Grounded Sensorimotor Interaction Histories for Ontogenetic Development in Robots

Naeem Assif Mirza

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

Doctor of Philosophy

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

April 2008

for

Jacqueline, Yousef, Samir and Kamran

Acknowledgements

Throughout my studies, I was unusually fortunate to have three wonderfully enthusiastic, knowlegeable and friendly superviors: Professor Chrystopher Nehaniv, Professor Kerstin Dautenhahn and Dr. René te Boekhorst, and I thank them for their invaluable support, infectious enthusiasm and willingness to share their vast experience. I have also been fortunate to have forged great academic and personal friendships with many fellow researchers in the Adaptive Systems Research Group, and can honestly say that these past few years would have not been nearly as much fun without them. Thank you Alex, Aris, Joe, Mike, Rob and the many others. I would particularly like to thank Lars Olsson who has been a real inspiration, epic travelling partner and true friend.

The RobotCub[1] project has played a pivotal role in how this research has been shaped, not only in terms of funding, but also in terms of the many intellectually challenging and inspirational discussions with the RobotCub partners.

Finally, none of this would be possible without my wife Jacqueline. At the start of this journey our eldest son was three years old and the twins were but an imagination. She has selflessly raised our three wonderful boys, and yet has always been there, picking me up and cheering me on. Thank You.

NAEEM ASSIF MIRZA

[1]EU Integrated Project RobotCub ("Robotic Open-architecture Technology for Cognition, Understanding, and Behaviours"), funded by the EC through the E5 Unit (Cognition) of FP6-IST under Contract FP6-004370.

Abstract

This thesis puts forward a computational framework that can be used by embodied artificial agents (and in particular autonomous robots) for ontogenetic development. The research investigates methods, endowed with which, an embodied agent can develop control structures for increasingly complex and better adapted behaviour, explicitly and incrementally from its history of interaction with its environment. The temporal horizon of an agent is extended so that past experience can be self-organized into a developing structure that can be used to anticipate the future and act appropriately in environments where state information is incomplete, such as a social environment.

A formal definition of sensorimotor experience is given, and Crutchfield's information metric is used as the basis for comparison of experiences. Information metrics are demonstrated to be able to characterize and identify time-extended behaviour. A definition of a metric space of experiences is followed by the introduction of an architecture that combines this with environmental reinforcement as the basis for a system for robot ontogeny.

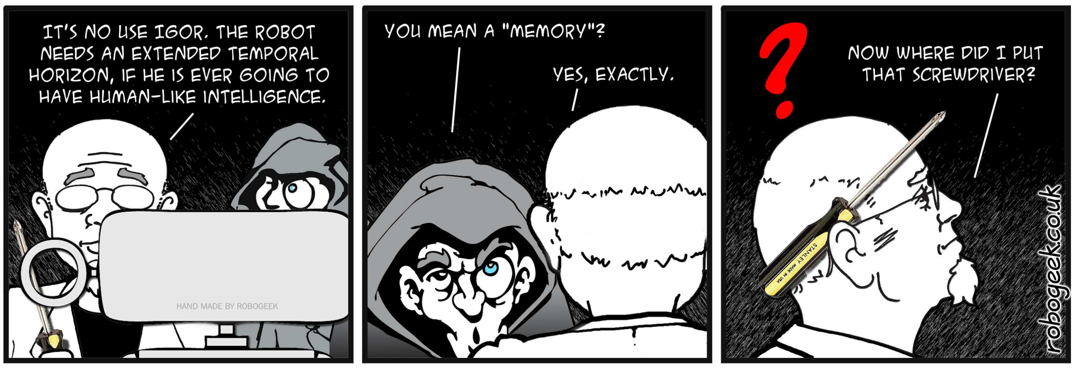
The architecture is demonstrated and tested in various robotic and simulation experiments. This thesis also introduces the early communication game "Peekaboo" as a tool for the study of human-robot interaction and development. The interaction history architecture is then used by two different robots to develop the capability to engage in the peekaboo game.

A joyful life is an individual creation that cannot be copied from a recipe.

Mihály Csíkszentmihályi,
Flow: The Psychology of Optimal Experience, 1990

His mother had often said, When you choose an action, you choose the consequences of that action. She had emphasized the corollary of this axiom even more vehemently: when you desired a consequence you had damned well better take the action that would create it.

Lois McMaster Bujold,
Memory, 1996



Kirk Valladares,
robogeek.co.uk, 2008

Contents

Chapter 1

Introduction

The research focus of this thesis is on the investigation and design of methods, endowed with which, an embodied agent can develop control structures for increasingly complex and better adapted behaviour, explicitly and incrementally from its history of interaction with its environment. This is conducted with the long-term goal of a general framework for ontogeny in robots and other embodied artificial agents.

This is an endeavour in pure Artificial Intelligence (AI); it is, after all, the underlying goal of all AI research to produce artificial entities that exhibit something that might be called "intelligent behaviour". By taking a *developmental* viewpoint however, the researcher avoids many of the problems of design and construction of "old-style" AI, and has the luxury of being able to sit back while the artifact does the hard job of building and developing their own intelligence. The researcher, instead of being an all-knowing designer, takes up the rather more tractable role of the teacher. Now, of course, the question is, what should be put into that artifact such that it can turn that teaching into intelligence?

Answering this question is the goal of the relatively new research field of developmental artificial intelligence[1] (Lungarella, Metta, Pfeifer and Sandini, 2003)

[1]Due to the embodied, situated nature of the field, this has become synonymous with Developmental Robotics, or Epigenetic Robotics, and is a research field that is now supported by at least two dedicated international conferences: the IEEE International Conference on Development and Learning, and the International Conference on Epigenetic Robotics.

and there has already been a great deal of work towards coming up with answers (and, it has to be said, there probably is no single way of achieving this). Being a new research field, it naturally draws from many disciplines, notably Developmental Psychology, Neuroscience, Cognitive Science, Robotics and Philosophy, the synthesis of which has never been seen before.

