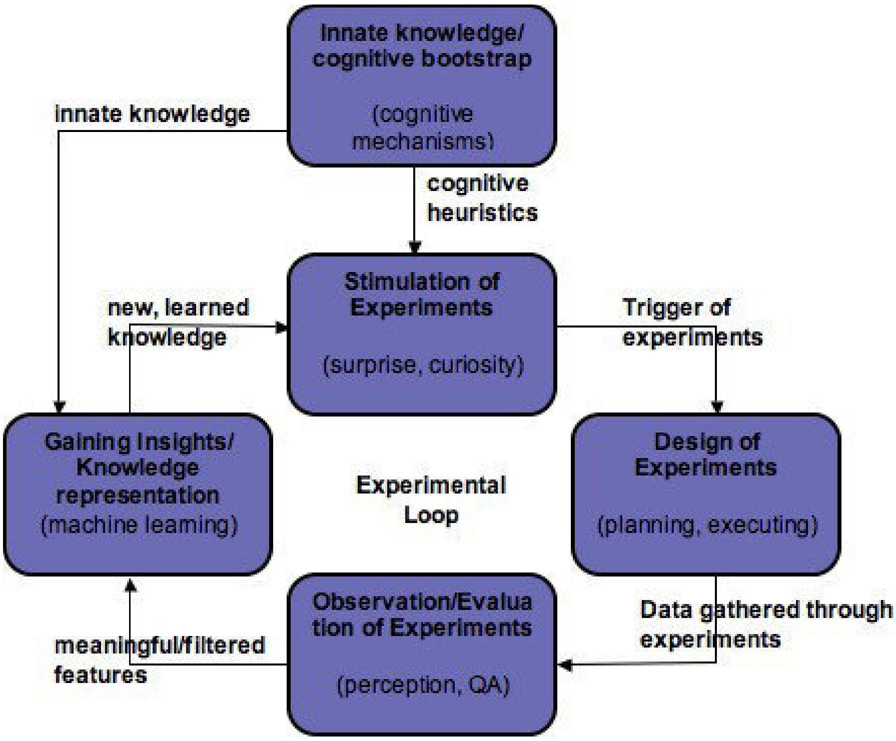


Figure 2.1: Conceptual architecture of the robotic learner used in the experiments.

The robotic learner that has been used as a testbed for the application of the model of robotic surprise, is part of the project XPERO [PJH07], which applies the paradigm of learning by experimentation in an *open-ended* learning domain. This means that the learning is not driven by a predefined target concept, but as a reaction to the robot *experience* in the *real world*; name it curiosity, surprise, or a more abstract and complex (*cognitive*) mechanism such as "hunger for knowledge".

Figure 2.1 shows a simplified version of the conceptual architecture of such agent. A *stimulation of experiments* (SoE) component provides the motivation for the robot to start the learning process. The design of experiments (DoE) component is in charge of design, plan and issue execution of experiments aimed to gather data on a specific event, which is validated by an *observation/evaluation* (OE) mechanism, to ensure that insights can be extracted from it. The *learning* component takes such information and tries to extract knowledge from it, assisted by the *innate knowledge/cognitive bootstrap* module, which provides cognitively inspired mechanisms that might help in discovering new insights.

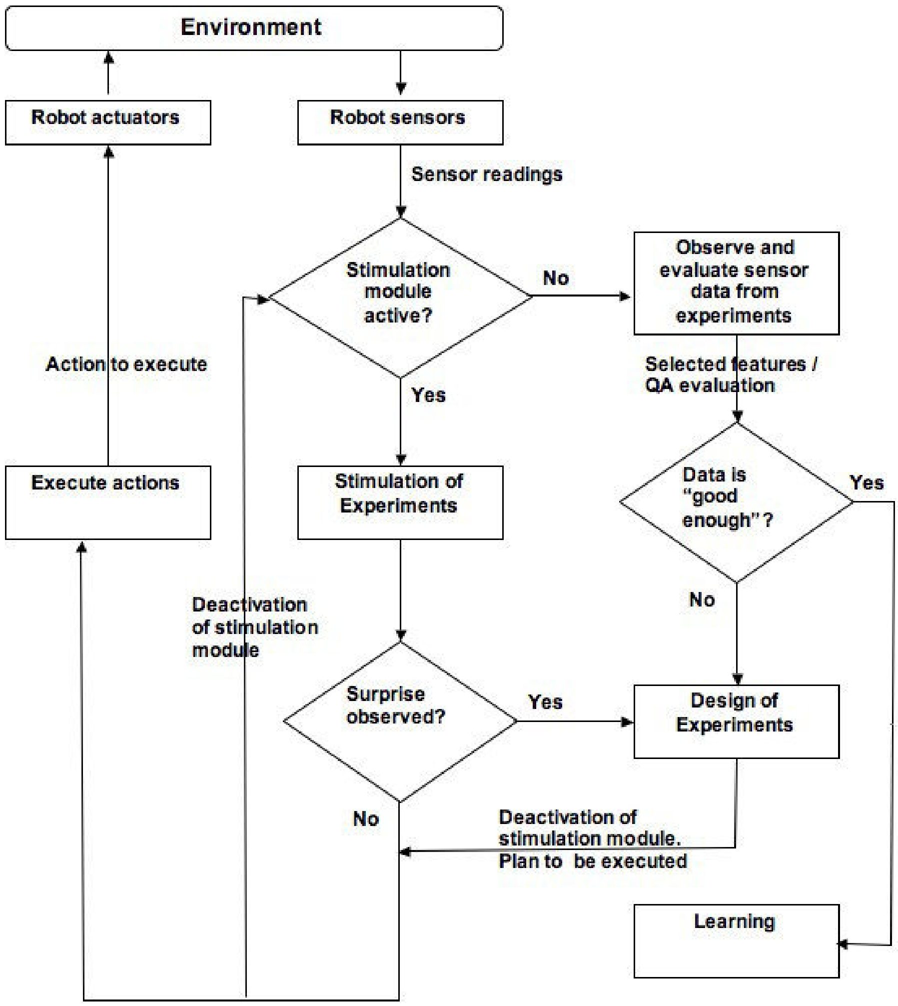


Figure 2.2: Diagram describing the flow of information between different components of the architecture of the robotic agent.

A closer look to the experimental loop process is shown in the flow diagram of Figure 2.2. While the robot is performing its "everyday actions", this is, executing whatever task it has been programmed to do as a default, it is likely to face unexpected situations. The robot's response to these situations is given by the *stimulation of experiments* (SoE) component.

The SoE component receives data from the robot's sensors and if surprise is triggered, it communicates with the DoE component, providing information on the unexpected event.

DoE designs, plan and issues the execution of experiments aimed to gather data on that situation. At the same time, it disables the SoE component to allow the robot to focus in the experimentation and gathering of data without being distracted by other events.

The robot executes the experiments and passes the result to the OE component, to be validated and evaluated, determining if the data gathered is "good enough", such that there exists the possibility to obtain any insight from it. In case that there is a need to collect more data, the OE component communicates again with DoE, for it to either design new experiments or reissue the execution of the previous ones.

Once the data has been thoroughly revised, it is passed to the *learning* component of the robot which, using traditional Machine Learning techniques, tries to obtain insights about the experience of the robot. The computational model of robotic surprise proposed in this thesis is an integral part or the *stimulation of experiments component described.*

2.2 Running Example

Section 2.1 presented an overview of the framework of a robotic agent that learns by experimentation. This section describes an example that utilizes such agent as a framework for the concepts and implementation of the model of surprise.

The robotic agent used is able obtain information on its position in the world, its contact with another object, and the position of other objects. Additionally, it has a list of predefined actions that it can execute. Such actions can be single (e.g. move forward for a specific time) or they can be composed of a sequence of actions (e.g. going to a destination, which may imply that the robot has to turn, and then move forward, until the desired location is reached). Table 2.1 shows a list of the actions available to the robot, and their corresponding command.

The running example consists of two scenarios:

1. The robotic learner starts executing one of its predefined actions (e.g. moving forward). An object (a box that the robot is able to move), is on the path of the robot, so that the robot bumps into it, and starts pushing it, resulting in a joint motion.
2. The robotic learner starts executing the same action (e.g. move forward), but this time the object (the same movable box), is not in its path. The robot, then, passes the object by, and continues moving until it eventually collides with a wall.

Being a robot that learns by experimentation, this two scenarios and the surprises that the agent might experiment while executing them, should initiate the process of experi-

Table 2.1: List of commands available to the robotic agent.

Command	Action
MOVE_FORWARD	Move forward a specific distance.
MOVE_BACKWARD	Move backward a specific distance.
TURN_IN_PLACE	Turn in place a specific angle.For simplicity, the robot turns only left, thus only positive angles are permitted.
GO_IN_CONTACT	Look for a specific object and move towards it until it touches the object.
GO_TO	Go to a specific position in global coordinates.

mentation and acquisition of knowledge that explains the surprising situation, giving the robot “experience” about it in case it reoccurs in the future.

This example will be used whenever possible through the remainder of this thesis, and should serve to demonstrate the validity of the computational model of surprise applied to a robotic learner with limited knowledge about the world. It must be noted that the example does not cover the previously described experimentation and learning processes that apply to this specific learning paradigm (*learning by experimentation*). This does not affect the validity of the approach in any case, but carries along an invitation to have in mind the context where the model of surprise is applied, as well as the final objective of the robot (acquiring knowledge that improves its own performance when facing the same situation again), when analyzing the results obtained from the different methods presented.

Chapter 3

RELATED WORK

Surprise has been largely studied by cognitive science. Psychologists studying the structure of the human mind have often defined it as a state of mind, a result of unexpected events that manifests itself as a "feeling", and that is observable in the appearance of behavioral indicators such as the interruption or delay of ongoing motor activities. Surprise is an instigator of investigative activities, and a cause of spontaneous physical expressions [Rei07].

Results of such research are theories of surprise that intend to describe the mechanisms behind the feeling and behaviors related to surprise. Such theories gave birth to models of surprise both of cognitive and computational kind. The following sections present cognitive and computational surprise models that are closely related and/or have been applied in the fields of artificial intelligence and robotics.

3.1 Cognitive Models of Surprise

Kahneman and Tversky [KT81] analyze surprise from the perspective of perceptual expectations. Expectations are mainly classified as active and passive. Active expectations are a conscious process that draws attention to a specific event, while passive expectations are automatic and effortless process that can be described better as a disposition than as an activity. Surprise is generated when the actual perception does not match an expectation (active or passive) that exhibits a high confidence, this is, a high degree of belief associated to it. Additionally, a single event can have both passive and active expectancies associated which in turn can be conflictive, as is the case when a consciously predicted event occurs and a physiological indicator of surprise is also evident. In this case, surprise is determined by the "weights" that the mind assigns to such expectations.

Ortony and Partridge [OP87] propose a model of surprise that follows the concept of expectancy generation and failure. The model proposes that expectations are encoded in the knowledge database as propositions. Such propositions can be either explicitly represented, or readily deducible from others that are explicit. Surprise is elicited when an input proposition (perception) conflicts with a proposition that is present in the knowledge base. If the existing proposition is consciously entertained, it is called active prediction, while if it is not, it is called passive expectation (this concept is similar to the one in the model by Kahneman and Tversky). As an example it is quoted from [OP87]:

> ... if one actively expects that the Democratic candidate will win the next presidential election, this prediction is rooted in the propositions that there is to be an election, and that there is a Democratic candidate. If the Republican candidate wins, one would be surprised as a result of the failure of an active expectation [...] However, if, while having the same database, one has never made such inference [...] the surprise would be the result of the failure of a passive expectation.

Additionally, a third kind of surprise that do not arise from any conflict with active predictions or passive expectations, is explained. In this case, there is not a proposition (explicit or deducible) that can be used to compare the perceptions. Surprise then, may only arise from a conflict between the input proposition and what, after the fact, may be judged to be normal or usual.

Meyer et al. [MRS97] view surprise as a mechanism which function is to enable and provide initial motivational impetus for event analysis and belief updates. Their model of surprise is formulated within a schema-theoretic framework in the context of action selection. It proposes a knowledge representation based on schematas (schemas), a complex structure that control perception, thought, action and emotion, and that can be selectively activated by different events. Surprise is defined then as a discrepancy between the active schema and the perceptual input detected. The evaluation of the surprising event is performed largely in an effortless, unconscious, automatic way, and is composed by four subprocesses namely: the verification of the schema discrepancy, the analysis of the causes of the unexpected event, the evaluation of the unexpected event's significance for the well-being and the assessment of its relevance for ongoing action. The result of a surprising event is the interruption of ongoing information processing and reallocation of processing resources, leading to a schema update.

Based on the model by Meyer et al., Reisenzein [Rei00] explores the subjective, conscious aspect of surprise. Surprise is seen as a "feeling", a centrally generated signal caused directly by the appraisal of unexpectedness. Unlike Kahneman and Tversky [KT81] and Ortony and Partridge [OP87] where the knowledge representation is of propositional nature that is either explicit or can be inferred from explicit ones, Reisenzein proposes that surprise is a special purpose, hardwired mechanism that outputs a "signal" which intensity represents the degree of discrepancy of the perception and expectations. When the intensity of the signal exceeds a threshold, it is felt as a possibly unique conscious quality of the perception of the event at hand. Table 3.1 shows a comparative table for the cognitive models of surprise.

Table 3.1: Cognitive models of surprise.

expectation	**Knowledge representation**	**Surprise definition**
onsciously ed) and Passive ic and effortless). .	Propositions with associated beliefs (probabilities).	Discrepancy between perception and expectations
xplicit propositions ase), Passive ions inferred from nes).	Propositions either explicit or inferred from explicit ones.	Conflict between input (perceptions) and expected (in knowledge database) propositions. Additional surprise from discrepancy between perceptions and what is considered normal (not related to active/passive expectations).
nd of expectation in an automatic tless way.	Schematas that control perception, action, thought and emotion.	Conflict between perceptions and schemas.
ions are consciously ed. Awareness of s given by this .	Schematas that control perception, action, thought and emotion.	High intensity on the signal that measures discrepancy of perceptions and schemas

Finally, masuring surprise is a challenging task given its subjective and emotional nature. Psychologists have devised several methods for it, for example, Meyer et al. [MRS97] measures the reaction time of the subjects when presented with a surprising (discrepant) event, adding a rating of the degree of surprise obtained by asking the opinion of the test subjects about it; Maguire et al. [MCK06, MK06] measures perceived surprise by asking test participants to differentiate between the level of predictability of different text sentences where forward inference (anticipate the upcoming result of the situation described by current and previous text) was possible. The results obtained by these and many other methods applied to obtain a measure of surprise, indicate that there is no unique and/or standard way of measuring this emotions in human beings.

3.2 Computational Models of Surprise

Computational models of surprise have been created by formalizing and implementing cognitive ones. Peters [Pet98] presents a model that encodes surprise in perceptual robots, aiming to produce creative and intelligent behavior. The model shares the idea of surprise as a two-component phenomenon: an expectation set on the basis of previous experience, and a departure, which is the discrepancy from what is projected to happen and what actually did happen. Surprise is defined as the absolute value of the difference between an input signal and a prediction value (see Equation 3.1). The prediction value is calculated as the ratio of what is stored in memory to the persistence of that memory (see Equation 3.2).

$$surprise = |signal - prediction| \tag{3.1}$$

$$prediction = memory \;/\; persistence \tag{3.2}$$

Additionally, Peters acknowledges that the detection of surprise must be performed at multiple levels. To account for this, a multi-layer surprise generator is proposed. Each layer tries to generate surprise and at the same time filters the input signal, removing superficial frequencies, and passing it to the next layer of abstraction. In this way, surprise can be detected at several levels of abstraction at a time.

Macedo and Cardoso [MC99] propose a model of human surprise for perceptual artificial agents. Such agent should become surprised when an unexpected part of its environment shows up, and should become curious about unknown things of the world. The mental representation used is graph based, and of semantic and episodic kind. This allows to represent episodes and cases (episodic knowledge) and theoretical knowledge (semantic knowledge). The measure of surprise is given by a measure of the novelty of an object, and the measure of the degree of unexpectedness of such object. The former is obtained by a mathematical formula that computes the distance between two graph-based representations (perceived and stored in memory) of the object, while the latter computes the probability of existence of a perceived object.

$$Surprise(Agt, Obj) = f(\text{Deg\_of\_not\_Expecting}(Obj, Agt(Memory)), arg_1, arg_2, ..., arg_n) \tag{3.3}$$

This is depicted in Equation 3.3 where the surprise that the agents“feels” is a function of the degree of not expecting that object with respect to the object models stored in the agent’s memory.

In later work, Macedo and Cardoso [MCR06, MC01b, MC01a] propose that surprise is integrated in the motivation mechanism of an artificial agent. Additionally, the authors propose a cognitive architecture composed of the components sensors/perception, memory, goals/desires, reasoning/decision making and motivations. Figure 3.1 illustrates such architecture and the relationship between its components.

Though in their original model of surprise [MC99] they propose a memory of both se-

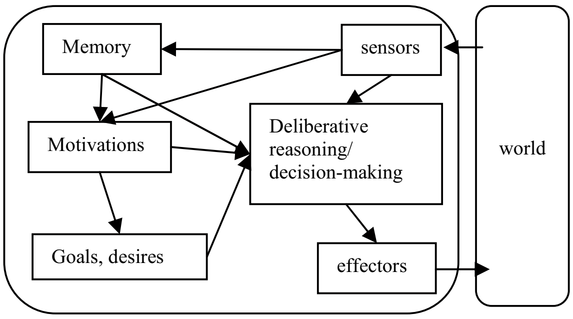


Figure 3.1: Surprise-based agent architecture presented by Macedo and Cardoso [MCR06].

mantic and episodic kind, the later model and its implementation considers only episodic knowledge, this is, an input proposition (or new belief) is compared with episodic representations of objects or events (or their properties). An analysis of different computations of the measure of surprise that include information theory and probability theory, and can be used for specific domains is presented. Episodic memory and its implementation is revised to assess its effects in the surprise value obtained with the mechanism mentioned before.

Lorini and Castelfranchi [LC07] present a model of surprise that can be integrated with a model of belief change. This approach claims that surprise is a belief-based phenomenon that relies in an actual or potential prediction, and which main effect is the revision of assumptions: new data must be assimilated in the knowledge base and belief associated must be revised. In this context, the authors propose two kinds of surprise: mismatch-based and astonishment.

Mismatch-based surprise is related to a conflict between the perceptions and a scrutinized knowledge representation, this is, a highly anticipated expectation of an event which is checked constantly. Astonishment is given by the recognition of the implausibility of a perceived event. Such implausibility is given by two causes: a perception corresponds to a fact that was not actively expected and that can not be retrieved from the background information (representations and expectations available in memory that work at an automatic level); and a conflict with a representation that has been inferred from the background knowledge (a similar concept as the one in Orthony and Partridge model). Figure 3.2 illustrates the idea behind this model.