Increasingly, the importance of embodiment and situatedness within complex and rich environments is becoming recognized as a crucially important factors in engendering intelligence in an artifact (see for example Clancey (1997); Pfeifer and Bongard (2007) and the philosophical position regarding "structural coupling" of Maturana and Varela (1987)). Moreover, it is in how an artificial agent develops its capabilities over a life-time of interactions (*ontogeny*) that is important in building a *grounded* intelligence, especially given the complexity of interactions in natural environments, and the richness of sensors available to modern robots. Grounding ("the symbol-grounding problem", Harnad, 1990) has long been a problem for AI. For symbols to have meaning for an artifact, they must be grounded in its own interaction with the real-world. An artificial agent that develops everything it "knows" through interactions with its environment, building a rich history of interaction grounded in its own sensorimotor experience, may avoid the problems of ungrounded symbolic artificial intelligence and take a step towards a grounded natural intelligence.

1.1 Research Questions and Challenges

The central thesis that is proposed, is

> *that ontogeny in an embodied agent can be based on, and built upon, a grounded sensorimotor history.*

Thus, the ultimate goal that this research hopes to contribute to is the realization of a general framework for behavioural ontogeny in embodied artificial agents. Within this larger endeavour, I identify the following research goals:

Goal 1: To add some formalism to key concepts in the ontogenetic paradigm. For example:

- What defines an experience for an embodied agent?
- What is a grounded history?

Goal 2: To establish quantitative methods for comparison of robot self-experience.

Goal 3: To find, implement and test mechanisms whereby an agent may autonomously and open-endedly shape its control structures for action and behaviour, based on its ongoing history of past experiences.

Thus, the following hypotheses are proposed:

Hypothesis 1: The changing gross informational relationships between groups of sensors of an embodied agent, situated and acting in an environment, can be used to characterize the behaviour of that agent (agent-environment interaction).

Hypothesis 2: It is possible for an agent to recognize its own behaviour in terms of these informational relationships between groups of sensors.

Hypothesis 3: By using a temporally extended history as the basis for action, links between experiences and actions may be built that allow the agent to act such that it exhibits the appearance of prospection of repeated and familiar events in its environment.

Hypothesis 4: A robot can use its own ongoing interaction history to develop the capability to engage in simple, social, communicative interaction with a human partner.

Hypothesis 5: A dynamically constructed history of interactions that is used to generate and select actions in an embodied agent can serve to scaffold the ontogenetic development of the agent.

1.2 Methodology

I take an empirical, constructive approach in this thesis. That is, by implementing architectures and demonstrating their capabilities, I hope to show that there is some validity to the theory, equations and architectures proposed. This stance is taken out of necessity, as I believe that description while often insightful, is ultimately inadequate when one talks of embodiment and experience - it has to be instantiated.

Thus, where possible, architectures and methods are implemented on real robots in natural environments. Simulations are only used as a stepping stone towards that. However, where theory and implications are discussed, I emphasize that they are equally applicable to a more general "embodied agent" and not just robots. Thus, implementations of proposed theory and architecture in a software agent in an immersive game world, or an internet "bot" inhabiting a world of data, are equally valid.

One of the major research tools employed in this research work is Information Theory, due to the enormous potential of these techniques in organizing and understanding relationships of real sensors (Olsson, Nehaniv and Polani, 2004; Lungarella, Pegors, Bulwinkle and Sporns, 2005; Sporns and Pegors, 2004). (Olsson, 2006, Chapter 7) shows that the information-based method is suited to finding relationships between robotic sensors and is "better", in this respect, than other measures. Brief comparisons to some other measures are made in this thesis (See Section 4.3.3), however a thorough comparison is not attempted here.

1.3 Contributions to Knowledge

The main contributions of this thesis are that it:

1. Defines "Interaction History" from the perspective of autonomous embodied artificial agents;

2. Shows that the information theoretic relationships between a robot's sensors (exterioceptive, interioceptive and proprioceptive) can be used to characterize behaviour (*i.e.* distinguish classes of behaviours) and identify behaviours (as being similar to one or another previously experienced behaviour or behaviour class);

3. Defines the Average Information Distance as a measure of sensory relations;

4. Operationalizes the meaning of "experience" from the perspective of embodied artificial agents and robots;

5. Introduces[2], validates and applies the "experience metric" (an information theoretic measure) to the comparison of experiences in robots, and shows that experiences with low values of the metric correspond to experiences that are similar as judged by an external observer;

6. Develops techniques for self-construction and modification of a metric space of experiences as a model of a temporally extended remembering/memory for robotic control systems;

7. Demonstrates the operation of an architecture, that chooses actions based on proximity of experiences in a growing metric space, on different robotic and simulated platforms and on different tasks;

8. Introduces "Peekaboo" as a tool for research in early communicative interaction of robots with humans and as a scenario in which ontogenetic development can be studied in robots.

[2]Concept of experience was co-developed with Chrystopher L. Nehaniv who also created the mathematical proof that it is a metric. All robotic implementation of the metric is my own individual work.

1.4 Overview of Chapters

Chapter 2: The thesis begins by presenting a definition of an Interaction History for an embodied agent along with a discussion of the research motivation and background literature to support the definition. For convenience of reference, an overview of research work that relates to the main themes of this thesis (developmental/learning architectures that use history) is also collected into that chapter (Section 2.6).

Chapter 3: presents technical background information regarding how robotic sensors can be viewed as information sources and a brief explanation of the Information Distance measure.

Chapter 4: then takes the first steps in using the information distance metric to identify and characterize robotic behaviour using the changing relationships between sensors over time.

Chapter 5: Presents formal definitions of an *experience* and the *experience metric* along with supporting experiments that show how sensorimotor experience can be predicted from a history of experience arranged in a *metric space of experiences*.

Chapter 6: Discusses issues regarding the computational scalability of incrementally constructing a metric space of experiences, merging, forgetting and the construction of grounded categories as emergent classes of experience.

Chapter 7: Introduces a computational scheme for ontogeny in artificial agents and robots, the *Interaction History Architecture*, that has at its centre the metric space of experiences developed over the previous chapters. This is followed by simulation experiments demonstrating robotic learning on a benchmark delayed-response task.

Chapter 8: The early interaction game, “Peekaboo” is introduced and the Interaction History Architecture is implemented in a SONY Aibo robotic dog.

The architecture is used as the basis for the robot to develop the capability to engage in a simplified version of the game. This chapter also further investigates the properties of the architecture in terms of appropriate horizon length of experiences.

Chapter 9: A more general version of the peekaboo game that requires appropriate feedback through audio and visual modalities is the subject of experiments in this chapter. The implementation is on an upper-body humanoid robot that is able to provide feedback to the interaction partner using both gestures of its arms and head as well as non-verbal expressive facial gestures.

Chapter 10: Summarizes and discusses the implications of the results, and outlines future directions and possible applications.

Chapter 2

Interaction Histories

2.1 Introduction

The central theme of this thesis is how embodied artificial autonomous agents can develop action capabilities based on a history of interactions grounded in their sensorimotor experience. This chapter begins by offering a definition of what an interaction history is for such an agent. The motivation for this definition is established by exploring literature from various fields that support an embodied view of cognition and memory that is dynamic, developmental and grounded in individual sensorimotor experience. The definition we formulate is particularly concerned with dynamic "memory" or remembering systems that are part of an embodied whole that encompasses the sensory, motor and control systems.

The contents of this chapter are organized as follows. Firstly a definition of interaction histories is presented and is followed by a review of the background literature that motivates it, separated into that from psychology and that from Artificial Intelligence (AI). Finally, we review the background literature as it relates to two key areas of the experimental work in this thesis: how behaviour and experience can be characterized from the agent perspective, and how a history of experience can be used as the basis for action.

2.2 Definition of an Interaction History

An *interaction history* of an embodied agent is defined, in this thesis, as:

> *the temporally extended, dynamically constructed, individual sensorimotor history of an agent situated and acting in its environment, including the social environment, that shapes current and future action.*

The key aspects of this definition are:

- *Temporal extension*: The overall horizon of an agent's experience extends into the past (potentially including all previous experience available to the agent) and also into the future in terms of prediction, anticipation and expectation.

- *Dynamical construction*: This indicates that the history is continually being both constructed and *re*constructed. Previous experiences are modified (including the recall potential of, and relations between, experiences) in both the processes of "storage" and recall, and potentially affect how new experiences are assimilated into the history in the future.

- *Grounding*: The history need not be symbolic (i.e. recorded in terms of externally imposed representations) and is grounded in the sensorimotor experience of the agent. Beyond innate structures for perception, new representations and categories may emerge in cognitive structures as a result of the agent-environment interaction.

- *Remembering in action*: The process of remembering drives and shapes the choice of current and future action, while also, itself, dynamically re-shaping the structures employed in remembering.

Note that the term *interaction* is used to indicate that this temporally extended history encompasses the sensorimotor history, the history of action as well as the feedback of action on the history. This definition encompasses all kinds of interaction with the environment, but specifically includes the social environment.

2.3 Memory and Remembering

The above definition of interaction histories implies a very specific view of what "memory" is for an embodied agent. That is, it is constructive and thoroughly grounded in embodiment and action, while encompassing both episodic as well as semantic aspects. In this section supporting literature for this view of memory is reviewed. Research from both human psychology as well as Artificial Intelligence (AI) and Cognitive Science is considered.

2.3.1 Remembering in Human Psychology

Types of Memory

In the majority of modern-day memory research (see Tulving and Craik, 2005; Baddeley, Conway and Aggleton, 2002), memory is separated into certain types, largely for convenience but also because often memory manifests (and fails) in different ways, under different circumstances and in many widely differing ways. The first distinction that is usually made is between *short term memory*, STM, and *long term memory*, LTM. STM is of the order of seconds, is associated with conscious awareness and appears to have a very different neurological basis to LTMs which are more durable memories, potentially lasting entire lifetimes. LTM can then be separated into *procedural* or *non-declarative* memory, which includes classical conditioning and memory for skills, and *propositional* or *declarative memory* which includes memory for events as well as "knowledge". Tulving (1983) identifies two kinds of declarative memory: *episodic* memory relating to memory of events or episodes (with a temporal aspect and, usually, a personal aspect), and *semantic* memory relating to "knowledge", "meaning" and "categories" unconnected to any particular event.

In terms of this traditional separation of memory types, interaction histories as defined above would be considered to be episodic in nature. However, it can be argued (see for instance Glenberg (1997)) that categories and "knowledge" may emerge from essentially episodic experience, and that while semantic memory is different in nature to episodic memory, that both are part of a continuum of processes. Interaction histories, then, can also be considered to have a semantic aspect, categories and "knowledge" emerging through the merging of similar experiences combined with the process of basing action on these categorical experiences.

A further category of memory that is of relevance to this thesis, cutting across the categories already described, is *autobiographical memory*, that is, memory that defines the individual. Tulving (2005) describes this as being both episodic in nature, containing time located events of significance to the individual as well a semantic knowledge relevant to the individual built-up from many episodes.

Memory as Recall *versus* Memory as Construction

Traditionally, human memory is seen as being able to store exact representations of scenes and events as actual "memories"; this view most probably arising from the very familiar experience of being able to recall what seems like very exact detail of events from the, potentially, far distant, past. Thus, memory seems like a vast warehouse of stored knowledge, the recall of which just needs the right index to the right shelf. In the late 19$^{\text{th}}$ century, work by neuroscientists such as Paul Broca found that the brain was organized with local functions in specific regions, Broca himself locating "the" centre for speech production in an area of the brain now known as *Broca's region*, by examination of cases of aphasia where that region was damaged. This localized view of brain functions, naturally resulted in a view of memory as being stored in precise places in the brain. (Rosenfield, 1988; Bartlett, 1932)

The view of memory as a fixed storehouse was first seriously challenged by Bartlett (1932): *"The first notion to get rid of is that memory is primarily or literally reduplicative, or reproductive."* (Bartlett, 1932, p204). Bartlett uses the example of a tennis stroke to illustrate how motor memory cannot have stored all possible positions and potential sensory inputs required to produce a stroke in all possible situations. It would be plainly impossible to remember exactly all the strokes one played as every one differs from every other. Instead he suggests that every new stroke is constructed afresh from the current context, as is in the case of recollection of memories of events. (Bartlett, 1932, p201-202)

In *"The Invention of Memory"* (Rosenfield, 1988), with reference to clinical cases, Rosenfield builds on the argument that human and animal "memory" should be viewed as a process of remembering. He argues that recollections of the past are constructed in terms of the present context, and as such, are not localized in fixed places or structures in the brain, but are instead reconstructed as required. Thus, this "constructivist" viewpoint states that memory consists not of static representations of the past that can be recalled with perfect clarity, but rather is the result of a dynamic accretion of interaction with the environment.