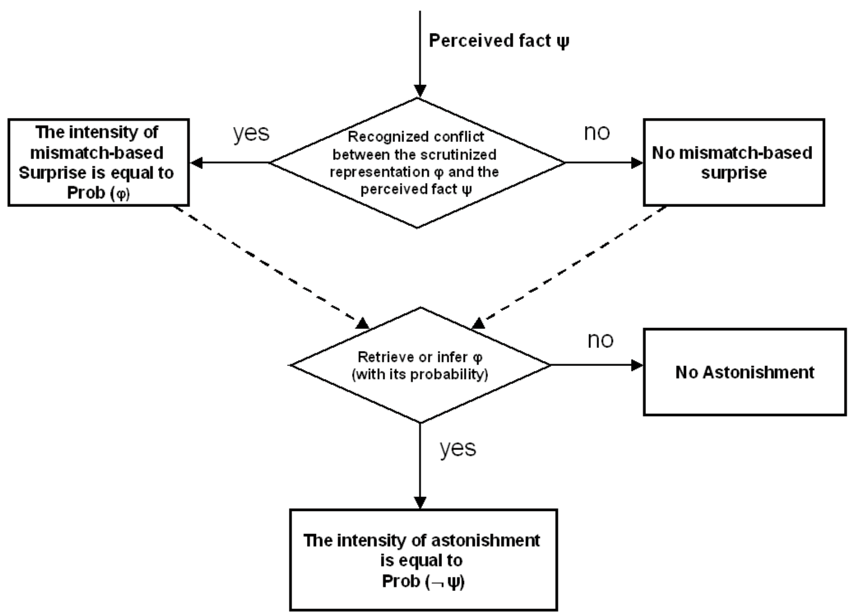


Figure 3.2: Computational model of surprise proposed by Lorini and Castelfranchi [LC07].

Itti el al. [IB05, IB06] propose a model of surprise based on an information-theoretic measure, but at the same time, incorporates elements from probability and decision theory that allow a level of reasoning about the uncertainty associated to surprise. In other words, they propose a definition of surprise that involves a) probabilistic concepts to cope with uncertainty, and b) prior and posterior distributions to capture subjective surprises.

Both objectives are accomplished by using a bayesian approach to estimate posterior distributions, over a set of hypothesis or models in space. To measure surprise, the Kullback-Leibler divergence of probability distributions is used. Such measure quantifies the distance, or dissimilarity, between the prior and the posterior probability distributions.

Table 3.2 summarizes the different computational models of surprise presented in this section.

3.3 Surprise and Motivation for Robots.

The application of emotional mechanisms to robots that interact with and adapt to their environment, has been addressed by researchers that try to combine knowledge from cognitive science and robotics. Emotions such as surprise, curiosity, anger or fear, have been used as motivations for diverse robot behaviors, such as search for food (survival), environment exploration (learning by interaction), belief revision and attention focusing, among others.

Gadanho and Hallam [GH01] present an approach where emotions are used to control an autonomous robot that adapts to its environment using reinforcement learning. The authors develop a model of robot control that combines four emotions: happiness, sadness, anger and fear. Additionally, a series of internal feelings (e.g. hunger, pain, restlessness) combine to define each emotion. The task of the robot is to collect energy from food sources through the environment. Emotions are then used to trigger state transitions for the control system of such robot: the intensity of the different feelings determines the intensity of different emotions. Reward functions are defined based on a policy of maximization of positive emotions and minimization of negative emotions.

Macedo and Cardoso [MC04] defend that in order to understand, build or model artificial agents that explore and interact with their environment like humans do, those agents must manifest at least curiosity and surprise. These should be the variables that directly instigate any exploratory process. Additionally, they propose that those emotions should be combined with the drive "hunger" that reflects the need of a power source. The motivations derived directly from emotions may be goals themselves, driving the exploratory activity to search for those states that result in a positive emotion (e.g. happiness or

Table 3.2: Computational models of surprise.

Model	**Kind of expectation**	**Knowledge representation**	**Surprise definition**
Peters	Prediction based on short term memory and the persistence of an object in such memory.	Signal encoded to a single value in memory.	Discrepancy between sign
Macedo and Cardoso	Degree of unexpectedness and degree of novelty of an object	Graph based, episodic representation of objects and events.	Degree of unexpectedness between the input and me of an object or event. Nov probability of existence of
Lorini and Castelfranchi	Scrutinized (highly anticipated) knowledge and Astonishment	Probability values assigned to different propositions.	Mismatch-based surprise (a scrutinized representatio of implausibility of a perc
Itti et al.	Prior probabilities assigned to hypothesis/models of the situation at hand.	Probability distributions that represent a degree of belief in the observed data.	Divergence (dissimilarity) probability distribution, th belief associated to the mo that is being observed.

surprise).

The problem of designing intrinsic motivation systems that allow active learning, resulting in a structure of the development of the robot has been addressed by Kaplan and Oudeyer [KO06]. They propose that such intrinsic motivation system is based in artificial curiosity and what is called meta-predictions (errors in the predictions made by the robot). In a later work [OKH07] several heuristics are proposed, in order to use the meta-predictions and trigger the process of learning: error maximization, progress maximization and similarity-based progress maximization. Learning is aimed to the prediction of sensori-motor activity of the robot, focusing in the problem of action selection: it intends to find out the next action that the robot must execute so that the error in the prediction is maximal, therefore maximizing the learning progress once it is learned to predict better the action.

Emotions also play a role when the aim is to build robots that are not only adaptive, but also responsive to social interactions. Velazquez [Vel99] proposes an emotion-based control for robots that integrates emotions, motivations, perception, attention and motor control. The function of emotions is to provide the motivational context to activate different behaviors, to facilitate attention focusing, to bias the perceptual system such that some stimuli are more attended or neglected, to establish appropriate emotional expressions, and to provide means by which the robot can learn from past emotional experiences. The emotional system is based in a combination of six emotions: anger, fear, distress/-sorrow, joy/happiness, disgust and surprise. The task of surprise is to deal with novelty, anticipatory expectancy, and to trigger reflex-like responses to stimuli.

In the same context, Breazeal [Bre98] presents a motivational system for an autonomous robot that is designed to regulate human-robot interaction. This system implements drives and emotions as components that regulate the interaction. Drives are processes that show a temporal cyclic nature that increases in the absence of stimuli unless it is satiated, e.g. hunger. Additionally, they influence the emotive state of the robot. Emotions, on the other hand, provide responses to specific stimuli. Its intensity provides the degree of change in the emotive state of the robot and trigger a series of processes that control facial expressions and the type of interaction needed with the human. Emotions used in this approach are anger, disgust, fear, happiness, surprise, interest and excitement.

3.4 Limitations

The state of the art reviewed in previous sections, reveals a large heterogeneity in the approaches to artificial surprise. However, few of such approaches addresses the needs of a robotic learner, in terms of when to start the learning process, and what to do with the new knowledge acquired.

Existing models of artificial surprise do not consider, for example, the use of complex knowledge structures, such as qualitative/quantitative models of physical phenomena (e.g. the behavior of a ball in free fall or rolling on the floor and colliding against a wall), that might be related to an unexpected event. If such knowledge exists, that can explain such seemingly surprising event, then the degree of surprise experienced by the robot should decrease considerably, or even be suppressed, given that it is very likely that nothing new could be learned from such experience.

How does such more abstract and powerful knowledge influences surprise, in a way that allows the robot to determine if it is meaningful to start a new learning process, is an issue that has not been addressed by existing approaches which, in most cases, limit themselves to acknowledge its importance, but choose to ignore it in the models.

Chapter 4

COMPUTATIONAL MODEL OF SURPRISE

This thesis proposes a computational model of robotic surprise that enables a robotic learner to experience and react to unexpected, surprising events, initiating a new learning process if available knowledge can not explain the robot observations. The model is based on two main components: a surprise mechanism in charge of processing the sensor data and reacting to surprises present therein, and a knowledge base that contains the results of the learning activity of the robotic learner. The interaction of the surprise mechanism with the knowledge base, has the purpose of utilizing such knowledge to influence the resulting measure of a surprise that has previously emerged from the sensor data.

This model of robotic surprise has four main characteristics to note:

- **Reacts to sudden, unexpected changes in sensor data**. A surprising event can only be recognized as such, if the robot perception is processed in a way that those events trigger an immediate response. This is achieved by a measure of surprise that is a function over sensor data. The response of such function varies its intensity depending on how surprising the event that produced the data is.

- **Considers the influence of short- and long-term memory**. The concept of memory, as a repository or mechanism which contains information that influences how and when surprise is detected, has been considered in the model of surprise.

  To illustrate this concept, consider the running example (see Section 2.2), specifically the scenario where the robot pushes an object. The contact of the robot and the object produces a surprise given by the bumper sensor suddenly activating, and the object moving. If this event has never been seen before, or has not been experienced in a long time, the likelihood of it occurring in the next instant its very low, therefore if it occurs it will be a surprise. Conversely, if such event has recently occurred, it will still be "fresh" in memory and the expectancy for it to re-occur in the next instant, will also be higher. Therefore, the level of surprise in that case will be much lower than in the previous one.

- **Makes use of existing/learned knowledge**. The existence of expectancies on the outcome of events, implies the use of knowledge, either acquired or a priori, to trigger surprise. Such knowledge comes in the form of beliefs and/or sensor models of the objects and their interaction with the robot's environment. The use of more complex models of the situations faced by the robot, can influence the trigger of surprise to a far greater extent than the information provided by the measure of surprise alone.

Consider now the scenario where the robot collides against a wall. Surprise will be triggered, as the robot is suddenly not moving anymore and it suddenly sensed a contact with something. However, if there exists a model of the world that tells the robot that this behavior is perfectly normal (e.g. a rule that states that the robot should not move if there is contact with anything), this event would not be surprising anymore. If on the other hand, the robot starts flying into the air after bumping, that event could not be explained by such more abstract model, and then a "real" surprise is triggered.

- **Takes into account the characteristics of the embodiment where the model will be implemented**. The kind of sensors that a robot is equipped with determines the kind of surprise that the robot may experience. Consider again the example of a robot that collides against a wall. If the robot does not possess any sensor/sensing mechanism that provides information on the robot motion (e.g. odometry, self-localization, etc.), it is impossible to obtain any surprise related to its movement. Though this affirmation may appear rhetorical, it is easily neglected while designing a surprise mechanism.

  A direct consequence of such constraints is a bias on the implementation of the surprise mechanisms: the selection of the sensor equipment determines how and which surprises can be recognized. Conversely, if there is a need to react to a specific kind of surprise, then the robot must be equipped with the appropriate sensors and knowledge/mechanisms to determine how to process the sensor data. While ideally the robot should react to any kind of surprise stimulus possible, based only on the sensor input data, in practice, a design decision must be taken regarding which surprises the robot will be able to react to, and how to implement them efficiently to ensure such reaction.

The outline of the model of robotic surprise is shown in Figure 4.1, which depicts the model components and the relations between them. Before the start of any action, the robot searches the *long-term memory* for related knowledge, loading it into special data structures where it will be available for the duration of the action. This knowledge includes information on how to process the sensor data, and more abstract knowledge structures, such as *first-order logic* and *qualitative models*, that influence the triggering of surprise.

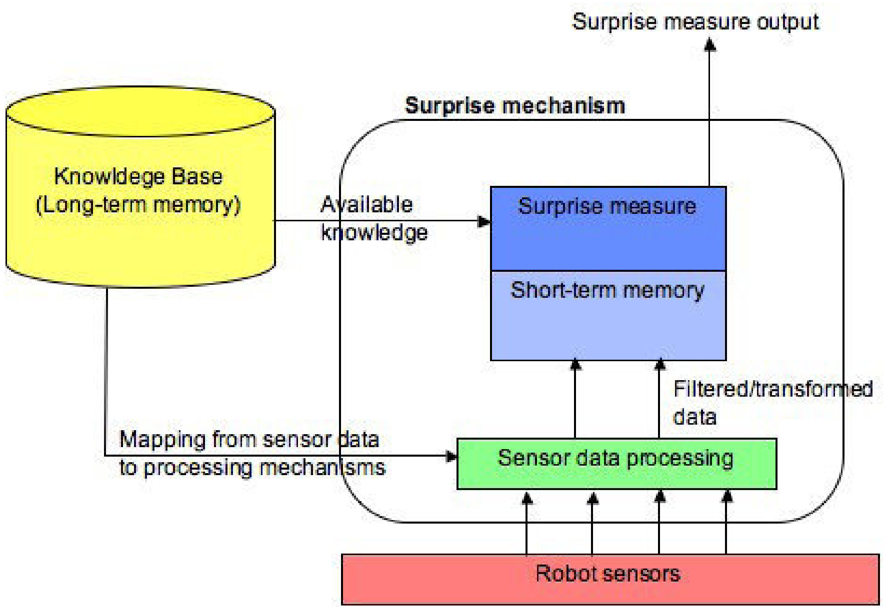


Figure 4.1: Computational model of robotic surprise.

Once the action starts, the robot sensors provide the surprise mechanism with information that may contain a potential indication of a surprising event. Such information is processed according to a mapping from sensor data to processing mechanisms (information previously obtained from the long-term memory). This association tells the robot how to transform and/or use the input data, in such a way that enables a reaction to surprise.

After that data has been processed, it is stored in the short-term memory. The measure of surprise has direct access to the short-term memory, and uses its contents to recognize sudden changes in the input data that correspond to surprises. Additionally, the measure of surprise uses the more abstract knowledge available to decide if the surprise observed can be explained by concepts previously learned/acquired by the robot, or if it is a “real” surprise.

4.1 Measure of Surprise

A measure of robotic surprise is a mechanism that allows to quantify the amount of surprise present in the sensor data, as a result of an event in the real world. Existing approaches propose the use of the “distance” between knowledge representations, the occurrence of highly informative events, or the divergence between the expected and the observed probability distribution for a given event/object. Most existing approaches can be divided into two groups: those that utilize purely information theoretic methods, and those that combine information and probability theory to provide a measure of surprise.

Purely information theory based measures of surprise use functions that measure how much information the observed data provides, e.g. entropy, mutual entropy or information gain. This implies the use of an *a priori* probability distribution associated with the possible events in the domain. The occurrence of an event with a low probability of being observed, provides a high information about a possible change in the environment. The value of the information gain is then taken as the measure of surprise. Most information theory measures are characterized by high sensitivity to noise in the input data, as well as the mentioned need for a prior knowledge of the probability distribution of the events in the domain. Information theory measures are known as *static*, as they do not consider a mechanism to adapt its knowledge (probability distribution) as new information is obtained. More information on variations of information theory measures, as well as an empirical study to relate the unexpectedness depicted by the measures, and the degree of surprise observed in humans can be found in [MC01a].

The second group of measures of surprise *combines information and probability theory* methods. Information theory measures such as relative entropy, are combined with estimation techniques such as Markov models, bayesian estimation, Kalman filtering/estimation, etc. to provide a *prediction* of the probability distribution associated to the event data. This prediction is a model that indicates the expected behavior, state or value of the input signal for the immediate future, given values observed previously (history). The information provided by the observed data with respect to this model, is taken as the measure that indicates how surprising the event is.

Most notably, the approach presented by [IB06], termed *Bayesian Surprise*, introduces a measure of surprise based on the divergence between a model probability distribution and the probability distribution of the observed signal. The model probability distribution is obtained gradually as new data is received. The modification of such distribution to account for new data is performed using a bayesian update. This approach has been successfully applied to react to surprise triggered by salient locations of an image.

The measure of surprise used in this thesis, takes the ideas proposed by [IB06] and applies them to the case of an autonomous robot, specifically, to a robotic learner. The basic concept behind the measure is to compare a model probability distribution of the sensory input, that is either given (*a piori*) or constructed from robot observations, with the probability distribution obtained from the observed data.

The mechanism used to measure the difference between two probability distributions, is the Kullback-Leibler divergence KLD. The KLD is a measure of the difference between a "true" probability distribution (a model) and an arbitrary probability distribution (observations). It was originally introduced by Solomon Kullback and Richard Leibler in 1951, and is also referred to as *information gain* or *relative entropy.* Equations 4.1 and 4.2 present its formulation for the discrete and continuous cases respectively, of two arbitrary probability distributions P and Q:

$$D_{KL}(P||Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)} \tag{4.1}$$

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} P(x) \log \frac{P(x)}{Q(x)} dx \tag{4.2}$$

If the probability distribution is known *a priori*, the application of the formula is straightforward. This is usually not the case for a robotic learner, as it will most likely have to face changing, possibly unseen situations, where the calculation of the distribution becomes a problem of fitting distributions to observed data. An alternative to overcome this problem is to assume that the data follows a known probability distribution. The approach presented in this thesis uses the Normal (Gaussian) distribution as a base probability model.

This selection offers several advantages: the Gaussian distribution is a parametric distribution based on two components, the mean and the variance, for which several well known estimation methods exist. This is extremely useful as the robot works with incomplete information of its domain, therefore making the estimation of parameters a necessity. Additionally, many unknown probability distributions can be approximated by a Gaussian or mixture of Gaussians, for which parameter estimation techniques also exist.

It is important to note that by assuming a probability distribution as a model for the sensor data, we are also introducing restrictions and limitations to the accuracy of such model. To illustrate what is meant, consider the case of the readings coming from a bumper sensor. This information behaves in a similar way as a boolean variable: true if there is contact with other objects, false if there is not. It is easy to see that the sensor readings obtained from a bumper as the robot drives around the world, would be better modeled by a binomial distribution.

The measure of surprise does not prevent the use of other probability distribution to model the observed data. In other cases (i.e. other sensors), the decision on which distribution to use might not be as easy and apparent as in the case of a bumper data. The successful results obtained in the experiments (see Chapter 5), suggest that modeling the sensor data using Gaussian distributions is feasible and valid, for the specific characteristics of the environment and embodiment used. This encouraged the use of a Gaussian distribution as a model for all the available sensor readings (which includes a bumper sensor in the robot).

The formula for the KLD for two Gaussians is shown in Equation 4.3, where X and Y are two Gaussian distributions and (μ_X, σ_X) and (μ_Y, σ_Y) are the parameters (mean and variance) of X and Y respectively.

$$D_{KL}(X||Y) = \frac{1}{2}(\log(\frac{\sigma_Y^2}{\sigma_X^2}) + \frac{\sigma_X^2}{\sigma_Y^2} + \frac{(\mu_Y - \mu_X)^2}{\sigma_Y^2} - 1) \tag{4.3}$$

The estimation of the parameters of a Gaussian distribution plays an important role, for it serves two purposes: it enables the calculation of the new probability distribution (the one that includes the observed data) prior to the comparison with the model distribution, and enables the update of the previous probability distribution after the comparison has been performed.

Mitchell [Mit97] suggests that a reliable method for predicting probabilities lies in the maximum likelihood estimation of hypotheses. This is equivalent to minimize the sum of squared error of the observations with respect to a prior hypothesis, which in bayesian terms is expressed by $h_{ML} = \arg\max p(D|h)$. This insight supports the selection of the maximum likelihood functions for a Gaussian distribution, as a valid mechanism to estimate the new probability distribution, based on previously observed data (previously estimated models).

Maximum likelihood estimation (MLE) allows to determine the parameters that maximize the probability (likelihood) of the sample data (sensor readings), in the case of uninformed prior probabilities. The implementation used in this thesis also assumes that the measurements used in the process are independent, and that it produces an estimation that is normally distributed, with expectation μ and variance $\sigma^2 > 0$. Equations 4.4 and 4.5 show the formulas used to obtain the ML estimate (mean and variance) in the discrete case. Here, x_i represents each observed (sensor) value, n represents the number of observations used to obtain the estimate, and $(\hat{\mu}, \hat{\sigma}^2)$ are the estimated parameters of the Gaussian distribution.

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.4}$$

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu})^2 \tag{4.5}$$

The process of measuring surprise is performed in three steps: estimation of the new (observed) probability distribution, comparison with the model probability distribution and update of the model probability distribution.

1. **Estimation of observed probability distribution**. As the robot sensors retrieve information, an estimation of the parameters of a Gaussian distribution that fits the data is performed. The ML estimate offers a good approximation requiring a relatively low number of sensor readings. This property is affected by the quality of the data received from sensors, this is, the amount of noise present, and the representative quality of the measures with respect to the underlying (possibly unknown) probability distribution.

2. **Comparison of observed and model probability distributions**. Once the observed probability distribution has been calculated, it is compared with the model probability distribution using the KLD measure. In case of analyzing multiple sensor information at the same time, e.g. surprises in a bumper sensor and camera images, the magnitude of the final surprise is calculated by using a proportional voting system.

3. **Update of the model probability distribution**. Once the comparison between the observed and model probability distribution has been made, the model probability distribution is updated to take into account the new information received. By this, the model is also being prepared to be used the next time the surprise mechanism becomes active. The reason for this update derives from the lack of information regarding the prior model distribution: if no prior information is available, the model must also be constructed as the robot receives input from its sensors. This also means that the observed distribution at one point in time, is incorporated into the model and used at the next timestep.

   A special case arises when there exists reliable and complete information (e.g. reliable parameters of the Gaussian or mixture of Gaussians distribution) regarding the model distribution that is to be used by the surprise mechanism. Here the update of the model to account for "new" data may be skipped, leaving the model distribution fixed throughout the process.

The process of measuring surprise is illustrated using the scenario where a robot collides against a wall. While the robot moves in its environment it builds up a confidence in its sensor readings. This confidence (modeled as a Gaussian probability distribution), represent a model that predicts the behavior of the input sensor data.

As the robot moves, it builds up the distribution over the distance it covers between sensor readings. The more time that the robot moves without colliding with the wall, the more it increases its confidence regarding its velocity. In probabilistic terms, this means that the standard deviation of the model probability distribution is small, provided that the noise level in the sensor data is either small or filtered out. When the robot collides against the wall, there is a drastic change in the travelled distance between sensor readings. This variation is soon reflected in a change in the estimated probability of the observed data. The comparison between the model and observed probability distributions will show a large divergence. This is then triggered as a surprise. Finally, the model probability distribution is updated to adapt to and account for the new values observed, hopefully providing a more accurate model of the sensor data.

4.2 Short-term Memory

An important influence on the measure of surprise is imposed by short-term memory. Some approaches to both cognitive and computational surprise recognize the crucial effects of "remembering" past states, values or behaviors when reacting to surprises. Examples of different implementations are the "forgetting factor" in [IB06], the belief change proportion [LC07], or the "persistence" factor in [Pet98].

In the context of the computational model of surprise, *short-term memory* is a storage system of limited capacity, fast access and temporal nature, that contains recent information on sensor readings, used to estimate the probability distribution of observed data. It uses the reasoning that events that have occurred recently (i.e. are "fresh" in memory) are more likely to happen in the near/immediate future, unless there is evidence or knowledge that states the opposite.

As an example, consider the scenario of the robot pushing an object (a box). The first time that the robot stumbles into the box and starts pushing it, there will be a trigger of surprise given by the contact of the robot and the box, and the box starting to move. If the robot continues to push the object, however, the continued contact and object motion will start to look less surprising with each sensor reading received, until it becomes "normal". Furthermore, if the robot stops pushing the box and goes back to wander around, the surprise caused by the contact and object motion, is gradually "forgotten". If the robot stumbles again against the box after some time moving around, this event will be surprising again.

Short-term memory is implemented by using a buffer of fixed size (i.e. the memory size) to calculate the observed probability distribution. The behavior of the buffer is similar to a *first-in, first-out queue* (FIFO) where the oldest values are discarded as new values are received (see Figure 4.2). The effect of this implementation, as noted by the example presented previously, is two-fold:

- It allows for an habituation to changes in the input data. This is specially important when considering the case of a surprise derived from a change of state of the robot. The robot should adapt to this change of state and gradually get “used to” the surprise, until it is not surprising anymore.
- Conversely, it allows to “forget” an event that has been experienced, enabling it to become surprising again. This enables the robot to experience surprise towards known events that have not occurred for some time.

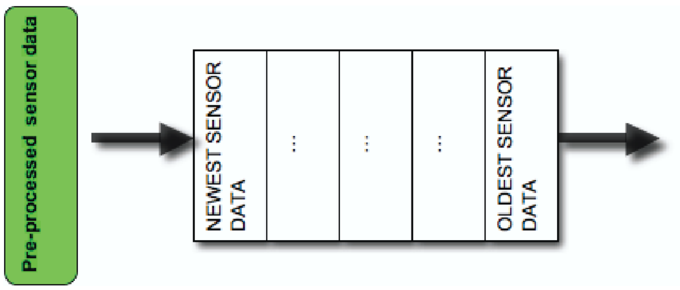


Figure 4.2: Short-term memory as a FIFO queue. Sensor data that grows “old” is discarded as new sensor data is received.

The exception to this reasoning occurs when there exists knowledge which claims that the observed event is “normal” and that the robot has just “forgotten” about it. The storage for such knowledge is what we call *long term memory*, which is discussed in the following sections.

4.3 Long-term Memory

A previous section of this document presented the influence of short-term memory on the intensity of surprise. A second memory component, namely the *long term memory*, a non-volatile storage of unlimited capacity, that contains knowledge acquired by the robot or given *a priori* by a human designer, can also have a great influence on the final outcome of the surprise mechanism. Long-term memory is also commonly known as *knowledge base*. Through the remainder of this document, the terms *knowledge base* and *long-term memory* are used as synonyms.

One of the most desirable behaviors of a surprise mechanism in a robotic learner is to be able to use the knowledge acquired through learning, in order to explain a surprising event or situation. The idea behind this statement can be explained by the scenario where the robot stumbles with a box, pushing it along. As it was seen previously, this scenario leads to a surprise generated by the contact between robot and object, and the consequent object motion.

However, if there is knowledge available in its *long-term memory* (possibly from past experiences) that is able to explain this surprising event, there is no need for the robot to initiate a new learning process: the robot already knows what a collision with (and posterior motion of) an object is, therefore it does not need to learn anything new from it. In this case, it is said that the explanation provided by that knowledge *subsumes* the initial surprise caused by the robot sensors.

The model of robotic surprise proposed in this thesis, utilizes knowledge that has been previously acquired (i.e. learned) by the autonomous robot, to determine when a surprise needs to be suppressed by the existence of an explanation for the observed phenomena. To achieve such behavior, a subsumption principle is applied to suppress the surprise output when a model of the observed phenomena exists. Subsumption has been successfully applied to robotics in the past decades, most notably as an approach to robot control. More information can be found in [Bro86].

Currently two kinds of models have been considered: qualitative and first order logic (FOL) models. Both kinds of models feature characteristics that make it possible to either explain the current "surprising" state of the world, or make predictions about the outcome of a physical phenomena caused by the robot actions.

4.3.1 First Order Logic Models

First order logic models are formed using variable, function, relation and constant symbols. Such models are capable of expressing concepts, states of the world, etc. by forming expressions that feature one or several components, e.g. $\exists apple\ (\mathrm{Fruit}(apple) \wedge \mathrm{Red}(apple))$ is a valid expression in first order logic.

Well known Machine Learning (ML) approaches and systems, such as Inductive Logic Programming (ILP) and Prolog, use this or a similar/extended type of knowledge representation because of its flexibility and expressive power to represent/learn abstract concepts [MR94, MLAS06]. The idea behind such representation is to enable a robotic learner that uses these ML methods, to employ the acquired knowledge to decide when a "surprising" event is really surprising.

Through the implementation of the concepts presented in subsequent sections, the naming conventions and notation introduced in [Gol05] and [RN03], will be used. Though the detailed revision of *first-order logic* paradigms is out of the scope of this thesis, some basic concepts will be presented for the better understanding of the experimental results.

$$\forall\, robot, box \quad : \mathrm{moving}(robot) \wedge \mathrm{moving}(box) \wedge \mathrm{contact}(robot, box) \Longrightarrow \mathrm{movable}(box) \quad (4.6)$$

Consider first, Equation 4.6. Here, the expression movable(box) is a construction composed of several other expressions and *logical connectives*. This is called a **formula**. Notice how each of the terms composing a formula can also be a formula by itself. The formulas moving($robot$) , moving(box) , contact($robot, box$) express a given fact, that is, a term that is **true** in a given model, and under a given interpretation. They are called **atomic formulas**. The quantifier $\forall\ robot, box$ indicates the generality of the knowledge represented by the formula: it holds *for all* the robots and *for all* the boxes known by the robot.

Though *first-order logic* provides a much richer language to represent knowledge, the computational model of surprise utilizes only a subset of it, formed by complex and atomic formulas similar to the ones shown in Equation 4.6, this is, using only universal quantifiers, negation and conjunctions to represent a formula. The possibility of transforming expressions involving other operators and quantifiers, suggests the feasibility of such representation. For a more complete insight of the possibilities offered by *first-order languages* in general, and specifically of *first-order logic*, consult [Gol05, RN03].

To illustrate how the proposed model of robotic surprise makes use of this kind of knowledge representation, consider the case of a robot equipped with a bumper and odometry sensors, as well as a sensor that allows to distinguish when other objects have moved. Such robot moves in an environment where the only object present is a box. While the robot wanders around the environment, eventually it bumps into the box, experiencing a surprise (given by the bumper suddenly activating, and/or the box moving).

Assume now, that there exists a formula in the robot's memory as shown in Equation 4.6 which should be read as "the box is movable if there is a contact between the robot and the box, and the robot is moving and the box is also moving". What must be noted is that this expression is able to explain the observation that caused the surprise: the bumper sensor went on, denoting a contact between the robot and the box, and the box moved as well as the robot. The surprise generated by the sensor information alone is then suppressed by the existing knowledge, which indicates a "normal" experience resulting from a collision between the robot and an object.

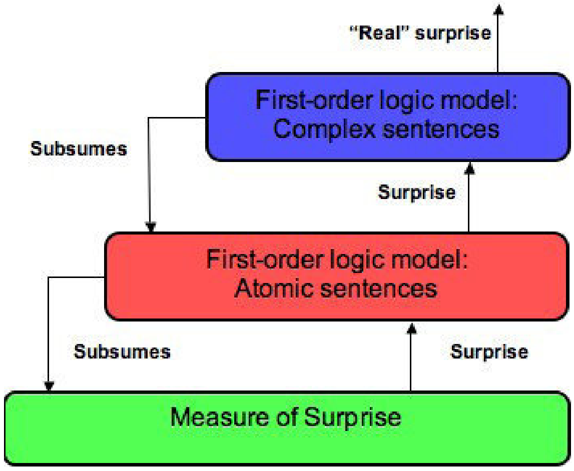


Figure 4.3: Surprise subsumption schema that depicts the use of first-order logic models to explain surprises experienced by the robot.

Figure 4.3 presents a schema of the subsumption process undergone by the measure of surprise, in presence of knowledge available in the *long-term memory*, in the form of FOL models. When the output of the measure of surprise indicates the occurrence of an unexpected event (contact with another object), the *long-term memory* is inquired regarding available knowledge that can explain the observation. The robot first looks among the *atomic formulas*, if any formula agrees with the observation that caused the surprise, the response of the measure of surprise is subsumed. Otherwise, the process is repeated with the *formulas*. If no explanation was found, the output of the surprise mechanism signals that surprise has been experienced.

4.3.2 Qualitative Models

Qualitative models are able to capture the distinctive characteristics of the systems they describe while ignoring their quantitative details. Furthermore, they have proved to be extremely efficient in representing incomplete knowledge. Qualitative modeling consists of creating a model of a system by means of qualitative descriptions. These descriptions are capable of representing incomplete knowledge of the structure and behavior of the system. Such behaviors can be predicted from the model which consists of parameters and their domains, various constraints on the parameters and an initial state definition of the system. More information on qualitative modeling as well as its application to robotic surprise can be found in [MJH07, Kui94].

The use of qualitative models to explain observations follow the ideas presented by Mohan et al. [MJH07] where a qualitative model is created prior to the robot executing an action.

Such model predicts the effects of the robot actions in the environment, where surprise is triggered if the observed effects differ from the prediction. To enable the comparison between the qualitative model and the observed phenomena, the variables that are present in the model and that are available from sensor data are analyzed. This analysis is performed by using a temporal abstraction and trend extraction mechanisms that obtain the qualitative information required for the comparison.

Let's take now the scenario of the robot pushing a box, and extend it so that the robot can obtain visual information from its environment trough a camera mounted on top of it. The extra information that this new sensor offers is the object perceived size and relative speed. As the robot starts moving towards the object, with a constant velocity, the bumper sensor is off, the object perceived size is increasing and its relative speed is not changing (state S1). At the moment of contact, the robot speed decreases, its bumper sensor is on, the object size remains constant and its speed starts to increase (state S2). What follows is the evolution of the physical phenomena: the robot speed starts to increase until it reaches again a constant speed, while the object speed first increases, and then decreases until the object stops (states S3-S5). Figure 4.4 shows a representation of a qualitative model associated with the phenomena described.

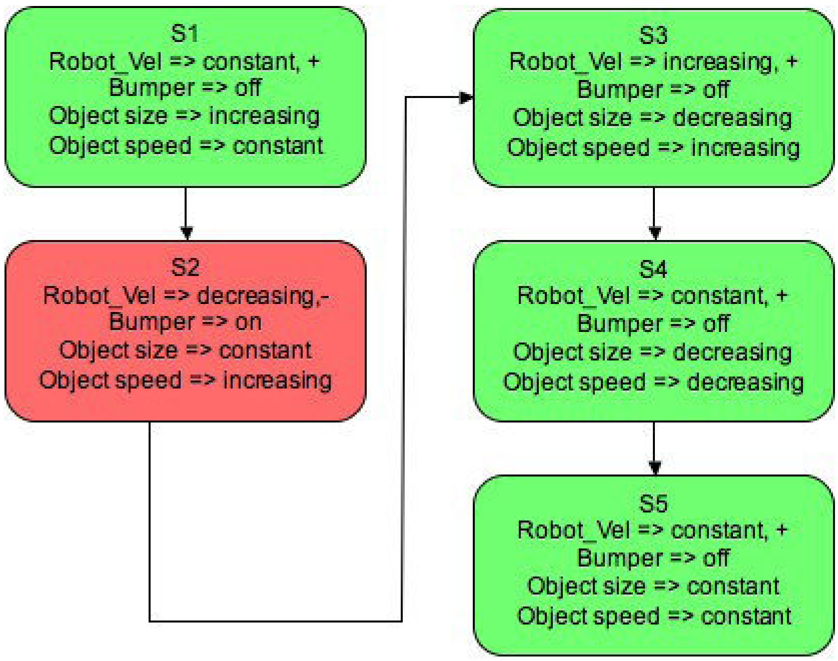


Figure 4.4: Qualitative model for a robot that bumps into an object, causing it to move. The surprise generated by the contact of the robot and the object (state S2, in red), can be explained by the subsequent states (S3-S5).

The bumping and motion of the object will generate a surprise observable in the sensor readings (e.g. a sudden activation of the bumper, a sudden decrease of the observed object size). However, as there exists a qualitative model that can explain the observation (namely the collision and posterior movement of the object), surprise is subsumed by the existing knowledge.

The use of qualitative models adds a robust representation of physical phenomena to the knowledge that may be available to explain a surprising event. However, it also introduces issues that must be taken into account, such as the decision of when to start and stop “observing” an event, how to create the qualitative model(s) that are stored in the long-term memory, and how to select among multiple, equally possible models that could explain a situation. The approaches taken towards solving these issues are discussed in Chapter 5.

4.4 Embodiment

Enabling the robot to react to surprises inevitably involves an influence from the embodiment and the system designer. While the former determines the kind of surprises the robot can react to, the latter provides the robot with the mechanisms to process the data coming from the sensors, so that the surprise can actually be measured.

These mechanisms must consider the use of several sensors in the detection of surprise, as well as the fact that the data received from one sensor can be processed in different ways, so that it provides information that enable the measurement of different “surprises”. For example, the odometry sensor can provide information on the speed, acceleration and pose of the robot simultaneously. Therefore, it can present one, two, three or no surprises at the same time, depending on the characteristics of the sensor, the way that the pose data of the robot is processed, the situation that the robot faces in the real world, and the knowledge that is available to the robot regarding such situation.

The model of surprise proposed in this thesis, accounts for the characteristics of the robot embodiment, associating the sensors available with different ways of processing sensor readings, ensuring that the desired surprises can be measured. To enable a robotic learner to react to a surprise from contact with another object, the robot must be told how to process information from the sensors in a way that symbolizes contact. If the robot has a bumper sensor, this is straightforward: contact will be directly related to the sensor turning on or off.

If the embodiment is modified, eliminating the bumper sensor, and adding proximity sensors and a range finder that provide the distance to close objects, contact could be defined as a being inside a minimum distance towards the object. Given that the information that both new sensors can provide is similar, there is no need to implement the way of obtaining contact information separately.

Figure 4.5 shows a representation of the previous example. Three sensors, a bumper, a laser range finder and IR proximity sensors provide different kind of data. While the bumper provides a boolean value (true/false), the range finder and IR sensor provide information on the distance to other objects. An implementation of the data processing function (CONTACT1) takes as input the distance to the object, and produces an output in a format that can be understood by the surprise measure while analyzing those values. The second implementation (CONTACT2) takes a boolean value as input (bumper), and produces the output in the same common format. Note how the implementation of CONTACT1 can be shared by the laser range finder and the IR sensor, only needing the association to the appropriate function. Similarly, if there exists another sensor that detects contact producing a boolean variable, the implementation CONTACT2 can be reused to transform/translate that data, to something that the measure of surprise can use.

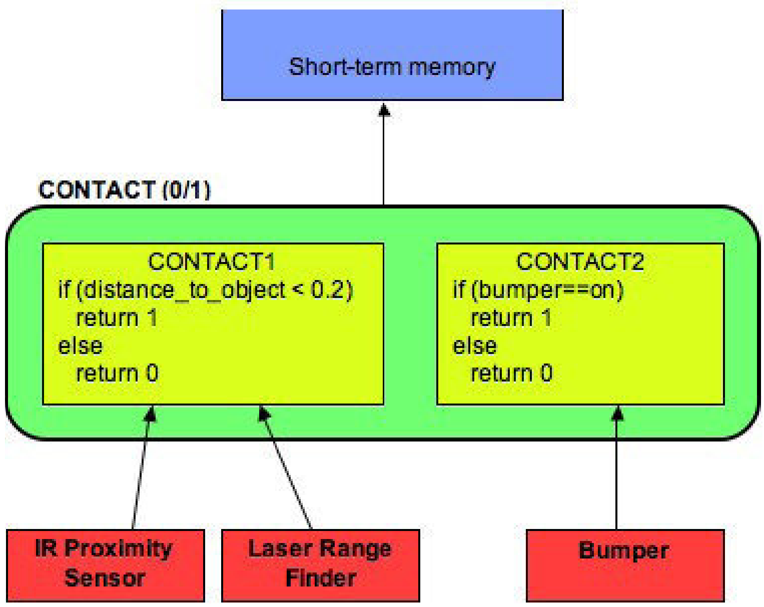


Figure 4.5: Mapping of different sensors to data processing mechanisms that allow to obtain a specific kind of surprise.

It is important to note that, while this does not eliminate the influence of the designer in defining the ways the sensor data must be processed (the robot still needs to know how to calculate velocities, accelerations, object edges, etc. if they are not provided directly from sensors), it allows for mechanisms to be reused while processing data from different sensors, offering a greater flexibility when reusing/changing the way data is processed.

While the association mechanism described might appear as a purely software engineering task, it certainly allows for an easier and efficient data pre-processing of multiple sensor input, which is of crucial importance to the measure of surprise.