However, for the constructivist point-of-view to be consistent, it has also to be able to account for such phenomena as *flash-bulb memories* (Conway, 1995) where remarkable level of detail is recalled about certain "personally significant" and "suprising" events, sometimes years after the event, and without overt rehearsal. Illustrating, Conway points to reports about flash-bulb memories *"containing detailed information concerning people, place, activity and source, and some 'irrelevant' details not usually retained in autobiographical memories."* (Conway, 1995, p59). However, Conway does not completely reject the constructivist conception of autobiographical memory, instead he suggests that the organizing structures for flash-bulb memories are particularly tightly and coherently organized, and the constructive process can retrieve "memories as a whole". The argument that the occurrence of high levels of detail in recall does not preclude reconstruction is however more difficult to sustain when considering "eidetic" or "photographic" memory. While instances of individuals with eidetic memory are disputed, the cases of certain "savants" such as Stephen Wiltshire and Leslie Lemke (Wisconsin Medical Society, 2007) indicate that in exceptional circumstances human memory, usually with a concomitant impairment of some other brain function, can indeed recall (and reproduce) vast detail from a single viewing of a scene or hearing of a piece of music (see, for instance (Miller, 1999)).

It seems likely in fact, that the true picture as far as human memory is concerned, lies between the two. That is, human-memory has the ability to store incredible detail, especially when the situation is extremely emotionally charged, however, for most of us, in most situations, recall is a constructive process resulting from many storage events - effectively filling in and completing detail.

2.3.2 Remembering in Artificial Intelligence (AI)

Dynamic Memory

While the above discussions of memory are based on clinical and psychological studies of human memories, a possibility of instantiation of such memories in artifacts is offered in (Schank, 1999). In earlier work, Schank and Abelson (1977) attempt to explain the organization and "understanding" of experience by proposing scripts, that is *"ldots groups of causal chains that represent knowledge about frequently experienced events."* (Schank and Abelson, 1977, ch. 3). Schank (1999) consolidating earlier research describes *dynamic memory*, which is based on the concept of "reminding", and memory structures that at the same time construct categories as the world is experienced and organize retrieval using those categories. In this revised view of dynamic memory, memory is essentially reconstructive. Categories are learnt and organized through experience and then current sensory input is interpreted in terms of these categories at many levels and memory is reconstructed. The structures used for remembering are Scripts (knowledge sources for controlling inferences in particular situations), Memory Organization Packets (MOPs - an organizing structure) and Thematic Organization Packets (TOPs) at the highest level. Furthermore, these organizing structures help to decide what to pay attention to; predictable, normal events are not "noted" in memory, but ones that do not match well to the structures are used to reorganize those structures. Thus dynamic memory is the *"... process of learning by explaining expectation failures engendered by predictions encoded in high-level memory structures ..."* (Schank, 1999, p17).

Case-Based Reasoning (CBR) (Kolodner, 1993) builds on the concept of reminding, building MOPs from the systems experiences (cases). CBR was derived from the earlier work of Schank and Abelson (1977) and Schank's subsequent research into dynamic memory, and had notable successes in producing knowledge based artificial intelligence systems including CLAVIER, a system for laying out composite parts in a fabrication process. CBR and "Continuous-CBR" are discussed further in Section 2.6.1.

Memory as embodied action

This thesis aligns with the "embodied cognition" hypothesis, that *"cognition is a highly embodied or situated activity and suggests that thinking beings ought therefore be considered first and foremost as acting beings."* (Anderson, 2003). Lakoff and Johnson (1999) argue that all cognition, including representations and memory of categories, eventually grounds out in embodiment. Glenberg (1997) argues that the purpose of perception and memory for the natural environment is to guide action and that even abstract concepts can be interpreted in terms of physical actions and properties. Edelman (1992) also supports an active process view of memory: *"By its nature, memory is procedural and involves continual motor activity and repeated rehearsal in different contexts."* (Edelman, 1992, p120). In general therefore, memory *manifests* itself as embodied action of some kind. That is, it is in actions resulting from recall that one witnesses memory and that recall itself is dependent on embodiment.

Clancey (1997) refers to the phenomena of memory as embodied action as "transactional experience", and considers even "deliberating" itself as an activity.

> "Speaking, visualizing and transforming things in the world occur *over time, in protracted activities, coordinated by cycles* of neural categorization and composition. Creating, manipulating and interpreting descriptions ... involve *a sequence of experiences.* Having an idea - even saying something to oneself - occurs in activity as an experience. ... This contrasts with the folk psychology distinction between think-

ing and acting in the sense of first deliberating and then carrying out a plan." (Clancey, 1997, p218,219, original emphasis).

This view from psychology and cognitive science is supported by the modern Artificial Intelligence (AI) community too. Pfeifer and Scheier (1999) for instance also argue for an embodied situated memory, and memory as re-categorization. The emphasis overall then is on the interaction with the environment and a process view of memory.

Autobiographic Agents

An interaction history, when concerned in particular with individual memory of events and meanings, can be considered to be an "autobiographic memory" (Tulving and Craik, 2005). Dautenhahn (1996) defines an *autobiographical agent*, as "an embodied agent that dynamically reconstructs its individual history (autobiography) during its lifetime". An autobiographic agent may also be able to communicate significant episodes in its past to other agents which could further increase the temporal horizon of the agent and that of others (Nehaniv, 1999*a*). Here the notion of recounting, or communication of that history, is important, particularly in social agents.

Ho, Dautenhahn and Nehaniv (2008) describe an architecture for virtual agents that build a reconstructive symbolic episodic memory of their interactions using bottom-up principles. Interestingly, the agents are able to communicate their autobiographies to other agents in the virtual world, and recount stories constructed from their own and others autobiographies. The authors test agents in a complex world and shows that having and communicating stories and histories in this way results in increased adaptation and success.

2.4 Motivation

In this section we consider and expand upon various other aspects of the definition of interaction histories offerred in Section 2.2 above. The themes reviewed here are the extension of the temporal horizon of an agent, dynamical systems, ontogenetic development and social interaction.

2.4.1 Temporal Horizon and Extension

The *temporal horizon* of an agent delimits the history (whether personal or socially acquired) that an agent has access to (Nehaniv, Dautenhahn and Loomes, 1999; Nehaniv, Polani, Dautenhahn, te Boekhorst and Cañamero, 2002). In humans this horizon is extremely broad, as is demonstrated by our story-telling and recall of long-past events as well as their impact on our present and future behaviour, and is also demonstrated by our ability to plan for situations far in the future, possibly even beyond our own lifetimes. Autonomous embodied artificial agents that make use of interaction histories in guiding their actions can be thought of as extending their temporal horizon beyond that of a simple *reactive agent* (for instance Braitenberg Vehicles (Braitenberg, 1984)). These agents become *post-reactive* systems when acting with respect to a broad temporal horizon by making use of temporally extended episodes in interaction dynamics (Nehaniv et al., 2002).