Chapter 5

EXPERIMENTAL RESULTS

This chapter presents the results obtained from the application of the computational model of robotic surprise to a robotic learner, specifically, one that learns by experimentation. Here the robotic agent executes its pre-defined actions until a surprising event occurs. When the observed event can not be explained by available knowledge, a learning process is triggered, aiming to acquire insights about such event through interaction with its environment, under the assumption that such knowledge might serve to explain future unexpected situations.

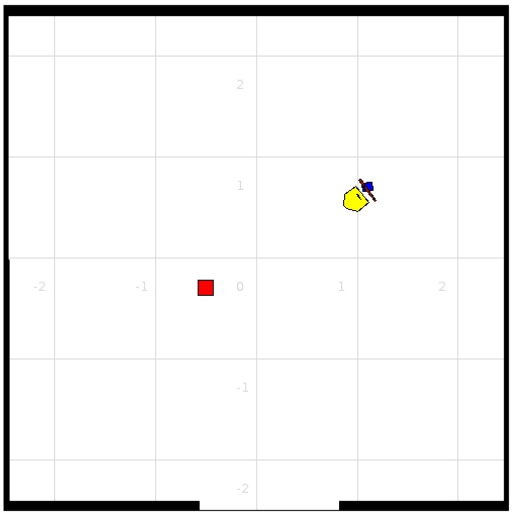
5.1 Experimental Setup

The running example scenarios described in Section 2.2 have been used to perform experiments that test the computational model of surprise. Two environments were used, one in simulation and one in the real world.

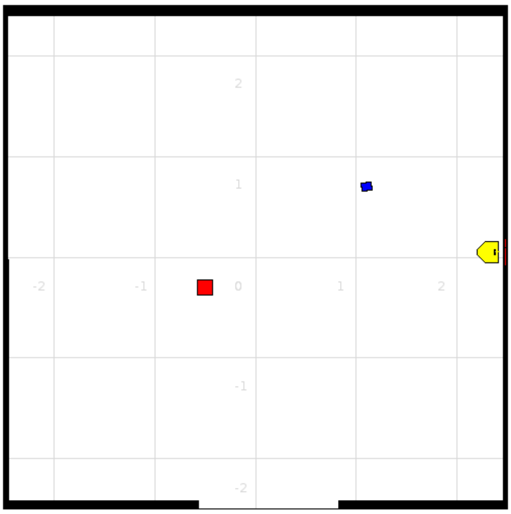
The simulated environment was built in Player/Stage simulation tool. Such simulation consists of a squared area enclosed by walls containing a robot, and two squared boxes that the robot can move. The size of this arena is 4X4 meters. The simulator directly provides information on object id's, object global pose, robot global pose, robot velocity, robot odometry, and robot bumper sensor.

Figure 5.1 depicts the two experimental scenarios, modeled in the robotic simulation tool. Figure 5.1 (a) shows the moment of collision of the robot with a blue box, while Figure 5.1 (b) shows the moment of collision with a wall. These situations cause the robot to experience different surprises, as will be seen in following sections of this chapter.

The real world environment consists of a rectangular area enclosed by walls, containing a robot and two carton boxes, movable by the robot. The size of the arena is 1.20X0.80 meters. An overview camera is located on top of the arena, with the image plane parallel to ground, covering a subset of the total area available for the robot. The image data delivered by this camera is pre-processed by computer vision algorithms, such that it is able to provide information on the objects in the arena, in a similar way as the simulation does, for example: object id's, object global pose, distances between objects, velocity of moving objects, etc.



(a) The robot running into an object.



(b) The robot running into a wall.

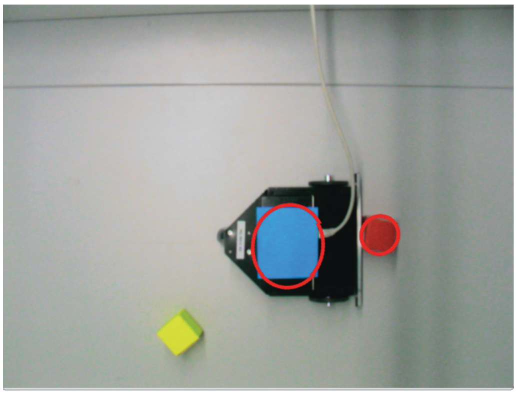
Figure 5.1: Two experimental scenarios to test the computational model of surprise. The robot faces two situations that can potentially generate surprise: (a) the robot stumbling into a movable object (and actually moving it) and (b) the robot running into a wall.

Figure 5.2 shows a snapshot of the real-world execution of the mentioned experiments. In (a), the robot has collided with one of the boxes in the environment, and is pushing it while moving. In (b) the robot has collided with one of the walls of the arena. The red ellipses on top of the robot and the box, represent the color tag that is being tracked, providing some of the desired sensor information (e.g. robot and object position). The bumper and odometry information is simultaneously obtained from the robot.

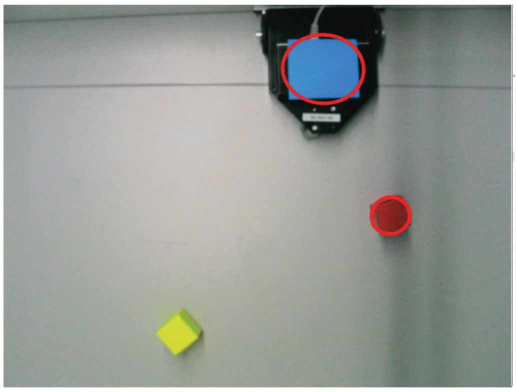
The robot embodiment used in the experiments (both in simulation and real-world environments) is an Eddy educational robot, supporting odometry, and bumper sensors. In the case of the real robot, the information gathered is not processed on-board, but sent to a host computer via TCP/IP communication. The robot is able to run autonomously or by tele-operation using a bluetooth controller. More information on the robot can be found at [RG].

5.2 Results: Measure of Surprise

The implementation of the measure of surprise for the experiments, assumes that the sensor input follows a Gaussian probability distribution. However, there is no prior information on this probabilities, which means that the distributions have to be obtained



(a) The real robot runs into an object.



(b) The real robot runs into a wall.

Figure 5.2: Images of the real robot used in the experiments. The two situations that the robot faces (as shown in Figure 5.1), are given by: (a) the robot stumbling into a movable object and (b) the robot running into a wall.

(estimated) as the robot wanders around.

Once the data coming from the sensors has been pre-processed, the measure of surprise receives four sensor data variables: displacement data from the red and blue boxes, and displacement and bumper data from the robot. The surprises that arise in the experimental scenarios are:

1. **The robot wanders around and bumps into a wall**. The surprise expected to arise should come from the sudden change in the displacement of the robot (as it is stopped by the wall), and the sudden activation of the bumper sensor (as it makes continuous contact with the wall).

2. **The robot wanders around and bumps into any object**. In this case, the expected surprise will also include the displacement of the object the robot collides with.

Though the surprise generated by both events is very similar, it will be later seen that the influence of knowledge available in the *long-term memory* plays a fundamental role in determining the final output of the surprise module. The following subsections present the results of applying the surprise mechanism to the two situations described.

5.2.1 The Robot Bumps into an Object

The robot faces a situation where it bumps into an object (a box) without stopping its own motion. The effect of its action is that the object moves along with the robot, pushed by it. This should generate simultaneous surprises from the sensor data that is being processed: the bumper data and the object displacement.

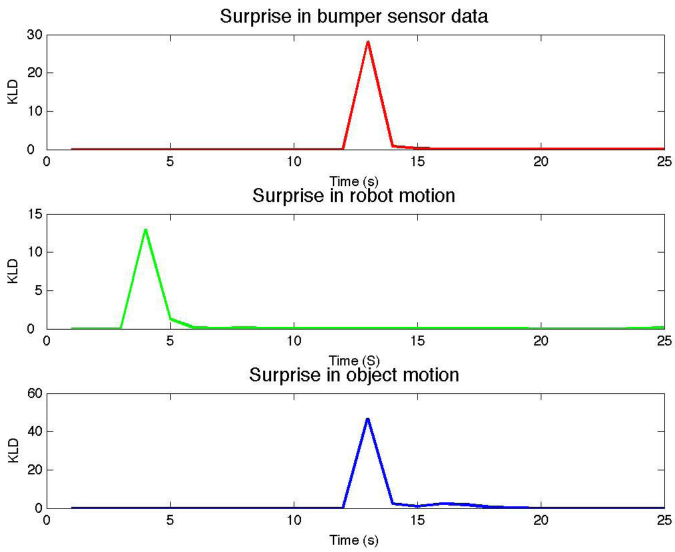
Figure 5.3 presents the results obtained through the experiment, for both simulation and real world environments. To better appreciate the behavior of the measure of surprise, three sensor information are shown simultaneously: bumper sensor data (top), robot displacement data (middle), object displacement data (bottom). No additional information regarding the phenomena was available in the *long-term memory.*

The first interesting result to observe is the behavior of the robot displacement for both figures. This variable shows an initial surprise that occurs when the robot starts to move ($t = 4$s for simulation, and $t = 19$s for the real robot). This is the result of the first sensor reading processed by the surprise mechanism. The robot passes from a state where it does not have any information, and thus no expectation, to a state where it already has an initial (very crude) estimate of the expected sensor values (an initial model distribution).

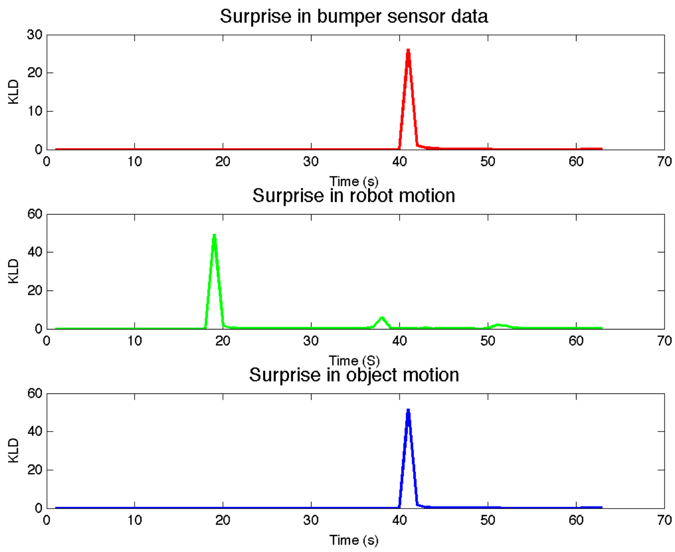
To enable such "first time" comparison, a standard normal distribution was provided to be compared with the first estimate, producing the mentioned value depicted in Figure 5.3. As more/new data is received, the robot improves its estimate of the probability distribution that describes the variable behavior, and the variation of the observed probability distribution with respect to the constructed model, starts to decrease until it reaches nearly zero, this is, a point where the model is good enough so that the observation is explained successfully.

It should be noted that the initial behavior of the variable corresponding to robot displacement, is not observed in the other two variables (bumper and object displacement data), since the initial displacement of the object, as well as the initial reading of bumper data have a value of 0, i.e. the object is not moving and the bumper sensor is not in contact with anything. This coincides with the parameter values of the standard normal deviation ($\mu = 0, \sigma^2 = 1$) used as the initial distribution that enables comparison, and thus, the KLD between the observed and model probability distributions is also zero.

The second result that must be pointed out is the simultaneous trigger of surprise in the bumper and object displacement data (see Figure 5.3 top and bottom respectively in both graphs). This corresponds with the sudden activation of the bumper due to contact with the object, and the object displacement as it is being pushed by the robot. Remarkably, the analysis of the robot displacement variable does not reveal any simultaneous surprise. This is explained by the fact that the robot has been moving before the con-

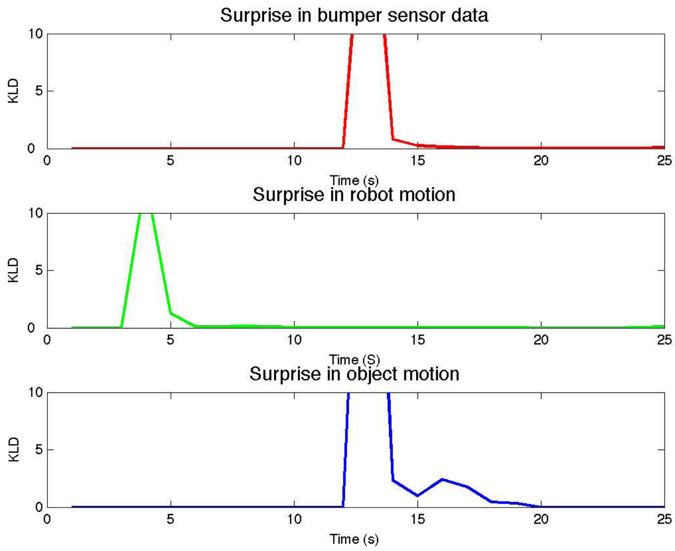


(a) Surprise while running into a movable object using simulation.

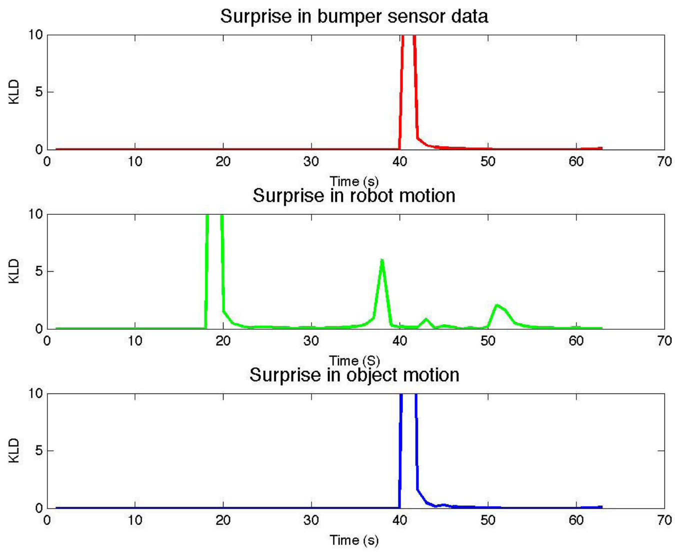


(b) Surprise while running into a movable object using a real robot.

Figure 5.3: Measure of surprise obtained in the experiment using (a) simulation and (b) a real robot. In both modalities the robot wanders around and suddenly it bumps into an object (a box that can be moved). The variables analyzed were: bumper sensor data (top), robot displacement data (middle), and object displacement (bottom), respectively.



(a) Surprise while running into a movable object using simulation.



(b) Surprise while running into a movable object using a real robot.

Figure 5.4: Detailed view of the behavior of the measure of surprise in the experiment involving the contact of the robot and a movable object, using (a) simulation and (b) a real robot.

tact, and continues to do so afterwards, thus, there is no surprise elicited by that behavior.

An important feature of the measure of surprise, is its ability to adapt to both new input values, and constant surprises. This is equivalent as to say that, if a surprise repeats constantly, it gradually stops being surprising. From the graphs in Figure 5.3, this feature is not observable given the magnitude of the KLD observed. Figure 5.4 presents a detailed view on the behavior of the measure of surprise. The size of the *short-term memory* plays an important role in determining how fast the KLD value decays over time, in presence of constant surprise. This is analyzed in later sections of the document.

Finally, Figure 5.4 (b) also reveals an interesting behavior of the variable *robot displacement* in the real-world environment: at $t = 38$s and $t = 51$s, there are two other peaks of surprise, of a much lesser magnitude than the previous one, which do not appear in simulation data. The explanation to such "surprise" peaks is given by the noisy nature of the data coming from the real robot. Unlike simulation, the real robot provides sensor data that is affected by several conditions, e.g. wheel slippage, inaccuracy in the robot tracking mechanism while using the overview camera, irregular robot motion between sensor measurements, etc. These factors condition the appearance of such surprise peaks, despite the application of basic data filtering techniques, such as signal smoothing and thresholding, to the raw sensor data.

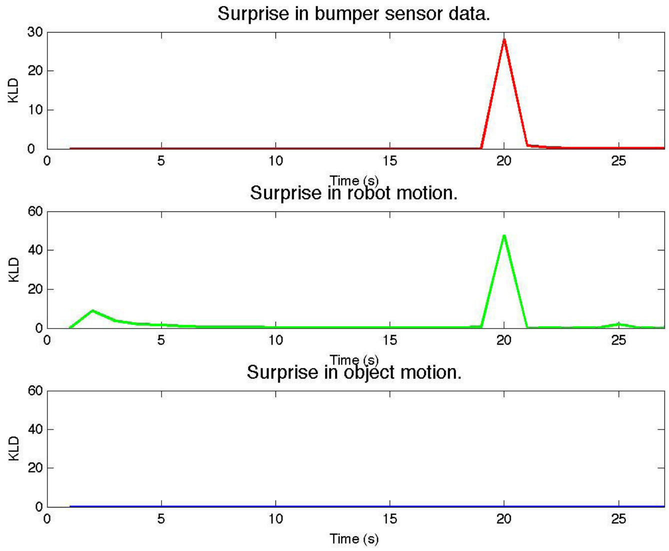
It is important to mention, however, that the appearance of such peaks does not invalidate the results produced by the measure of surprise: the response to a real surprising event is by far, more significant than the response to noise in the measurements. Thus, a simple thresholding technique can be used to suppress such undesired surprises, resulting in a reliable measure of response to unexpected events.

5.2.2 The Robot Bumps into a Wall

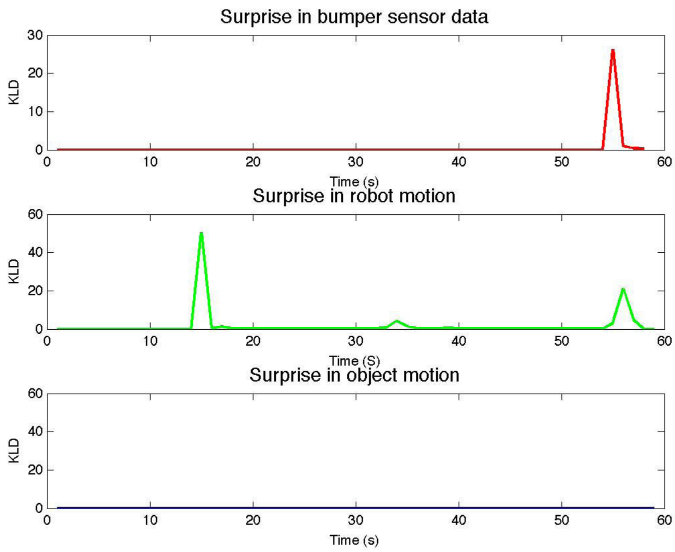
This scenario presents the robot with a new situation: the robot faces a surprise elicited by the contact with the wall, and a surprise generated by the impossibility of moving further.

The system settings are the same as in the previous scenario, namely, three variables are analyzed (object and robot displacement and robot bumper data), and they are assumed to follow a Gaussian probability distribution. No additional knowledge is available to the robot in the *long-term memory* (therefore, no surprise subsumption).

Figure 5.5 shows the results obtained when measuring the surprise of such event. As the



(a) Surprise while colliding with a wall using simulation.



(b) Surprise while colliding with a wall using a real robot.

Figure 5.5: Measure of surprise obtained in the experiment using (a) simulation and (b) a real robot. In both cases the robot moves straight until it collides with a wall. As in the previous experiment, the variables analyzed were: bumper sensor data (top), robot displacement data (middle), and object displacement (bottom), respectively.

robot starts to move, the behavior of the observed variables is similar to the one described in Section 5.2.1. The remarkable events occur almost at the end of the observation (time $t - 20$s in simulation, and time $t = 55$s in real-world).

At this point, the robot collides with the wall, experiencing the surprise of contact (represented by the bumper suddenly turning on), and of the impossibility to move further (represented by an abrupt change in the displacement data). Additionally, the object displacement shows no surprise at all, given that it is not moved by the robot during the course of the experiment.

Again it is interesting to pay attention to the behavior of the variable *robot displacement* (Figure 5.5(b)). The magnitude of the surprise experienced by the impossibility of further movement, is lower than the one experienced when the robot started moving for the first time. This behavior is due to the characteristics of the data received from the sensors, and the noise associated to it: the underlying probability distribution at this point in time, is more "permissive" than the one at the start of the observation (as many different, noisy values have been already observed and used to estimate the future behavior of the variable). However, the magnitude of the surprise is significative enough to not be considered "noise" surprise.

5.3 Results: Short-term Memory

To illustrate the effect of short-term memory when experiencing a surprising situation, let's consider what happens before, during and after the surprise, that it supposes for the robot to collide with one of the boxes. Figure 5.6 shows the behavior of the estimated probability distribution for the two variables that elicit surprise in the experiment, namely, *contact* of the robot and the object, and *object motion*.

The instant before the collision, the expectation (estimated probability distribution) created around the sensor input values (curves in color *brown* in both graphs) shows a high expectation about the bumper sensor being off ($\mu = 0.01, \sigma^2 = 0.07$) and the box not moving ($\mu = 0.005, \sigma^2 = 0.003$). The surprising event (the collision) causes an immediate change in the estimated probability distribution (curves in *red* in both graphs).

The result is also shown in Figure 5.6: the estimated probability distribution is updated with each new information received (curves in *blue*), towards a new probability distribution (curves in *green*). Consequently, the memory is gradually "forgetting" about the original probability distribution.

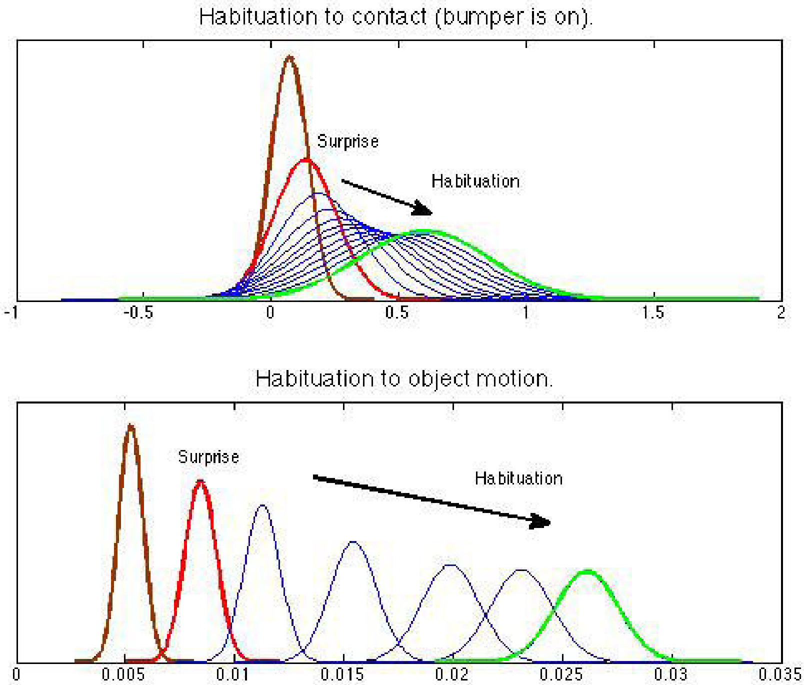


Figure 5.6: Habituation to the surprise generated by the continued contact with an object, after wandering around in the environment without bumping.

Table 5.1: List of number of sensor measurement required for the surprise mechanism to adapt to a new situation.

Memory size	Simulation		Real robot	
	Contact	Object motion	Contact	Object motion
5	5	11	5	15
10	9	10	9	19
20	9	9	10	8
30	7	9	10	9

Table 5.1 presents results on the time it takes the surprise mechanism to adapt to new/noisy data received from the robot sensors, for both simulated and real robot. The results displayed, show the number of sensor measurements needed (in average) to adapt to new data, for the variables contact (bump sensor) and object motion (object displacement), using different short-term memory sizes. The memory size (number of sensor measurements that the robot could "remember") was varied to store 5, 10, 20 and 30 measurements.

It can be observed how for memory sizes 20 and 30, the estimated probability distribution takes averages of 9 ± 1 sensor readings to habituate (converge to a stable value), for both simulation and real robot, and regardless of the variables analyzed. The results from memory sizes 5 and 10, however, differ significantly: while the *contact* variable needs to empty its associated memory, replacing old measurements by new ones, to habituate to a new event, the *object motion* variable needs almost double its memory size to habituate.

An explanation for this behavior lies in the different nature of the *contact* and *object motion* variables: while the former measurement only presents two values (on/off) with very low error, the latter shows a much greater variety (as well as associated noise) of possible values. A short memory size, does not provide enough measurements to properly estimate the a probability distribution that accounts for such different values. Thus, it remains unstable for a longer period than when using a larger memory size.

5.4 Results: Long-term Memory

The experiments performed have considered so far, the application of the measure of surprise without the influence of knowledge stored in the *long-term memory*. This section presents the implementation of a concept presented previously: existing knowledge that is able to explain a surprising situation, and is available to the robot, subsumes the surprise output.

5.4.1 First Order Logic Models

The first knowledge representation used to experiment with the influence of *long-term memory* on surprise, is *first-order logic* models. As outlined in Section 4.3, *first-order logic* is a declarative language that has expressive power that allows it to establish relationships between pieces of knowledge, deal with partial information (by using disjunction and negation), and provide compositionality (the meaning of a formula is a function of the meaning of its parts). All this characteristics are highly desirable in a system that has to perform a form of reasoning, which in our case, is a robot that tries to explain a surprising event.

The experimental scenarios used previously (robot bumping into a box, and robot bumping into a wall), were repeated, making the knowledge depicted in Equations 5.1 and 5.2 available to the robot. Equation 5.1 shows the atomic formulas that are valid for the robot in the context provided for the experiments performed.

$$\begin{aligned} \forall\, robot &: \text{moving}(robot) \\ \forall\, object &: \neg\text{moving}(object) \\ \forall\, robot &: \neg\text{contact}(robot, object) \end{aligned} \tag{5.1}$$

These atomic formulas should be read as follows: "for all robots, the robot is moving, for all the boxes present in the environment they are not moving, for all the robots present in the environment they are not in contact with another object". This knowledge express states that are "acceptable" for the robot, meaning, events that are explainable. Appendix B, Listing B.1 shows the XML representation used to store the atomic formulas in the knowledge base.

Consider the surprise experienced by the robot starting to move: instead of immediately triggering a surprise, the robot looks for knowledge stored in its *long-term memory* and concludes that the state where the robot is moving, is an "acceptable" or "normal" state. Therefore, surprise is subsumed.

The surprise generated by the robot bumping into an object or a wall, however, cannot be explained by this atomic formula, as it states that the acceptable value for *contact* is that the bumper sensor is off (false). It is here where more complex formulas play an important role in determining the amount of surprise in the event.

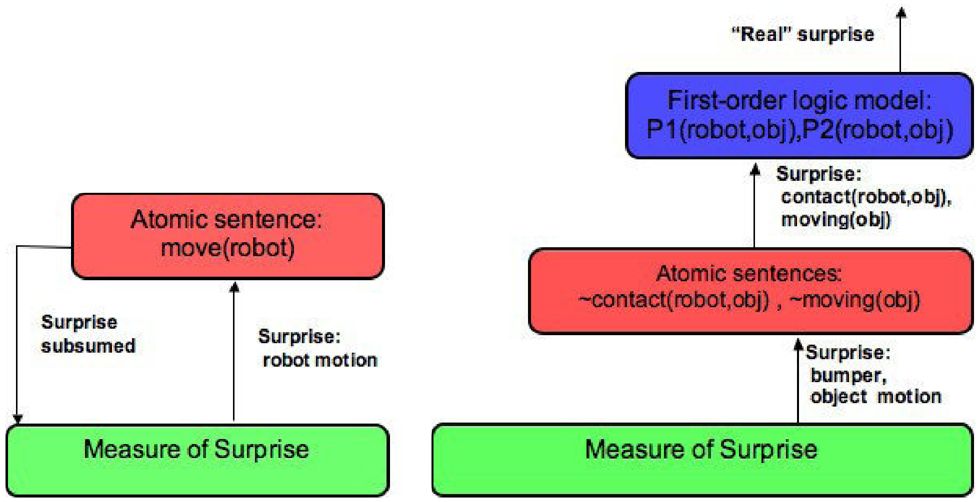
$$\begin{aligned} \forall\, robot, obj &: \text{moving}(robot) \wedge \neg\text{contact}(robot, obj) \wedge \neg\text{moving}(obj) \Longrightarrow \text{P1}(robot, obj) \\ \forall\, robot, obj &: \text{moving}(robot) \wedge \text{contact}(robot, obj) \wedge \neg\text{moving}(obj) \Longrightarrow \text{P2}(robot, obj) \end{aligned} \tag{5.2}$$

The formulas containing the knowledge available to the object are shown in Equations 5.2. Following with the *contact* example, lets focus in the second one. This formula states that *P2(robot,obj)* is possible if the state of the world indicates that: the robot is moving (or intending to move), there is contact, but the object does not move. What must be noticed is the underlying knowledge pointed out by such expression: the robot can be in contact with an object as long as it does not move.

This formula, thus, can explain the surprising observation: the robot has already learned that the bumper sensor can go on, thus, it is not a "real" surprise, and it is subsumed. The XML representation of these formulas, can be seen in Appendix B, Listing B.2.

Figure 5.7 represents the results obtained when applied the subsumption mechanism, to the experiment where the real robot bumps into an object. The first result (depicted in Figure 5.7(a)), was produced at time $t = 19$s, when the robot starts moving. As mentioned previously, the robot experiences a surprise by starting its motion.

When this happens, the robot looks for knowledge available regarding its own motion, and finds that among the atomic formulas, there exist one (moving($robot$)) that states that the surprises caused by the robot motion should be suppressed. The subsumption mechanism then activates, and no surprise is triggered in the output of the surprise mechanism.



(a) Surprise subsumption: the available knowledge explains the surprise.

(b) Surprise triggered: the knowledge available does not explain the surprise.

Figure 5.7: Surprise subsumption applied to the data obtained in the experiments (see Figure 5.3(b)). 5.7(a) shows a case where surprise is subsumed. 5.7(b) shows the case where surprise can not be explained by the available knowledge, triggering a "real" surprise.

Figure 5.7(b) shows a different situation that occurs at $t = 41$s, when both the object motion and bumper sensor fire surprise. The robot looks again for available knowledge that explains such experience. What it finds, instead, is that two of the atomic formulas available are contradicted: $\neg$contact$(robot, object)$ and $\neg$moving$(object)$ are *false*. In another attempt to find knowledge to explain the observation, the surprise mechanism looks in the complex formulas available. Here it finds that the formula P2$(robot, obj)$ is able to explain the contact between robot and object. However, none of the available formulas is able to explain the object motion. This results in a surprise present in the output of the surprise mechanism.

5.4.2 Qualitative Models

The second kind of knowledge used by the surprise mechanism to try to explain an observed surprising event, is the one represented by qualitative models. Section 4.3.2 presented the basic concepts behind the use of qualitative descriptions of the phenomena experienced by the robot. Such models have been used in the experiments to explain the effects of the robot actions in its environment.

To illustrate the application of such approach, consider again the first experimental scenario, where the robot is moving forward in its environment and stumbling into a movable object, then pushing it for some time. The prediction of the outcome of the event, was performed by qualitative simulation [MJH07], producing the qualitative model shown in Figure 5.8.

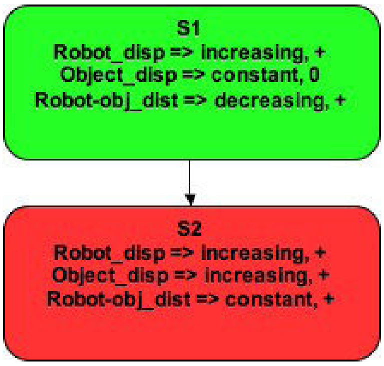
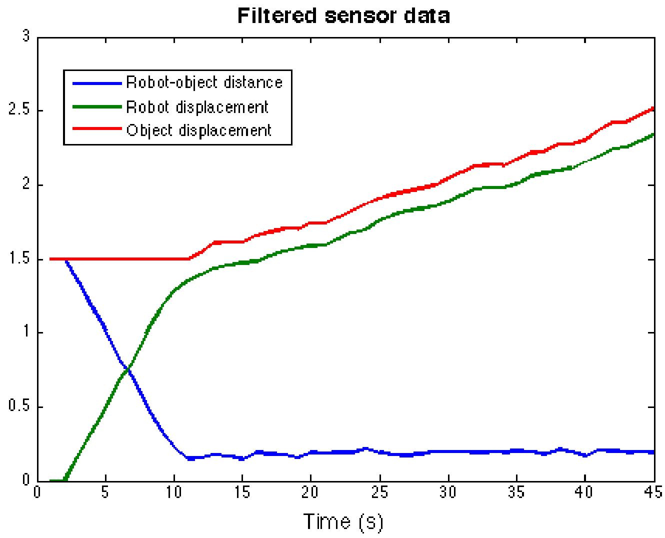


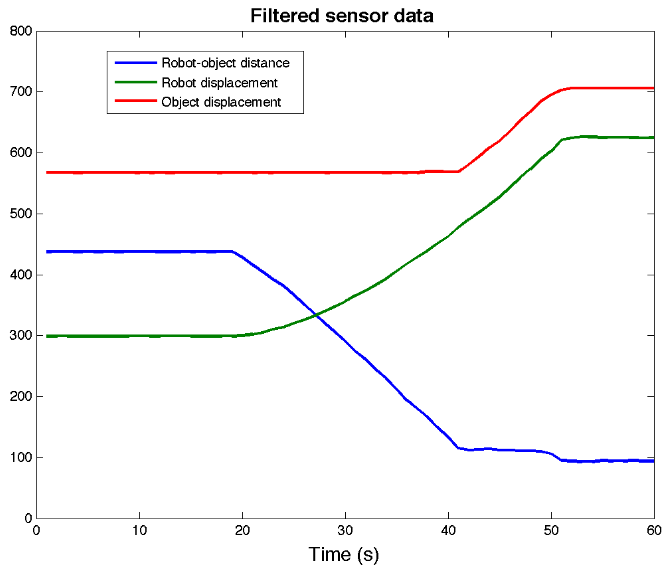
Figure 5.8: Representation of the qualitative model that predicts the outcome of the collision of the robot and a movable object.

The model bases the prediction in three variables: the *robot and object displacement* from their *original position*, and the *distance between robot and object*. Essentially, what this model tells is that, from the beginning of the action until the moment of the collision, the robot displacement *increases* (with respect to its original position), while the distance between itself and the object *decreases* (as it approaches the object), and the object displacement remains *constant* with a value of zero (as it is not moving).

From the moment of collision onwards, the robot displacement continues to *increase*, while the object displacement starts also *increasing*, and the distance between the robot and the object becomes *constant* (as they are moving together). The sensor data obtained as the robot executes this action, is shown in Figure 5.9, and the trends extracted from such



(a) Sensor data obtained from simulation.



(b) Sensor data obtained from the real robot.

Figure 5.9: Filtered sensor data corresponding to the experiment where the robot moves forward, collides with an object, and pushes it for some time, for: (a) the simulation and (b) the real robot.

data, are shown in Figure 5.10.

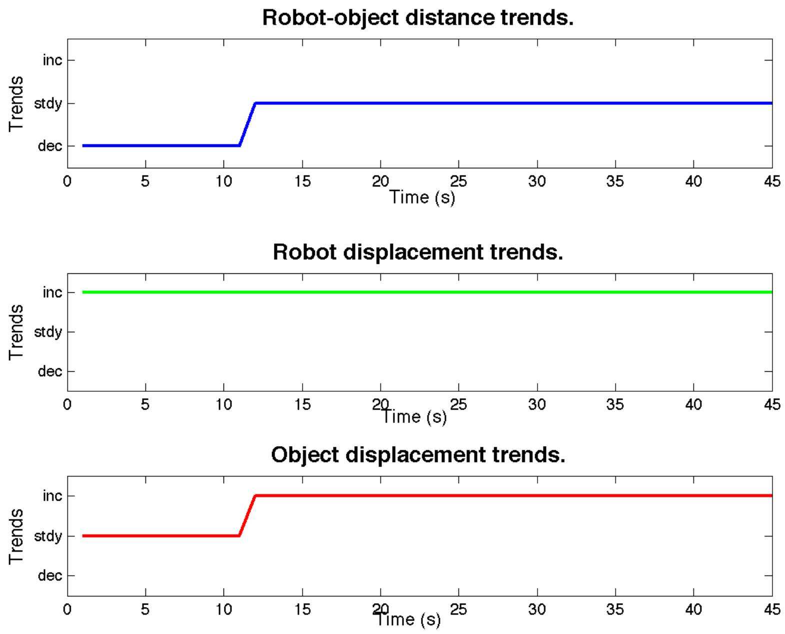


Figure 5.10: Trends extracted from the sensor data shown in Figure 5.9(a).

The data obtained in simulation and from the real world environments shows a great similarity in qualitative terms, therefore, it makes sense to analyze the trends extracted for any of them, with confidence that the results are fully applicable to the other. In this case, the results analyzed correspond to the simulated data. The behavior of the sensor data analyzed corresponds with the one stated in the model: the robot displacement with respect to its initial position starts increasing (time $t_1 = 0$ to $t_2 = 11$), as its distance with respect to the object decreases. From the moment of contact onwards ($t > 11$) the object starts moving as it is pushed by the robot. In this case, the surprising situation caused by the perceived contact of robot and object, and posterior motion of the latter (given by the bumper sensor turning on, and the object moving), is successfully explained by the model stored in the *long-term memory*. Therefore, surprise is *subsumed* by this knowledge and no trigger is produced in the output of the surprise module.

To analyze the case when the model cannot explain a surprising event, the experiment is repeated using the second scenario: the robot starts moving forward but, instead of colliding with the object, it passes it by, with a posterior collision with the wall (which produces the surprise). The qualitative model used is the same as in the previous experiment. Figure 5.11 shows the sensor data received during the experiment, where is possible to appreciate the behavior of the analyzed variables: the robot displacement increases from the start of the robot action until the collision with the wall, the object displacement remains constant with value zero (as it has not been moved by the robot), and the distance between robot and object decreases as the robot approaches to the object, and then increases as it passes it by in its way to the wall.

The trends extracted from the sensor data are shown in Figure 5.12, confirming the insights previously mentioned. When the robot collides with the wall, the robot experiences a surprise. The qualitative model stored in its *long-term memory* however, cannot explain the observations: the object displacement from the origin remains constant through the experiment, while the distance between the object and the robot first decreases, to start increasing once this has passed the object by.