Emotional state, mood and affect are also mechanisms that lead to broader temporal horizons in animals and potentially artificial agents. Internal state as used in *affective agents* can extend the temporal scope of the agent (potentially indefinitely, but usually for the short or medium term), as previous interactions can affect later actions through the agents' affective state. For example, Avila-García and Cañamero (2005) describe a situation where hormonal (affective) state can modulate action-selection in a competitive two-resource problem where simple reactive action-selection fails. However, in general this approach does not allow for access to episodic historical events and so cannot, for instance, suggest more complex alternative courses of action (Scheult and Logan, 2001).

The temporal horizon for an agent potentially encompasses the entire past history of the agent, although it can be focused on episodes of horizon of arbitrary size. History may inform *forward* temporal extension in, for example, prediction, anticipation and planning. The size of the temporal horizon influencing behaviour can be varied and does vary between natural agents. Some agents, it seems, live only in the present. Examples are the simpler Braitenberg Vehicles that do not have a memory, and perhaps bacteria. For instance, the bacterium *Escherichia coli* can be considered to have a certain minimal level of embodiment (Quick, Dautenhahn, Nehaniv and Roberts, 1999) and 'cognition' (van Duijn, Keijzer and Franken, 2006), and are able, without a nervous system, to exploit fairly simple sensor-motor coupling through limited low-bandwidth channels to achieve reactive behaviour such as chemotaxis.

Research in developmental psychology of human infants points to the importance of anticipation and prediction in the development of cognitive capabilities (see, for example, von Hofsten (1993)). A traditional artificial intelligence approach to achieving this might be to build an internal model of the process or task in question, and then to use that model to predict future states. However, I argue that by using a temporally extended history as the basis for action, links between experiences and actions may be built that allow the agent to act such that it exhibits the appearance of prospection of repeated and familiar events in its environment.

2.4.2 Dynamical Systems

A model of a dynamical system describes how the state of the system model evolves over time, usually by means of a set of differential equations. Depending on the relation, very complex non-linear behaviour exhibited by such systems can be described by the model. Models of non-linear dynamical systems are usually characterized by their stable states (or fixed points) and repeating cycles of states (limit-cycles) and operate in a "state-space".

Cognitive systems can be viewed as processes, patterns and structures of dynamical systems operating in various kinds of state spaces (agent-environment, sensorimotor, perception-action, *etc.*) (Thelen and Smith, 1994; Kelso, 1995; Dautenhahn and Christaller, 1996). Regions and attractors (or structures) of these dynamical systems may reflect interesting areas in terms of remembering and adaptive action. These structures are created by the activity of the dynamical system consisting of the embodied cognitive system and its interaction with its environment.

The coupling of agent and environment in this way is referred to as *structural coupling* (Maturana and Varela, 1987). Moreover, the generation of "good" sensory data for a system is enhanced by sensory-motor coordination, and this structuring of sensory data helps in turn to generate structure in the control systems themselves, see for example (Lungarella and Sporns, 2005).

From an action oriented viewpoint, an agent's interaction with the environment can construct the structures that are used for remembering how to act. Reconstruction, in this context, may then involve altering the detail of the original structures, changing the relative importance of them, or, in terms of dynamical systems, moving and altering the attractors. To illustrate, consider auto-associative Hopfield artificial neural networks (Gurney, 1997). The dynamics of such networks resolve to particular attractors (memories) on presentation of particular inputs. Learning new patterns has an effect on what is already stored, and if the network were able to learn while recalling, recall would also modify "stored" memories.

2.4.3 Ontogenetic Development

Ontogenetic development in artificial and natural organisms can be seen as an incremental, possibly open-ended, self-organizing process of change where an organism refines its current capabilities by using internally generated drives and motivations and exploration of its environment and embodiment to generate new goals, capabilities and behaviours (Lungarella et al., 2003).

Human developmental psychology research teaches us that learning and development however proceeds best when tasks that are being learnt are only just beyond the developmental capability of the learner. It is this situation, where a child learns through social interactions (with a teacher), that Vygotsky (1978) refers to as the "zone of proximal development". Thus, human development is continually scaffolded by building capabilities on top of existing, mastered ones. Learning proceeds at the periphery of known experience and already mastered interaction skills, enabling a progressive development of capabilities keeping pace with unfolding physical development.

Blank, Kumar, Meeden and Marshall (2005) identify three essential mechanisms in their "intrinsic developmental algorithm": *abstraction*, the ability to find relevant features from high-dimensional sensory data; *anticipation*, to go beyond simple reactive control; and *self-motivation*, pushing the system toward further abstractions and more complex anticipations. These mechanisms are implemented using self-organizing maps for abstraction and an Elman-style simple recurrent network for anticipation and production of appropriate actions. While their system is limited, not least due to the number of training cycles required and the lack of a demonstrated self-motivation system, the principles of development proposed are very interesting.

I hypothesize that a dynamically constructed history of interactions that is used to generate and select actions in an embodied agent can serve as the basis for ontogenetic development of the agent. The history of interactions, if self-organized, can provide abstraction as well as anticipation. Development in this case can be seen as *the increasing richness of the connections of experience with action*, mediated by suitable mechanisms. Such a history can facilitate incremental development at the borders of experience.

The development process also depends on drives and motivation. Classical conditioning and two-process reinforcement learning based on positive and negative reinforcers, e.g. (Rolls, 1999), are potential mechanisms for connecting previous experience with choice of action. For a review of computational approaches to classical conditioning, see (Balkenius and Morén, 1998). It is important however to provide reinforcement that is at the same time meaningful for the task at hand and general enough not to be merely task specific. I hypothesize that a combination of general environmental reinforcement, coupled with an interaction history that can suggest learning experiences "proximal" to currently mastered experience, can provide that kind of meaningful reinforcement.

Another approach to engendering drives and motivations in a developmental system is to encourage a search for "novelty" and "challenge" in the learning tasks. Novelty in a task can be determined by its predictability which can be measured by comparing the expected and actual outcome of an action, for example (Marshall and Meeden, 2004). A more sophisticated approach would be to take into account not how novel a task is, but instead how much can potentially be learnt from it. Oudeyer, Kaplan, Hafner and Whyte (2005) describe "Intelligent Adaptive Curiosity", an intrinsic motivation scheme based around "progress niches" that maximizes the potential to learn in tasks it chooses to undertake. The system is tested on an Aibo robot that has a number of objects within reach. It is observed that the robot progresses through stages of behaviour of increasing complexity. Starting with body-babbling type exploration, it moves to sensing visual changes in the environment, both external to and caused by its actions. It then moves onto trying various actions with specific toys, till finally, it uses actions appropriate to the affordances presented by the objects.

Kozima, Nakagawa and Yano (2005) study human (infant) social development using the robots "Keepon" and "Infanoid" both as tools for psychological investigation of humans and their interactions with the robots, and as platforms on which models of the developmental of social intelligence are tested. The intelligence is not designed, but is allowed to emerge through interaction by endowing the robots with basic capabilities and allowing open ontogenetic development of those capabilities. In (Kozima, 2002), Infanoid first acquires a kind of "intentionality" - that is, goal-directed spontaneous behaviour - and then uses joint-attention to identify with others and "understand" the communicative intentions of the behaviour of others.

Weng, Evans, Hwang and Lee (1999) explore a developmental learning algorithm, named "AA-Learning". They test it in experiments where faces are recognized and an appropriate greeting uttered (the "Robot Receptionist" experiment), and where a robot learns navigation by means of vocal commands, reinforced by pulling on a "rein" (the "Robot Horse" experiment). AA-Learning is a general learning algorithm that uses current sensor reading and "brain-state" to decide on the next state, with temporal extension achieved by a simple recursive sensor averaging. States are organized using a tree structure that encourages a hierarchical organization leading to the formation of state prototypes. States that do not occur frequently can also be removed ("forgetting"). Reinforcement, as well as supervised learning are used. The Interaction History Architecture, presented in this thesis, shares many similarities to this work, but fundamentally differs in that it directly compares temporally extended experiences.

2.4.4 Social Environment

That environment, embodiment and situatedness are important in the development of cognition is widely accepted (see for example Clancey, 1997; Pfeifer and Scheier, 1999; Maturana and Varela, 1987; Varela, Thompson and Rosch, 1991; Lindblom and Ziemke, 2003). However, the role of the social environment and social embeddedness for the development of both human-like cognition, and arguably, for many forms of higher animal cognition, is only recently [1] being supported by research in AI even though it has been established in both philosophical and developmental psychology for some time. Moreover, it is argued that the complex requirements of the social environment and social culture coupled with the necessity of placing oneself in the mind of others was a contributing factor to the drive toward primate and ultimately human intelligence (Machiavellian intelligence) (Byrne and Whiten, 1988). The theory (generally credited to Jolly (1966)) that primate intelligence originated to solve social intelligence and was only recently extended to be used outside the social domain, is referred to as the "social intelligence hypothesis". Recent neurological research (Rizzolatti, Fadiga, Gallese and Fogassi, 1996) shows that this is not merely a matter of learning and experience, but that there exist neural structures that have the purpose of

[1] A notable early exception is the work of W. Grey Walter, particularly the experiments with the robot 'Elsie', in the early '50s. For a review see (Holland, 2003).

"understanding" another's actions in the same terms, and using the same neural structures as the production of one's own actions.

While a multi-agent environment provides practical difficulties of unknown state (of other participants in the environment - non-Markovian environments), and so is vastly more complex than environments only occupied by static objects, social embedding requires knowledge of social rules, accepted practise and positions and roles within social hierarchies and systems.

In some of the earliest work in modern AI research to explicitly take into account social aspects of robotic interaction, Dautenhahn (1994), inspired by the social intelligence hypothesis, uses imitation as a social tool for robots to recognize each other and learn new movement skills through play. Dautenhahn (1999) identifies key aspects of social agents or robots as being embodied individuals in a social group that recognize each other, interact with each other, have histories (*i.e.* perceiving themselves in terms of their experiences), and communicate with each other through shared context. To illustrate, she also describes a robot-human "dancing" experiment studying the change in temporal coordination between human and robot. In (Billard and Dautenhahn, 1998) the roles of teacher and learner in a social learning environment are explored, with the "learner" robot learning to associate words with grounded experiences of hills and planes[2].

Robotics experiments with social interaction as key aspects are becoming more common. For an overview of socially intelligent agents see (Fong, Nourbakhsh and Dautenhahn, 2003), (Dautenhahn, Bond, Cañamero and Edmonds, 2002) and (Nehaniv and Dautenhahn, 2007). Examples of social agent systems designed to model social interactions are (Bond, 2002) and (Edmonds, 2002). An important aspect of robotics in mediating and, potentially, therapeutic roles can be seen within the Aurora project (Dautenhahn and Werry, 2001). This work is extended also with "Robota" (Billard and Hayes, 1998), a doll-like robot toy, and experiments using imitation and play (Billard, Robins, Dautenhahn and Nadel, 2006).

[2]This work draws direct inspiration from Vygotsky's theories of socio-cultural situatedness as a cornerstone of intelligence.

Kose-Bagci, Dautenhahn, Syrdal and Nehaniv (2007) use drumming as a mediation tool to study the interaction between an expressive humanoid robot and a human drumming partner.

Breazeal and Scassellati (2000) investigate the social interaction that occurs between a human-infant and caregiver with the robot Kismet taking the place of the infant. Aspects of the design of a robot to elicit and engage in expressive "emotion-inducing" interactive exchange are explored, with Kismet learning to regulate the interactions such that it is continually but not over stimulated. The regulation of interaction focusing on the interaction *kinesics* is also explored in (Robins, Dautenhahn, Nehaniv, Mirza, François and Olsson, 2005). See also the work of Kozima et al. (2005) in development of social interactive behaviour as discussed in the preceding section.

2.5 Characterizations of Behaviour and Experience

The approach taken in this thesis is to allow embodied agents to be able to develop in their action capabilities by considering and building upon the agent's own interaction with the environment. It is particularly important that it is not external representations and characterizations imposed by a human observer that are used to drive this ontogeny, but instead that the robot self-characterizes its own behaviour in order to generate action. Therefore this section briefly reviews other examples in the literature that describe robots able to characterize their own behaviour. For reference, this thesis considers the characterization of behaviour in terms of the changing informational relationships between a robot's sensors in Chapter 4 and by considering the comparison of experiences in Chapter 5. The characterizations are only relevant when applied to generating action as described in Chapter 7.