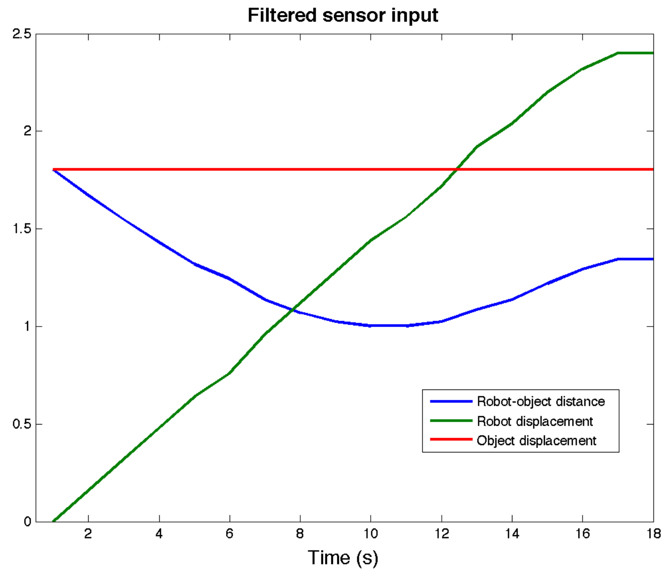
In this case, the surprise mechanism decides that the surprise perceived is a "real" one (the observed event cannot be explained with the available knowledge), and thus it triggers the design of experiment to start with the learning cycle, to acquire knowledge that explain the observation. It is important to notice that the surprise is not given by inaccuracy of the model, but the fact that the model was not designed to explain the new situation created in the experiment.

5.5 Discussion

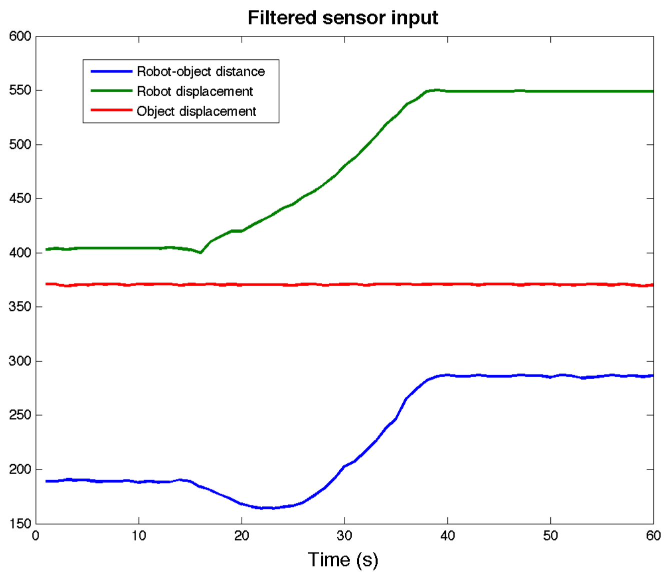
The results obtained from the application of the measure of surprise to two situations, that a robotic learner faces while executing "everyday tasks" (pushing a box and colliding into a wall), support the validity of the proposed approach to trigger artificial robotic surprise in a robotic agent.

The selection of a probabilistic approach to define a measure of surprise, implies the recognition of two important characteristics of surprise: that it is closely related to uncertainty, i.e. a degree of belief in "explanations" (models) to observed phenomena, and that it arises when such models are not able to satisfactorily explain the observations.

The measure of surprise used is the Kullback-Leibler divergence (KLD) of two probability distributions, also known as relative entropy. The two probability distributions compared, are the distribution prior to the observation (prior belief) and the estimation of the observed probability distribution. This means that the relative entropy measures how much the confidence in the model of the situation, has increased or reduced.



(a) Sensor data obtained from simulation.



(b) Sensor data obtained from the real robot.

Figure 5.11: Filtered sensor data corresponding to the experiment where the robot moves forward, passes by an object, and continues moving forward until it collides with a wall, for: (a) the simulation and (b) the real robot.

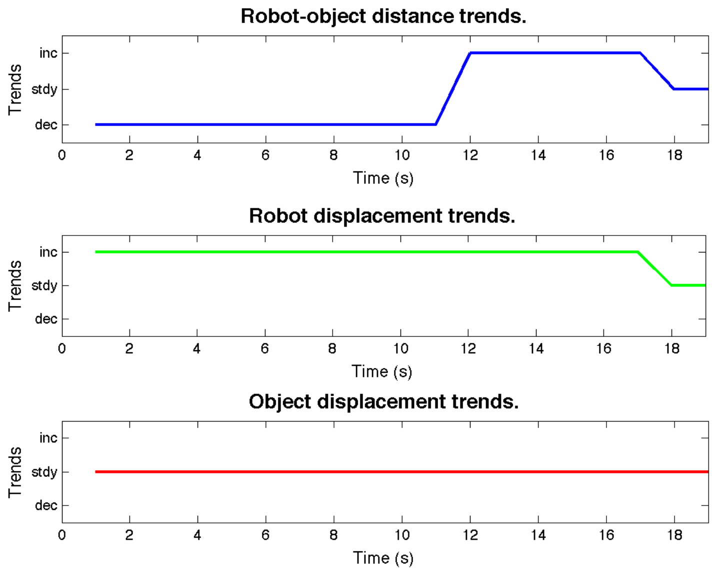


Figure 5.12: Trends extracted from the sensor data shown in Figure 5.11.

While this is hardly the only information-related measure for surprise, its selection originates in the fact that static metrics such as entropy, work directly over the data sets disregarding any notion of model. In the KLD measure the model plays a fundamental role: a probability distribution (estimated model) associated to the observations, that should be compared to a model distribution. The KLD measure has proven to outperform the best static metric (Shannon's entropy), for specific implementations appeared in [BI05, YW04].

The selection of the Gaussian distribution as model for the underlying distribution of the sensor data, offers several benefits. Gaussian distributions provide several well known analytic mechanisms to perform an online update of the distribution, as new observations are received from sensors, relieving from the need to calculate it numerically. Many known probability distributions, noise models and filters can be approximated by Gaussian (or mixture of Gaussians) distributions, for both discrete and continuous spaces [TBD05].

The drawback of this selection seems apparent: many (possibly unknown) probability distributions do not follow Gaussian characteristics and therefore, the bias imposed by assuming this distribution might result in an over generalization of the estimated distribution (e.g. it might converge to a uniform distribution, or never converge at all), becoming a poor match for the true underlying distribution and therefore, a poor model estimation prone to trigger numerous surprises.

Despite the shortcomings exposed, both the impossibility of obtaining a true distribution for all possible situations that the robotic learner might face (or even for a subset of them), and the desired autonomy in learning, strongly advocate for the assumption of a well-known probability distribution as a model for the sensor data received.

It must also be noticed that the measure of surprise is flexible enough to allow the substitution of the assumed probability for another. This allows the human researcher to incorporate an even greater bias to the surprise mechanism: if the human researcher knows that the situations faced by the robot are better expressed by a different distribution model, this can be easily substituted in the model, a priori resulting in improved performance.

The effect of short-term memory in the behavior of the surprise mechanism, is given by the gradual habituation to new sensor input/surprising events. This follows the notion exposed by [IB05, IB06] that a surprise mechanism must present a behavior such that a repeated surprise stops to be surprising, as well as an experienced surprise that "grows old", should be "forgotten" for it to become surprising again.

The implementation of such memory is based in a memory size that corresponds to how many sensor measurements (and the corresponding probability distribution estimations and surprise values), are stored, readily accessible to influence the measure of surprise.

Successful results in habituation to surprising events, support the proposed *short-term memory* mechanism: surprising events soon become un-surprising as the robot continues experiencing them (e.g. repeated contact between robot and object, repeated object motion). It is interesting however, to discuss the results presented in Table 5.1. While there was no exhaustive investigation on the cause of the high number of measurements needed to habituate, when using a small memory size (5 sensor readings), a priori considerations point to the behavior of the variable value, caused by the irregular motion of the object. The small memory size causes the estimate of the new probability distribution, associated to the observation and executed action, to be poor thus needing more time before it adapts to the new values observed. A larger memory size, on the other hand, favors a better estimation of the observed probability distribution, consequently providing a faster adaptation.

A more thorough evaluation of the causes and effects, of the variation of the memory size in the habituation of the measure of surprise to new/unexpected values, needs to be performed. Additionally, a study of the applicability of other short-term memory approaches such as [Rem07, Bak02], that might add robustness to the memory behavior, needs to be investigated.

The results obtained by using knowledge available in the *long-term memory*, to influence the final surprise output, support the notion that background knowledge improves the robustness of the approach, by adding some basic reasoning to the surprise mechanism. If the robot can explain a candidate surprise fired at sensor level, such surprise is not relevant to drive the robot towards the acquisition of new knowledge, and does not represent a significative event that the robot must attend to.

A fair criticism, however, arises from such statement: there might be situations where such basic reasoning (subsume surprise if you can explain it), is not desirable, e.g. a situation where the robot must react before imminent danger, signaled by a "surprising" event (for example, an alarm going on). The robot might need to react to such threat, even knowing the meaning of the surprising event perceived at low/sensor level.

A subjective appreciation of this case, however, indicates that it is not the surprise mechanism the one that should react towards such hazardous signal, but a different motivation mechanism that compels the robot to avoid danger. This insight can be extended towards other situations that are not specially dangerous as in the example, but require similar reactions from the robot. This new motivation mechanism should overcome the result of surprise to offer a proper response.

A second observation is directed towards the knowledge representations selected for the approach. While FOL and qualitative models have a great expressive power, able to deal with incomplete knowledge, it is also known that they have problems dealing with uncertainty, numerical and temporal conditions. The restrictions imposed to the FOL used (use only universal quantifiers and conjunctive normal forms) facilitates the correct behavior and implementation of the surprise mechanism, but restrict the kind of concepts that can be represented. The use of other FOL components, however, must be cautious. For example, the inclusion of existential quantifiers implies a search in the domain of the variables affected, which could result in an impractical approach.

Qualitative models, on the other side, are restricted to observable phenomena that can be represented by the rate of change in the variables analyzed. Surprising events, however, occur in specific time-points that can not be modeled qualitatively. The solution provided by the presented approach is to extract information not only at the moment of surprise but during the span of the action executed by the robot. While this proved enough for simple situations, more complex ones might require different mechanisms to obtain the information needed, to be compared with the model.

From these observations, it seems clear the need for a comparative study on the knowledge representations selected, as well as other state of the art alternatives, such as: semi-quantitative models, decision trees or bayesian networks. Such study should shed a light on the best representation(s) to be included in the computational model of robotic surprise.

Another observation can be made regarding the different time scales of the surprise mechanism and the knowledge available in the *long-term memory*. The reaction of the surprise mechanism to an unexpected event must be immediate. However, the process of searching the knowledge base for appropriate pieces of knowledge may cause the reaction to be delayed, specially if the robot has already acquired a significant amount of experience about its surrounding world. This thesis has partially solved that problem by associating pieces of knowledge with specific robot actions, such that when those are executed, the knowledge is loaded into memory and can be readily used by the robot, without having to access the knowledge base every time a surprise takes place.

Undoubtedly, this solution opens the door to new scientific issues: how to automatically associate the knowledge to specific actions? wouldn't it be better to associate knowledge to experiments? which knowledge has to be used and why? wouldn't it be better to extend the short-term memory such that some knowledge is partially "forgotten" after some time, resembling more closely the way human memory works? These questions will need to be answered in a near future to add robustness to the model of surprise.

The computational model of robotic surprise has been successfully applied to the domain of a robotic learner, specifically one that learns by experimentation. In the line of the discussion on *long-term memory*, it is important to mention that a surprise mechanism should not be the sole and only means for the robot to react to the stimuli from its environment. The limitations pointed out strongly suggest that such robot, should be equipped with a series of motivations and drives that enable it to react, reason and actuate accordingly to the kind of situation it faces.

This issue is currently being investigated in the context of Project XPERO, and the surprise mechanism is one of the building blocks of such body of (cognitive) mechanisms that enable the robot to experiment with its environment, gaining insights.

Finally, it might be argued that the experiments designed to test the surprise are not complex enough to represent a real-world situation for a robotic agent. However, this scenario has proven to offer severe challenges for a robot with minimal knowledge, to learn a concept that for a human being seems given, namely, movability of objects. Therefore, it can be concluded that the experiments designed present a credible showcase to test surprise in a robot that learns by experimentation. This does not exclude, of course, the future application of the computational model to more complex cases where its limits can be tested, expanded and improved to cover more general situations.

Chapter 6

CONCLUSION

A computational model of robotic surprise that provides a robotic learner, with the appropriate mechanisms to react to unexpected events, initiating a learning activity to obtain new knowledge about the world, has been presented. The model introduces the use of short- and long-term memory as a factor that influences the output of the surprise mechanism. The implementation of the model of surprise will become a building block for a motivational system, used by a robotic agent that learns by interacting and experimenting with its environment.

The model includes a measure of surprise that provides an output of variable intensity, that is a function of the degree of unexpectedness of observed events. The amount of surprise of an event is measured by comparing the data received from the robot sensors (observations), with a model probability distribution. This comparison provides the degree of divergence between both models. If the departure is significative, surprise is triggered.

The measure of surprise is flexible enough to allow the human researcher to incorporate a bias to the surprise mechanism: if the human researcher knows that the situations faced by the robot, are better expressed by a specific model of the probability distribution followed by the sensor data, such model can be incorporated to the surprise measure.

A *short-term memory* mechanism that influences the behavior of the surprise mechanism, has been presented. Such influence is given by the gradual habituation to new sensor input/surprising events. This follows the notion that a surprise mechanism must present a behavior such that a repeated surprise stops to be surprising after some time, as well as an experienced surprise that "grows old", should be "forgotten" for it to become surprising again.

A *long-term memory* mechanism that utilizes available knowledge to influence the output of the surprise module, has been included in the model. The results obtained support the conclusion that background knowledge improves the robustness of the approach, by adding some basic reasoning to the surprise mechanism. If the robot can explain a candidate surprise fired at sensor level, such surprise is not relevant to drive the robot towards the acquisition of new knowledge, and does not represent a significative event that the robot must attend to.

The characteristics of the embodiment where the model is implemented, play a fundamental role while reacting to surprises. Therefore, a mechanism to associate the sensor data to appropriate data filtering and transformation functions has been created. This allows an easier and efficient data pre-processing of multiple sensor input, which is of crucial importance to the measure of surprise.

The computational model of robotic surprise has been successfully applied to the domain of a robotic learner, specifically one that learns by experimentation. The scenarios proposed, where the learner must gain insights about the phenomena it experiences, has proven to offer great challenges for a robot with minimal knowledge, that tries to learn a concept such as movability of objects. Therefore, it can be asserted that the experiments designed based on such scenarios, present a credible showcase to test surprise in a robotic learner. Moreover, the successful results obtained support the conclusion that the model of robotic surprise is a fundamental component in the architecture of a robot that learns by experimentation.

6.1 Lessons Learned

The development of this thesis has provided several insights that might need further attention in future work:

Cognitive and evolutionary robotics are still young disciplines, therefore, the mechanisms based on ideas coming from those areas are yet to mature. In robotic surprise, this condition became evident when researching the state of the art: there seems to be an initial agreement on the basic characteristics of surprise (as most existing computational approaches follow a cognitive model of surprise), however, there is no clear paradigm on how surprise should be applied to a robot, which should be its “standard” components, and how they should be interrelated and/or implemented.

While this is an exciting situation from the scientific point of view, as it allows for a large margin of research and development, it poses several challenges and questions marks, as pragmatic decisions had to be taken in order to obtain satisfactory results. As an example, take the case of the measure of surprise: the absence of a comparative study of the existing measures of surprise applied to a real robot, made the task of determining the “correct” measure, a challenging one.

This work presents an answer and solution to part of the problem, by clearly defining the components of a model of robotic surprise and their relationship however, there is still a need to assess the use of one method over another while implementing each component. While the decisions taken led to successful results in the showcase scenario for a robot that learns by experimentation, the applicability of the proposed mechanisms to other situations might require variations or modifications to the existing methods, hopefully resulting in the establishment of a paradigm in robotic surprise.

Another interesting insight emerged from the application of the model of surprise to a robot that learns by experimentation. The challenges faced by such robotic agent, turned the use of available knowledge to influence the outcome of the surprise mechanism, into a key element to the extent that the best performance of the robot in the implementation of the framework established in project XPERO, was reached when background knowledge was incorporated to the process.

This is an encouraging result, as it supposes an advance in a direction that has been neglected so far. However, it also imposes the obligation to study other knowledge representations, as well as their effect in the kind of surprises that the robot should experience.

6.2 Future Work

The results obtained while completing this work, as well as the discussions that originated from it, indicate several possible directions for future work:

- The measure of surprise relies in several assumptions, derived from the decisions taken on how to implement the model of surprise. Despite the successful results, there is a need to re-assess such decisions, on the light of possible enhancements that add robustness to the approach, e.g. the inclusion of noise models, the assumption of gaussian mixture models (GMM) to represent the model and observed probabilities, and the use of other measure functions in combination with KLD.

- There is a need to assess the influence of short-term memory in the output of the surprise mechanism. A study of the applicability of alternative short-term memory approaches related to artificial neural networks, or those that are biologically inspired, needs to be investigated.

- Long-term memory presents three future directions of work: First, a study and evaluation of the influence of the knowledge representation used (FOL and qualitative models), needs to be performed. This might allow a better understanding of the possibilities of combination of such knowledge, to cover a larger span of situations that the robot might face. Second, other forms of knowledge representation, must also be considered. Representations such as semi-quantitative models and decision trees seem promising enough to, in combination with first order logic and qualitative models, overcome some of their limitations. And third, it is required to investigate how to make the knowledge available to the surprise mechanism, without causing a delay on the triggering of surprise. Possible solutions go in the direction of extending the short-term memory to account not only for sensor readings, but also for more complex knowledge that is readily available, but partially "forgotten" with time, resembling more closely the way the human memory works.

- Imposing the robot's ability to react to external stimuli, to a surprise mechanism only, seems a rather bold decision. Instead, other motivational mechanisms must be analyzed in detail such that their combination with the surprise model, provide a robust motivational system to be used in the framework of a robot that learns by experimentation. Candidate approaches are proactive mechanisms such as attention and curiosity, and reactive drives such as hunger and survival.

- The method that provides the association of sensor readings to data filtering/transformation mechanisms, must be revised to account for more complex situations, such as several layers of data transformation/filtering, and online adjustment/adaptation of processing methods. Additionally, a mechanism that accounts for the concurrent application of different kinds of available knowledge (models), to the action that is being executed by the robot, is needed. A probable consequence of this new mechanisms, is that the subsumption process will need to be revised, modified or even substituted, to account for more complex knowledge structures for which a simple suppression of the surprise output, reduces the power of such knowledge to explain the unexpected event.

BIBLIOGRAPHY

[AB83] J. Archer and L. Birke, editors. *Exploration in animals and Humans.* Van Nostrand Reinhold, London, 1983.

[Bak02] B. Bakker. Reinforcement learning with long short-term memory. In *Advances in Neural Information Processing Systems*, volume 14, 2002.

[Ber60] D. Berlyne. *Conflict, Arousal and Curiosity.* McGraw-Hill, New York, 1960.

[BI05] P. Baldi and L. Itti. Attention: Bits versus wows. In M. Zhao and Z. Shi, editors, *Proc. IEEE International Conference on Neural Networks and Brain, Beijing, China*, volume 1, pages PL56–PL61, Oct 2005.

[Bre98] Cynthia Breazeal. A motivational system for regulating human-robot interaction. In *AAAI/IAAI*, pages 54–61, 1998.

[Bro86] R. Brooks. A robust layered control system for a mobile robot. In *IEEE Journal of Robotics and Automation*, pages 14–23, April 1986.