In robotics, dimension reduction and clustering techniques have been widely used although usually in the domains of pattern or object detection and localization. To do so to characterize the robot-environmental relationship is however less common. Notable exceptions are: the work of Oates, Schmill and Cohen (2000), who cluster sensorimotor experiences following action in order to predict outcomes of actions; Kaplan and Hafner (2005), who use information theoretic tools to characterize behaviour; and Poelz and Prem (2003), who use the Isomap non-linear dimension reduction method to ground symbols in sensorimotor data. These three groups of research and others are discussed in the following paragraphs.

Independently of the work of this thesis, (Oates et al., 2000) describe experiences as time-series of multi-variate sensor data, computing distance between time-series and clustering experiences to produce prototypes. Experiences are associated with the actions that initiated them, suggesting a robot could generalize about potential outcomes of its actions. Distances between experiences are calculated by using Dynamic Time Warping followed by measuring the area between the curves, and clusters formed by taking averages of time-warped experience curves. In contrast, information-theoretic metrics are used in this thesis to compare experiences. Furthermore, this thesis goes further by demonstrating how robots can direct their actions based on such experiences.

Kaplan and Hafner (2005) use information distances between sensors in an Aibo robot to compare simple behaviours of the robot. In that method, rather than reducing the dimension by summation within groups as I have done, they consider distances between different behaviours as distances between the full matrix of distances between all sensors. Long continuous examples of each behaviour (1000 timesteps) are used, and the whole sequence used rather than a moving window. The resulting distances between behaviours are shown as a projection onto a 2-dimensional map, and they find that similar behaviours group together. This research, which was carried out at a similar time to that reported in this thesis, supports the view that robot behaviour can be clustered using information relationships between sensor time-series. However, the research in this thesis goes further using an incremental formulation using a moving window creating trajectories through a low-dimensional information space. In addition, the use of the experience metric, combined with environmental reinforcement and its application to shaping action rather than behaviour, further distinguishes the work in this thesis.

In a series of papers Prem, Hoertnagl and Poelz (2003); Poelz and Prem (2003); Hoernagl, Poelz and Prem (2004) look at how symbols can be grounded in the sensory streams of a robot exploring its environment using a trajectory in a space analogous to ours. The approach is based on “Isomap”, a type of Multi-Dimensional Scaling capable of finding the intrinsic dimensionality of high dimensional data while preserving the non-linear structure. Isomap is able to find distances in non-linear manifolds by incrementally summing shorter geodesic paths. Sensor readings are divided into windows of different lengths containing events significant to the robot then trajectories are plotted in an Isomap reduced 3-dimensional space and compared with each other in terms of Euclidean (and other) distances between corresponding points of the trajectories. Prem et al. (2003) considers sensory systems in two groups corresponding to different sides of the robot to preserve lateral differentiation. In (Hoernagl et al., 2004) object recognition based on previous experience encoded into an Isomap representation is described, and this work echoes work conducted for this thesis, described in Chapter 5, on recognizing previous experience.

Although not characterizing robot behaviour in terms of its sensors, Nehmzow (2003) describes a method to quantitatively measure the behaviour of a mobile robot in terms of its trajectory in physical (Cartesian) space. He uses a method inspired by dynamical systems analysis called Error Growth Factor that shows whether trajectories diverge or converge from similar starting conditions.

2.6 Experience as the Basis For Action

In accordance with the embodied cognition perspective, this thesis does not stop at considering experiences and their relational properties, but instead proposes an architecture whereby action can be based on a continually developing history of experience (Chapter 7). Sensorimotor experiences are continually collected by the embodied agent, associated with action and environmental reinforcement, and related to each other by means of an information-theoretic metric measure (Chapter 5). Current experience is used to select a similar historical experience from which the next action is derived. This scheme is novel in certain key respects. Firstly it is the first to use a information-theoretic metric measure on sensorimotor experience for directing robot action, and moreover accomplishes this in real-time. The continual reconstruction and reorganization of the space of available experiences through forgetting and merging of experiences is also unique in this context. Finally, bringing this together in a human-robot interaction scenario using a variety of robotic platforms including a complex humanoid robot is an important contribution.

In this section significant architectures from the literature that also direct behaviour of robots using some form of history of interaction are reviewed.

2.6.1 Case-Based Reasoning

The concept of an agent learning from its past experience is one also used by the Case-Based Reasoning (CBR) approach (Kolodner, 1993). A descendant of Schank's dynamic memory (Schank and Abelson, 1977; Schank, 1999), CBR uses past experience represented as individual cases to make decisions about presented problems.

The basic process in CBR involves first examining a target problem and retrieving the best matching case from memory. These specific cases are then adapted to match the current situation and tested, with successful outcomes forming updating and adding to the cases in the history.

While CBR had great success in producing expert-system-like solutions to real-world problems, it was inherently ungrounded, representational and applicable only to problems that could be symbolically decomposed. Extension to the continuous domain (Ram and Santamaria, 1997), however, brings the approach much closer to the learning from interaction histories in robots as described in this thesis. Ram and Santamaria (1997) describe a real-time system that operates on the sensory data from a robot solving a navigation task. Their system is a hybrid of CBR and Reinforcement Learning approaches with control achieved through adaptation of schema-based reactive control. The level of control then is in adapting the parameters of a set of parallel reactive modules. The operation of the system cycles through *perceive*, *retrieve*, *adapt* and *learn* phases. In perceiving, the system reads sensor information and this is used in retrieving cases from memory by matching a recent sequence of sensor and control parameters with sequences represented as cases in the history. The matching is a simple squared distance between groups (*associations*) of time-series. The adapt phase then uses the parameters in the best matching case to update the current control parameters in a way that depends on reward received.

2.6.2 Reinforcement Learning with Memory

The work described in this thesis is also related to reinforcement learning (*e.g.* Sutton and Barto, 1998), particularly those examples that use intrinsic motivation (*e.g.* Barto and Şimşek, 2005; Bonarini, Lazaric, Restelli and Vitali, 2006) and memory-based approaches (*e.g.* Lin and Mitchell, 1992; Bakker, 2002; McCallum, 1996). In contrast to traditional reinforcement learning, the Interaction History Architecture approach uses temporally extended experience rather than the instantaneous values of the sensorimotor and internal variables (*state*). This distinction is important as, particularly where there is an interaction partner or other agents, the environment cannot be modelled as a simple Markov Decision Process. Q-Learning relies on this assumption, and is not guaranteed to find an optimal solution where state information is incomplete (Lin and Mitchell, 1992). This is also known as the Hidden State problem, and is generally addressed by including memory into the reinforcement model.