[CF05] S. Chaffar and C. Frasson. The emotional conditions of learning. In *FLAIRS Conference*, 2005.

[CG87] Jaime G. Carbonell and Yolanda Gil. Learning by experimentation: The operator refinement method. Technical Report Technical Report CMU-CS-87-167, September 1987, 1987.

[GH01] S. Gadanho and J. Hallam. *Emotion triggered learning in autonomous robot control*, volume 32, chapter Grounding emotions in adaptive systems, pages 531–559. Taylor & Francis, a special issue of cybernetics and systems edition, July 2001.

[Gil94] Y. Gil. Learning by experimentation: Incremental refinement of incomplete planning domains. In *Proceddings of the Eleventh Internationsl Conference on Machine Learning*, July 1994.

[Gol05] Derek Goldrei. *Propositional and Predicate Calculus: A Model of Argument.* Springer-Verlag, London Ltd., 2005.

[IB05] L. Itti and P. Baldi. A principled approach to detecting surprising events in video. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 631–637, San Siego, CA, Jun 2005.

[IB06] L. Itti and P. Baldi. Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems, Vol. 19 (NIPS*2005)*, pages 1–8, Cambridge, MA, 2006. MIT Press.

[KO06] F. Kaplan and P-Y. Oudeyer. Curiosity-driven development. In *Proceedings of the International Workshop on Synergistic Intelligence Dynamics*, 2006.

[KT81] D. Kahneman and A. Tversky. Variants of uncertainty. Technical Report ADA099503, Stanford University, CA. - Dept. of Psychology, May 1981.

[Kui94] B. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. Artificial Intelligence. MIT Press, Cambridge, MA, 1994.

[LC07] Emiliano Lorini and Cristiano Castelfranchi. The cognitive structure of surprise: Looking for basic principles. *TOPOI*, 26:133–149(17), March 2007.

[MC99] L. Macedo and A. Cardoso. Towards artificial forms of surprise and curiosity. In S. Bagnara, editor, *Proceedings of the European Conference on Cognitive Science*, pages 139–144, 1999.

[MC01a] L. Macedo and A. Cardoso. Modeling forms of surprise in an artificial agent. In *Proceedings of the 23rd. Annual Conference of the Cognitive Science Society*, 2001.

[MC01b] Luis Macedo and Amilcar Cardoso. The influence of the size of the episodic memory on the surprise-value of the creative agent's products. In R. Webber and C. Gresse von Wagenheim, editors, *Proceedings of the Workshop Program at the 4th. International Conference on Case-based Reasoning*, July 2001.

[MC04] L. Macedo and A. Cardoso. Exploration of unknown environments with motivational agents. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 2004.

[MCK06] R. Maguire, F. Costello, and M. Keane. A cognitive model of surprise judgments. In *Proceedings of the 28th. Annual Conference of the Cognitive Science Society*, pages 531–536, 2006.

[MCR06] L. Macedo, A. Cardoso, and R. Reisenzein. A surprise-based agent architecture. In R. Trappl, editor, *Cybernetics and Systems*, volume 2. Austrian Society for Cybernetics Studies, 2006.

[Mit97] T. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. McGraw-Hill, 1997.

[MJH07] A. Mohan, A. Juarez, and T. Henne. Qpole: A qualitative prediction-observation loop for learning by experimentation. In *6th. EUROSIM Congress on Modelling and Simulation*, 2007.

[MK06] R. Maguire and M.T. Keane. Surprise: disconfirmed expectations or representation-fit? In *Twenty-Eight Annual Conference of the Cognitive Science Society*, 2006.

[MLAS06] S. H. Muggleton, H. Lodhi, A. Amini, and M. J. E. Sternberg. Support vector inductive logic programming. *Innovations in Machine Learning*, pages 113–135, 2006.

[MR94] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679, 1994.

[MRS97] W-U. Meyer, R. Reisenzein, and A. Schutzwohl. Toward a process analysis of emotions: The case of surprise. *Motivation and Emotion*, 21:251–274 (24), September 1997.

[Mun] N. Mundhenk. Bayesian surprise toolkit for matlab.

[OKH07] P-Y. Oudeyer, F. Kaplan, and V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286, 2007.

[OP87] A. Ortony and D. Partridge. Surprisingness and expectation failure: Whatś the difference? In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, 1987.

[Pet98] M. Peters. Towards artificial forms of intelligence, creativity, and surprise. In *Proceedings of the 20 th Meeting of the Cognitive Science Society*, pages 836–841, 1998.

[PJH07] E. Prassler, A. Juarez, and T. Henne. Robotic discovery and learning by experimentation: an overview. Technical report, Univ. of Applied Sciences Bonn-Rhein-Sieg, 2007.

[Rei00] R. Reisenzein. *The Subjective Experience of Surprise*, chapter 15, pages 262–279. The message within: The role of subjective experience in social cognition and behavior. Psychology Press, 2000.

[Rei07] R. Reisenzein. *Oxford companion to the affective sciences.*, chapter Surprise. Oxford University Press, 2007.

[Rem07] C.W. Rempis. Short-term memory structures in recurrent neural networks, April 2007.

[RG] M Reggiani and L. Grespan. Eddy: an educational robot device.

[RN03] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 2003.

[TBD05] S. Thrun, W. Burgard, and F. Dieter. *Probabilistic Robotics.* Intelligent Robotics and Autonomous Agents. MIT Press, 2005.

[Vel99] J. Velasquez. An emotion-based approach to robotics. In *Proceddings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999.

[YW04] P. Yue and A. Waibel. Minimum kullback-leibler distance based multivariate gaussian feature adaptation for distant-talking speech recognition. *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, 1:I–1029–32 vol.1, 17-21 May 2004.

GLOSSARY

Atomic sentence In predicate logic: a sentence that is true in a given model and under a given interpretation.

Complex sentence In predicate logic: a sentence that is composed of other sentences (either atomic or complex) and logical connectives.

GMM Gaussian mixture models, a statistical method intensively used for clustering and probability density estimation. A common application is the sampling of a large population that is composed of several sub-populations, where each sub-population is known (or approximated) to follow a Gaussian distribution.

FOL First order logic is a deductive system that uses a formal language interpreted by mathematical structures.

FOL models First order logic models are composed of terms, sentences, constants and functions related by logical operators and quantifiers. Such models are capable of expressing concepts, states of the world, etc. by forming expressions that feature one or several components e.g. $\exists apple\ (\mathrm{Fruit}(apple) \wedge \mathrm{Red}(apple))$.

Habituation Habituation refers to the ability of the surprise mechanism to adapt to constant surprises/new sensor values. This is achieved by the application of a short term memory that allows to "forget" old measurements, generating the desired adaptation effect.

ICE Internet communications engine, a middleware for distributed programming that provides a high-performance communications platform, including layered services and plug-ins.

ILP Inductive logic programming, a subfield of machine learning which uses logic programming as a uniform representation for examples, background knowledge and hypotheses.

KLD Kullback-Leibler divergence of two probability distributions. Measures how different is one probability from another based on a non-simmetric application of information gain.

Knowledge base See *long-term memory.*

Learning by experimentation A learning paradigm where an intelligent autonomous agent learns by interacting and experimenting with objects it finds in its environment, gaining new insights about the surrounding world and physical phenomena.

Long-term memory A non-volatile storage of unlimited capacity, that contains knowledge acquired by the robot or given *a priori* by a human designer. The information is used to try to explain a seemingly surprising event.

Measure of surprise A mechanism that allows to quantify the amount of surprise present in the sensor data, as a result of an event in the real world. See also *KLD.*

ML Machine Learning, a subfield of artificial intelligence concerned with the question of how to construct computer programs that automatically improve with experience.

MLE Maximum likelihood estimation is a popular statistical method used to fit a mathematical model to some data. It is used to estimate the Gaussian probability distribution of the observed sensor data.

Open-ended learning Learning methodology that is not limited to a target concept, a particular learning level, or a particular purpose. Instead, the learner's target concept naturally arises intrinsically or from its interaction with the environment. agent applies its own learning strategies .

Robotic learner Refers to a robotic agent designed to autonomously learn and acquire new knowledge from its environment, in order to survive by adapting to new/changing situations.

Shannon's entropy A measure of the information contained in a piece of data: it's the minimum average message length in bits, that must be sent to communicate the true value of a random variable to a recipient.

Short-term memory Storage system of limited capacity, fast access and temporal nature, that contains recent information on sensor readings, used to estimate the probability distribution of observed data.

Subsumption In robotics: term introduced by Brooks et al. [] to designate a robot architecture that organizes robot "behaviors" into layers, where higher layers equal to higher abstraction. Each layer's response subsumes (inhibits) the underlying layer's response.

Surprise Divergence between an observation and a prediction. Experience derived from unexpected situations perceived by the robot.

XML Extensible markup language, a general-purpose markup language that allows users to define their own tags. Its primary purpose its to facilitate sharing of structured data across different information systems.

Appendix A

MEASURE OF SURPRISE: AN EARLY TEST

This appendix presents an early test to the plausibility of the proposed measure of surprise, as well as a comparison with another closely related model of surprise: the Bayesian Surprise model [IB05, IB06]. To achieve this, an experiment was designed in simulation using Player/Stage simulation tool. The experiment does not feature a robotic learner, but an autonomous robot with predefined (fixed) behaviors. More details are presented in the following sections.

A.1 Simulated Environment

The simulated environment created using Player/Stage/Matlab, consists of a two dimensional world limited by walls, creating a rectangular arena of 8X4 meters. The robotic agent is a Pioneer robot equipped with differential drive, odometry sensor, and a sick laser scanner (aperture angle of 60 degrees, maximum range of 2 meters and precision of 0.001 meters). Besides the robot, the only object used in the experiment is a green cube of 0.5X0.5 meters.

The sensor information available are the range finder readings, namely, the distance to objects located to the front of the robot; as well as the distance travelled, which is retrieved directly from the odometry sensor. Figure A.1 presents a snapshot of the environment described.

A.2 Experiment Description

The robot can perform a single built-in action, namely, moving forward. The situation that the robot faces is: starting from a random position on the left half of the arena, move forward until a contact is detected. The initial position of the robot is randomly calculated for both x and y coordinates inside the left half of the arena. Contact is defined as the detection of an object at a distance less than 0.3 meters from the front of the robot (by means of the range finder information). The speed of the robot is set to a constant value of 0.1 m/s. Sensor information is retrieved every second (1 Hz).

The experiment follows the procedure indicated in Section 4.1, namely, estimation of the new (observed) probability distribution, comparison with the model probability distribution and update of the model probability distribution (if needed).

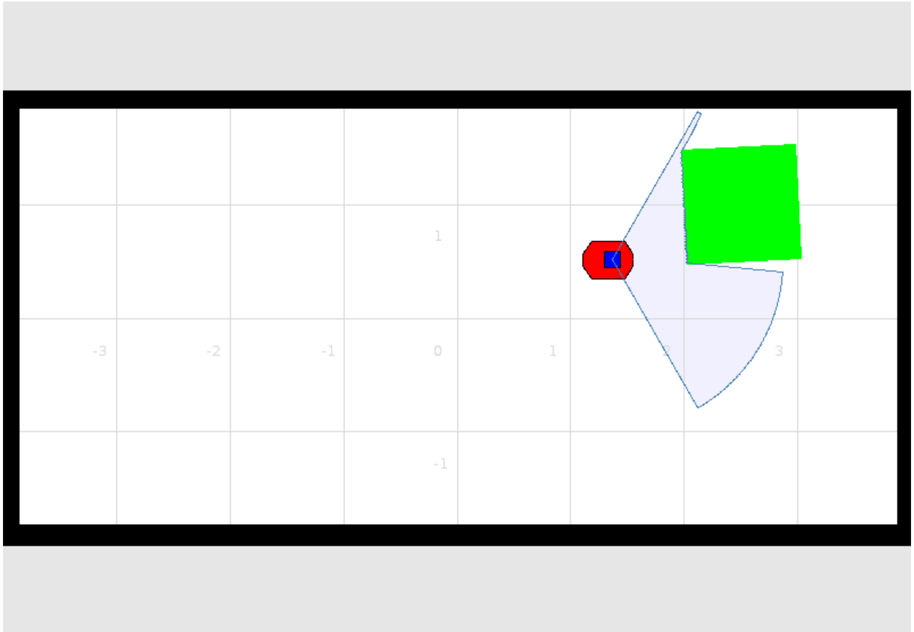


Figure A.1: Simulated environment for the first experiment: a test of the plausibility of the computational model of robotic surprise.

Initially, the arena is empty and the robot moves forward until it makes contact with the opposite wall. After some time, an object is "dropped" into the arena. A surprise should quickly arise as the robot starts to make contact with the cube occasionally. This means that the probability distribution associated to the distance that the robot is able to cover without bumping starts to change, and the measure of surprise will show an increasing value.Finally, the object is removed from the environment.

A.3 Results: Measure of Surprise Tests

Two tests demonstrate the feasibility of the proposed measure of robotic surprise: a first test that applies the basic formula for the KLD (Equation 4.1) to the probability distributions, obtained with and without the presence of the green box in the environment; and a second test that assumes a Gaussian distribution of the observed data and applies the corresponding KLD formula (Equation 4.3) sequentially.

The initial test has the objective of determining if the KLD measure presents the expected behavior of surprise, namely increase in magnitude when the object is dropped into the box, and gradually decrease when the object is withdrawn. The idea behind the test is simple: obtain a probability distribution from many repetitions of the robot actions without the object present in the environment. This will be the model probability distribution. The results of this procedure are shown in Figure A.2. The distribution was obtained using a Matlab module that implements a binning algorithm to group the travelled distances into 10 classes.

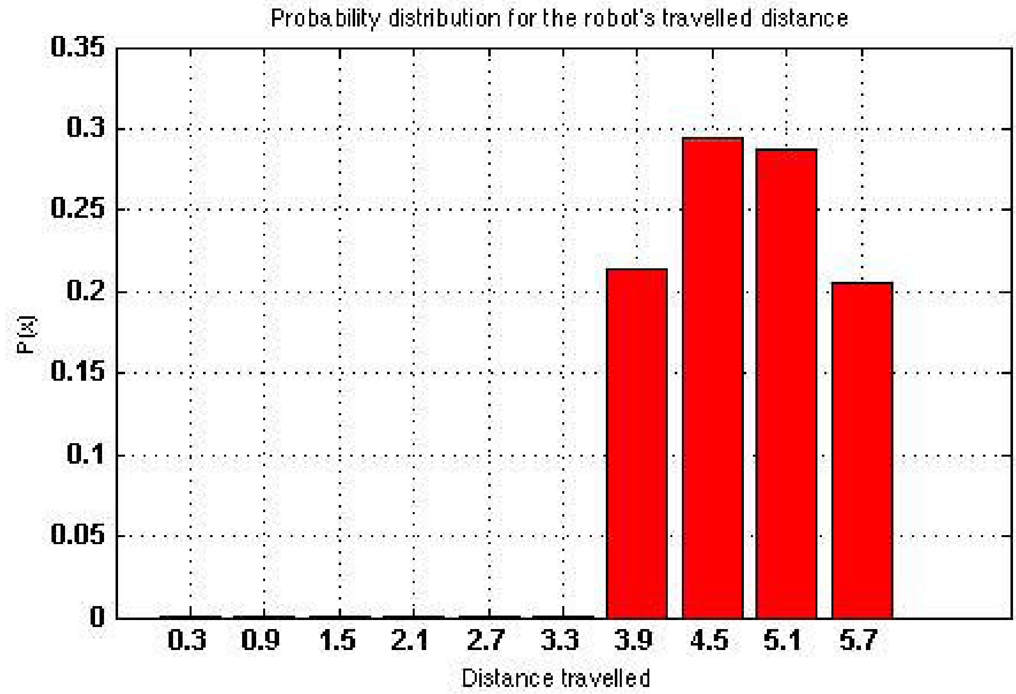


Figure A.2: Model probability distribution obtained from approx. 5000 runs.

Once the model probability has been obtained, we perform the first two steps required to measure surprise: estimation of the new (observed) probability distribution, comparison with the model probability distribution. The third step (update of the model distribution) is not performed given that we have prior information regarding the model distribution, i.e. it has been already obtained and it is assumed that it is the true distribution.

Initially, the environment is the same used to obtain the model distribution (no cube in the box). For each run, the observed probability is compared to the model distribution. This produces no surprises as expected.

Afterwards the box is dropped in the environment as the robot continues executing its predefined actions . With each run, a copy of the model distribution is updated and compared it to the original model. Figure A.3 show the updated probability distribution after approx. 4000 runs, and before the object is removed from the environment. The update of the probability distribution is carried out by the same Matlab module used in the previous phase.

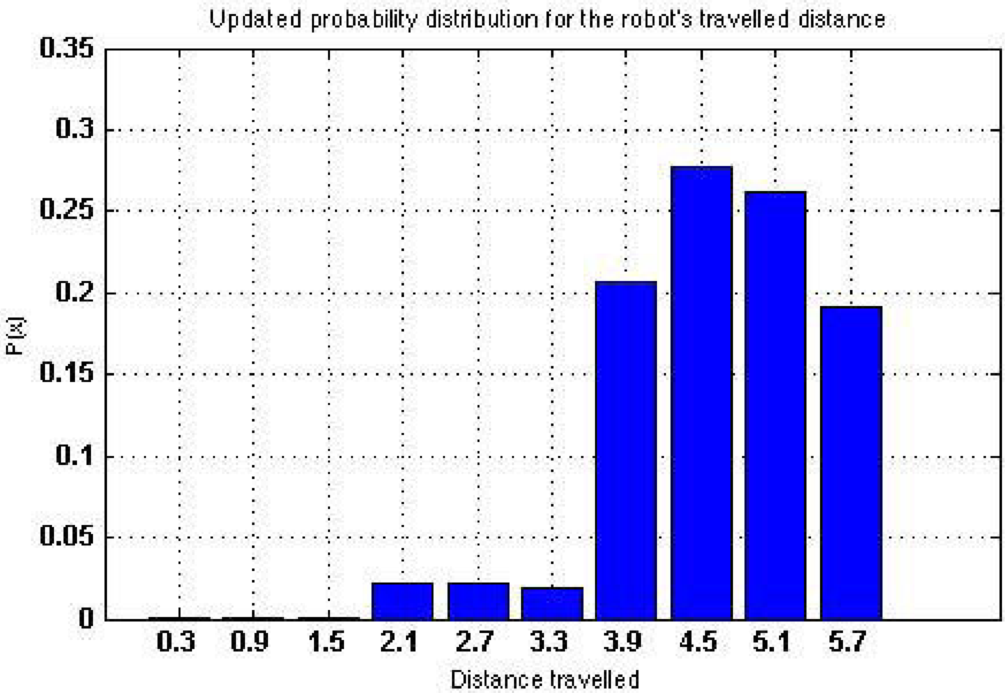


Figure A.3: Updated probability distribution obtained after approx. 4000 runs.

Finally, the box is removed from the environment causing the probability distribution to start going back to "normal", this is, the magnitude of the surprise response decreases over time as more runs with no object present, are executed by the robot. Figure A.4 shows the behavior of the surprise value through the phases previously described: observed probability distribution where no object is present (input 0-2700), an object is dropped in the environment and the new observed probability is compared to the model, experiencing an increase in the surprise value (input 2700-4000), and finally, the object is removed and the surprise starts decreasing (input 4000 onwards).

The second test that demonstrates the feasibility of the proposed measure of surprise, assumes that the observed data follows a Gaussian distribution. To come closer to a situation that would be faced by a robotic learner, no model distribution was obtained previous to the experiment. This means that, unlike the previous test, the model distribution will be obtained as the robot executes its predefined actions.

This also means that the third step for measuring surprise (update of the model probability distribution), has to be carried out, and that the resulting updated distribution is used as the model for the next run of the experiment. The effect of this set up is that

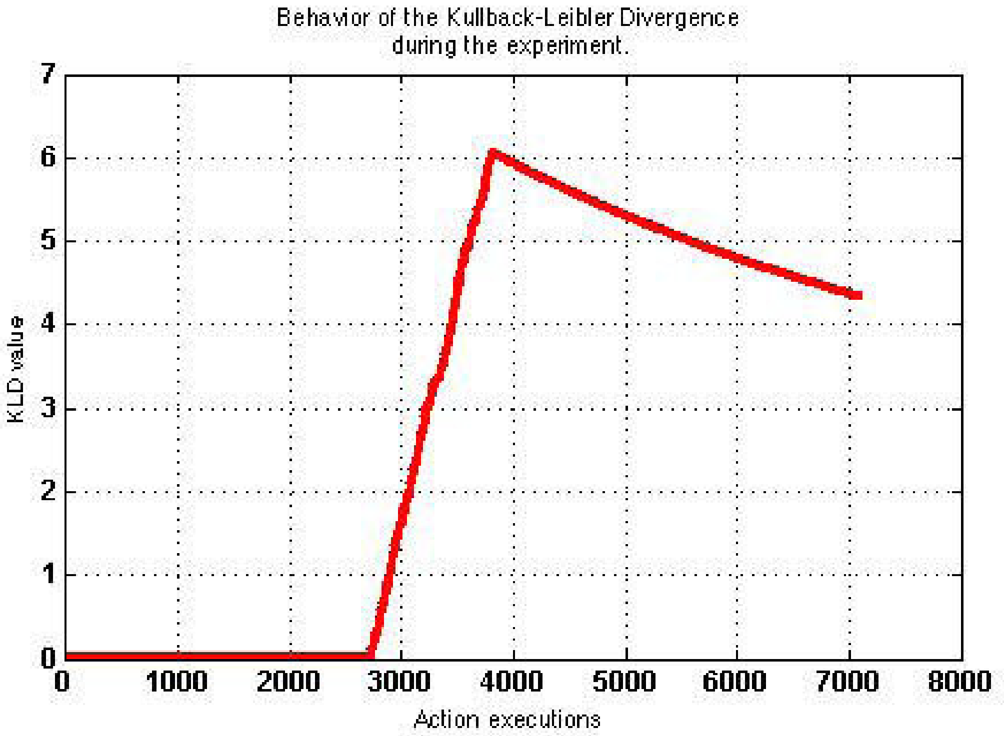


Figure A.4: Behavior of the KLD measure of surprise through the three experimental phases.

the habituation to a new behavior (expressed by the change in the travelled distance), is present. This contrasts with the initial test where there was no habituation to the new situation, as we considered the priori probability as the "true" distribution, and we updated only a "copy" of such distribution with the new information. Both setups are possible in an autonomous robot, and covered by the model, depending on the knowledge available and/or a design decision of which to use.

Figure A.5 presents the results obtained by applying the second test to the same experimental data. For legibility reasons, only 2500 runs of the 7500 available are presented, however, the data includes the introduction of the object into the environment as the robot executes its predefined actions.

The memory size for the experiment, was set such that the robot remembers the data from the past 15 actions. No prior probability was provided to the robot regarding the input data thus, it was constructed as the robot received sensor readings. No knowledge was available in the long term memory that could explain the surprises, therefore no subsumption was triggered. In the graph can be observed the estimated mean (light blue) and

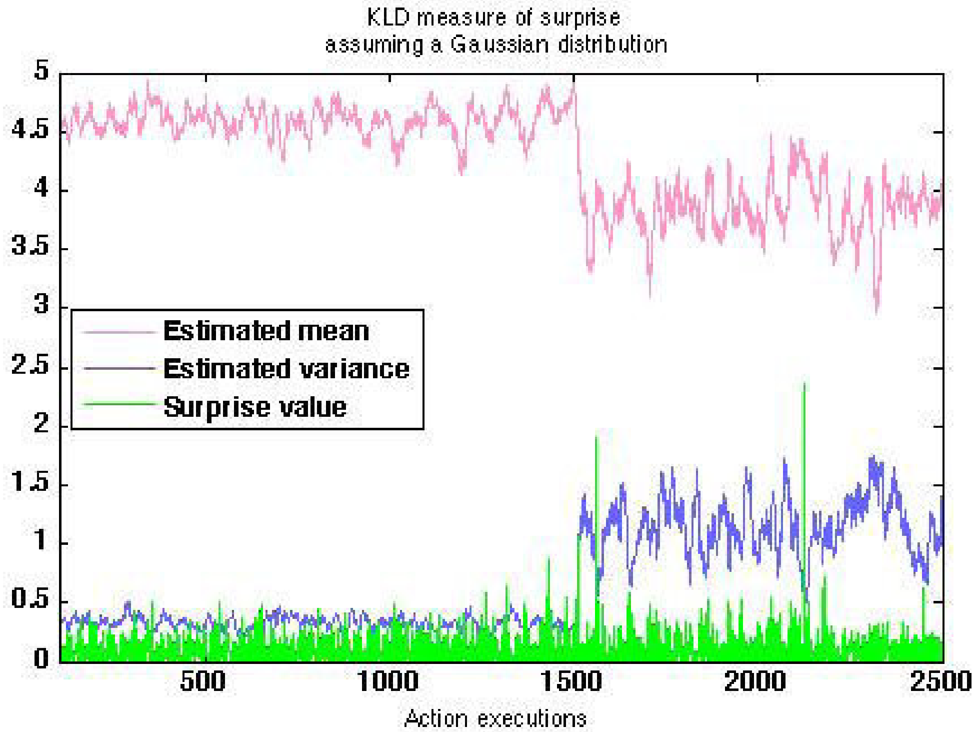


Figure A.5: Behavior of the KLD measure of surprise assuming that the input data follows a Gaussian distribution.

variance (light pink) as well as the surprise value obtained via the KLD measure (green).

It is important to notice the behavior of the measure: the surprise values keep a reasonably low value as the data received from the sensors correspond to the "normal" setup. Once the object is introduced (between run 1500 and 1510), there is a peak in the surprise value that slowly goes back to normal once the robot adapts to the new probability distribution and "forgets" about the previous behavior. This can be see in a clearer way in Figure A.6

A.4 Results: Applying the Bayesian Surprise Model

Section 2 mentioned an approach by Itti et al. [IB05, IB06], which ideas form the base of the proposed measure of surprise. It is interesting at this point to show how this approach behaves when fed with the sensor data used to test out measure of surprise.

This new test was performed by using the Bayesian Surprise Toolkit for Matlab [Mun] which implements the approach described in [IB06]. Figure A.7 shows the output received from the Matlab procedure that implements the Bayesian Surprise (BS) approach. The

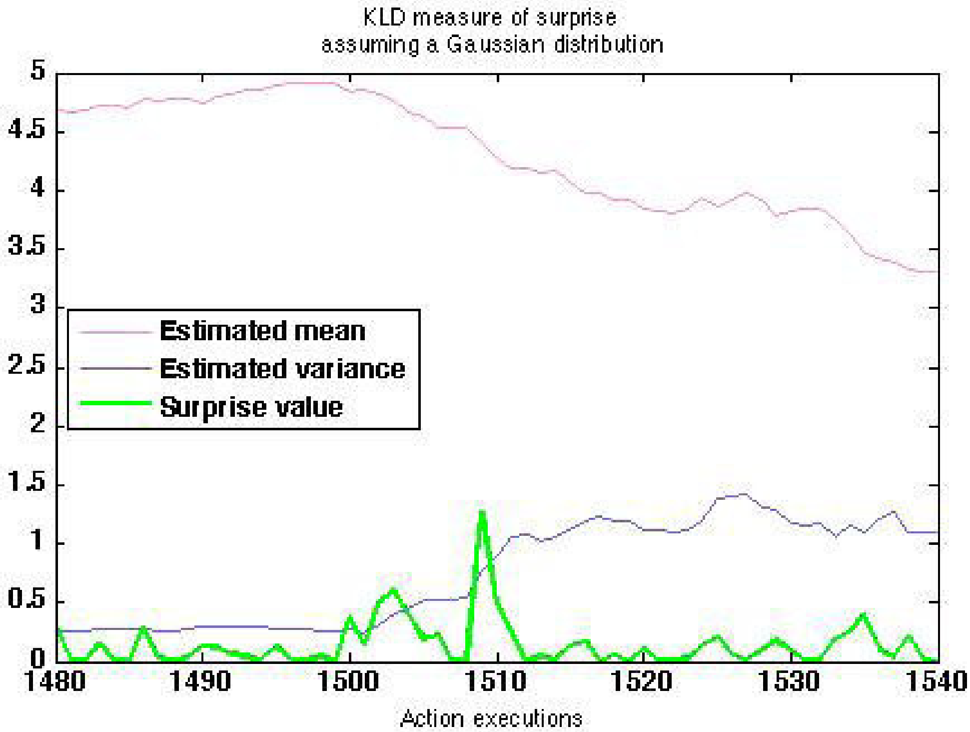


Figure A.6: Detail of behavior of the measure of surprise when the surprising event occurs.

surprise value shows a behavior that follows the original "signal", this is the sensor input. When the object is introduced into the environment (around run 3000), reducing the distance travelled by the robot for some of the runs, the surprise values far from experiencing a peak, suffer a drop when "following" the input signal.

An explanation for such behavior, is that the BS approach models the input data as a Poisson distribution, this is, as counts of successful observations for a given period of time. The reason for this modeling is that the BS approach was designed to be applied to saliency maps of images, where saliency is defined as the concentration (count) of "interesting" pixels in a determined spatial image region. This count becomes the input of the BS mechanism to detect surprises across frames in video images. A clearer picture of the behavior of BS when applied to the test experiment can be seen in Figure A.8.

From the observed behavior, we can conclude that the BS approach in its original form is not suitable for situations like the one modeled in the test experiment, a situation that is valid for an autonomous robot, and that is covered by the computational model of surprise proposed in this Thesis.

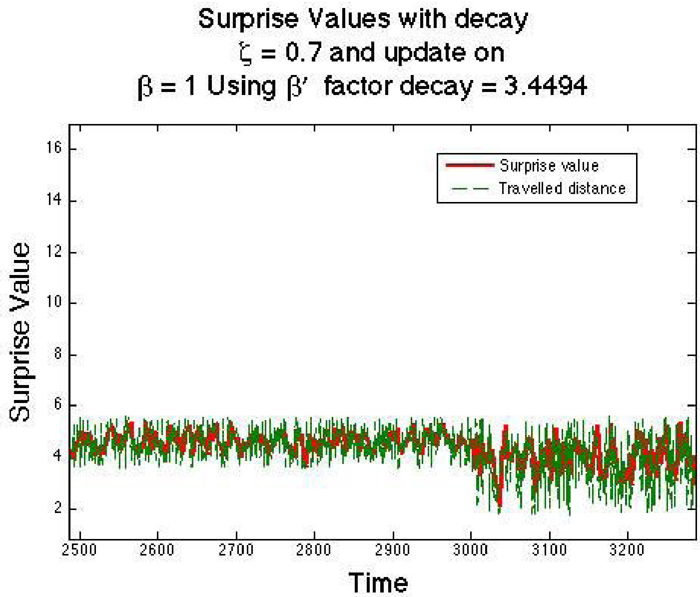


Figure A.7: Result of applying the Bayesian Surprise approach to the data used in the experiment.

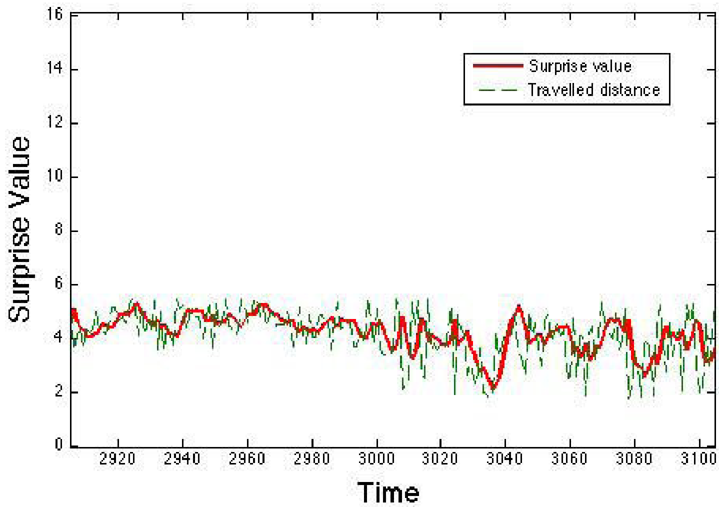


Figure A.8: A zoom into the Bayesian Surprise approach result (see Figure A.7), when applied to the data used in the experiment.

Appendix B

LISTINGS

This appendix presents listings containing XML representations of knowledge available to the robot, in its knowledge base.

Listings related to first-order logic models constructs (Section 5.4.1)

Listing B.1 shows the XML representation of the expressions shown in Equation 5.1 (see Section 5.4.1). Here each atomic formula is given by the nodes formed by the tags labeled *atom*. The variables associated to each expression are delimited by the tags labeled *component*. The information on the condition that makes the atomic formula to evaluate to *true*, is found on the node labeled as *condition*

```
<atomic_formulas>
 <atom name="moving_robot">
     <component type="robot"></component>
     <condition sensor="pose" value="change"></condition>
 </atom>
 <atom name="moving_object">
     <component type="object"></component>
     <condition sensor="pose" value="nochange"></condition>
 </atom>
 <atom name="contact">
     <component type="robot"></component>
     <component type="object"></component>
     <condition sensor="bumper" value="false"></condition>
 </atom>
</atomic_formulas>
```

Listing B.1: XML file containing the atomic formulas available to the robot as knowledge to be used when trying to explain surprises.

As an example, consider the third node labeled as *atom*. The information in the node is interpreted by the robot as “this is an atomic formula named *contact*, which relates a *robot* and an *object*, and the condition for this relation is that the *bumper* sensor value must be *false*. In case of a surprise generated by bumper sensor data in the robot, the robot looks for any condition that might offer an explanation to it, and that is related to the appropriate component. If it finds any condition that satisfy the observation, this means that the atomic formula can explain the surprise experienced by the robot, and therefore, it is not a "real" surprise, and the response recorded by the measure of surprise is subsumed.

```
<complex_formulas>
 <formula name="P1" type="conjunction">
     <atomic name="moving_robot" value="true" ></atomic>
     <atomic name="contact" value="false" ></atomic>
     <atomic name="moving_object" value="false" ></atomic>
 </formula>
 <formula name="P2" type="conjunction">
     <atomic name="moving_robot" value="true" ></atomic>
     <atomic name="contact" value="true" ></atomic>
     <atomic name="moving_object" value="false" ></atomic>
 </formula>
</complex_formulas>
```

Listing B.2: XML file containing the knowledge available to the robot in first order logic formulas.

In a similar way as it was done with the atomic formulas, complex formulas were also encoded into XML, in order to be stored in the *long-term memory*, as shown in Listing B.2. A given node representing a formula in the XML tree (i.e. any node labeled *formula*), can be read as: "The formula *P1* is a conjunction of atomic formulas: *moving_robot* with value **true**, *contact* with value **true**, and *moving_object* with value **false**".

Listings related to qualitative models constructs (Section 5.4.2)

The XML representation of the qualitative model shown in Figure 5.8, is presented in Listing B.3. Each node labeled *qmodel* contains the qualitative model for a specific variable, indicating the name and type of that variable, which objects are involved in the comparison of the qualitative trends extracted from sensor data, and what is the predicted qualitative value(s) of the variable throughout the phenomenon.

As an example, consider the first node labeled *qmodel* in the XML tree. This node will be interpreted by the robot as "this is a qualitative model named *QM1*, which corresponds to a variable named *distance_object_robot*. This variable involves a robot and an object for which the qualitative information on the distance between them must be calculated.

Finally, the qualitative behavior of this variable (the distance between the robot and the object), shows an initial decreasing trend, followed by a steady trend". This qualitative prediction corresponds to that of a robot that is getting closer to another object until it touches it and starts moving it along.

```
<qpredictions>
<qmodel name="QM1">
    <qvariable name="distance_object_robot" type="distance">
        <component type="robot" name="Robot1"></component>
        <component type="object" name="Object1"></component>
    </qvariable>
    <qtree>
        <qnode>
            <qvalue value="dec"></qvalue>
            <successors>
                <qnode>
                    <qvalue value="std"></qvalue>
                </qnode>
            </successors>
        </qnode>
    </qtree>
</qmodel>
<qmodel name="QM2">
    <qvariable name="robot_disp" type="distance">
        <component type="robot" name="Robot1"></component>
        <component type="origin" name="Origin"></component>
    </qvariable>
    <qtree>
        <qnode>
            <qvalue value="inc"></qvalue>
        </qnode>
    </qtree>
</qmodel>
<qmodel name="QM3">
    <qvariable name="obj_disp" type="distance">
        <component type="object" name="Object1"></component>
        <component type="origin" name="Origin"></component>
    </qvariable>
    <qtree>
        <qnode>
            <qvalue value="std"></qvalue>
            <successors>
```

```
                  <qnode>
                      <qvalue  value="inc"></qvalue>
                  </qnode>
              </successors>
          </qnode>
      </qtree>
</qmodel>
</qpredictions>
```

Listing B.3: XML file containing the knowledge available to the robot in first order logic predicates.

Listings related to embodiment and sensor data processing (Section 4.4)

Section 4.4 presented a mechanism that allowed to associate specific data processing mechanisms to the data coming directly from the sensors. Listing B.4 shows an example of such mapping, represented as an XML tree that is stored in the knowledge base. Each entry of such mapping contains information on the type and id of the object we are referring to. In case that it refers to a robot, it provides the kind of sensors it has, identifiers for both the robot and the sensors, and the kind of processing that each sensor reading deserves.

```
<mappings>
   <object  name=“Robot”  type=“robot”>
    <sensor  name=“LocRobot1”  type=“pose”  process=“disp”></sensor>
    <sensor  name=“BumpRobot1”  type=“bumper”  process=“boolean”>
        </sensor>
   </object>
   <object  name=“Object1”  type=“object”>
    <sensor  name=“LocObject1”  type=“pose”  process=“disp”></sensor>
   </object>
   <object  name=“Object2”  type=“object”>
    <sensor  name=“LocObject2”  type=“pose”  process=“disp”></sensor>
   </object>
</mappings>
```

Listing B.4: XML file containing a mapping from sensor data to processing mechanisms.

To illustrate how the mapping works, consider the first entry in the XML code shown in Listing B.4. This entry states that in the environment there exists an object named “Robot1” of type “robot” , which refers to the robot itself. This robot is equipped with two “sensors”, which refers not necessarily to the sensor itself, but to the data coming from the sensors (which in some cases will be equivalent, e.g. bumper sensor). The first sensor is of type “pose”, which tells the robot to look for the data coming from the robot and select only that data labeled as “pose” data, i.e. the position information. The processing mechanism associated to such data, is “disp” which tells the robot to use its internal function to calculate the displacement between the last and current position. A similar reading can be done for the other sensor: the robot looks for the sensor input labeled “bumper”, and processes the data as a “boolean”, telling the robot that this is a boolean variable, and that it should use its internal function to convert such boolean to a numeric value, that can be used by the surprise measure.

LIST OF FIGURES

LIST OF TABLES