Lin and Mitchell (1992) describe Q-learning architectures utilizing feed-forward artificial neural networks to approximate the reinforcement learning Q function. They test three different architectures each introducing recent sensory and action history in different ways to the Q-Learning function approximating network. Testing on both a "cup picking-up" problem and a modified (more difficult) pole-balancing problem, where only the positions and not the velocities of the cart and pole are available, they find success for both recurrent architectures and a simple fixed-sized history window architecture. Success is dependent on memory depth and pay-off delay, the difficult of learning an effective policy increasing with both the memory depth and the length of the action sequence necessary before reinforcement payoff. I speculate that the length of history is an inherent problem with such architectures as it becomes increasingly difficult to access historical information the further back it is. However, with an architecture such as presented in this thesis, where the history is explicitly stored rather than being encoded in a function or network weights, the temporal distance to relevant experience is not an issue.

The problem of learning despite arbitrarily long time-lags is addressed by Bakker (2002) by using a recurrent neural network architecture, known as "Long Short-Term Memory" (LSTM). LSTM is designed for supervised time-series learning, to learn to infer the environmental state at any point, and provide this as input into a modified Q-learning system. The system is tested on a "road-sign" problem similar to the one used for testing the interaction history architecture in Section 7.3, and also on the hidden-state form of the pole-balancing problem referred to in the preceding paragraph. They find that the approach can handle longer-term dependencies than the architectures it was compared with: an Elman–style simple recurrent network, and a table-based system with memory.

These approaches require many cycles of presentation of a task to learn the solution to the problem, and this cannot be appropriate for developmental ontogeny in a robot as the cost of repeated failure can be high. McCallum (1996), however, describes "Nearest Sequence Memory", an "instance-based" state identification approach to the hidden state problem. Nearest neighbours to the current percept are found using two different methods: geometrically, or sequence match length, as appropriate for the representation. The geometric case measures the

Euclidean distance between the multi-dimensional instance data. Sequence match length finds, from the history, the longest sequence of exactly matching percepts to the current sequence of percepts. The nearest neighbours are then used in the Q-learning system.

Of the approaches discussed (Lin and Mitchell, 1992; Bakker, 2002; McCallum, 1996), the Nearest Sequence Memory utilizing geometric comparison of neighbours, seems closest to the approach taken in this thesis, and shows remarkable success, outperforming other approaches in terms of number of cycles, including the recurrent-Q of Lin and Mitchell (1992), by at least an order of magnitude. (McCallum, 1996) identifies the “raw experience” as important to this success, and this is echoed in the approach taken in this thesis. In addition he cites the distance metric as an area where the system could be improved, and I believe the experience metric is crucial in the success of the architecture presented in this thesis. The Interaction History Architecture approach (Chapter 7), then, does not require a Markovian environment and learns rapidly (typically within a few presentations). Furthermore, it does not require a static state space to be circumscribed at the outset, but instead uses a growing and changing space of experiences, where potentially in the course of ontogeny the set and character of sensors, actuators, and embodiment may change.

2.6.3 Artificial Neural Networks with History

Connectionist systems that have memory include, for instance Elman networks or other recurrent neural networks. Rylatt and Czarnecki (2000) showed that generally, recurrent neural networks are not well suited to learning delayed response tasks *i.e.* tasks where the appropriate action is dependent not only on current input but also on some previous input. Additionally, recurrent networks are very hard to design beyond a certain size and this requires that sensory input be encoded and reduced in quantity. Approaches such as Echo State Networks and Liquid State Machines attempt to address this limitation by training only the output nodes of a complex recurrent neural network (Jaeger and Haas, 2004).

Many models of associative memory and episodic memory have been proposed using artificial neural networks, which exhibit properties similar to human memory such as content-addressing, recall using partial cues and overlapping and interference of storage (Miikkulainen, 1992). A well known model of a content-addressable associative memory is the fully-connected, or Hopfield network. The connection weights, trained by Hebbian learning, determine the attractors of a dynamical system capable of storing patterns (Gurney, 1997). Modifications allow for hetero-associativity, and improved capacity. Another commonly used model is that of the topological feature map (Kohonen, 1984). Miikkulainen (1992) uses a hierarchical collection of modified feature-maps that show graceful degradation to model some features of a human episodic memory. See also (Vogel, 2005) describing interconnected regions of associative and heter-oassociative networks that model various types of memory and learning, including conditioned learning.

The Interaction History Architecture differs from these and many other similar approaches as no attempt is made to model the structure or process of human-like memory at neural level, instead emphasis is given to associating specific and general sensorimotor episodic experience directly with action control of an embodied artificial agent. Indeed, not having a neural structure avoids the problem of developing and growing neural structure as development proceeds. Instead, the cognitive structures that drive action and behaviour are built on the histories themselves. Additionally, in most artificial neural network approaches to the modeling of episodic memory, the memory of episodes appear only as weights and attractors of the system and so it is difficult to compare different memories either within the system or for an experimenter analysing the system.

A fundamental problem with many applications of connectionist systems is that of "designed ontology" (Clancey, 1997, p71), or a tendency for the network to operate as a function mapping while the real intelligence is in the human designed input vectors. I believe the approach of this thesis avoids this problem by operating from grounded sensory data only and outputting only embodied action. However, that is not to say that this cannot be achieved using a connectionist approach. An example where this problem is addressed is (Wermter and Elshaw, 2003) where action representation and semantic meaning of words is grounded in sensory data by applying a more biologically plausible structure in a distributed hierarchical collection of self-organizing memories.

The stance of this work is that the choice of whether to use a connectionist approach or one using some other computational scheme, is largely one of implementation and level of modelling. Instead the question of how history can drive embodied action in ontogeny, regardless of implementation, is considered to be more important in the focus of this thesis.

2.6.4 Other Architectures

Other architectures that take history of action and interaction (episodic memory) into account include top-down deliberative architectures, such as ACT-R (Anderson, 1996), which include memory storage and retrieval. Others such as Soar (a general, representational, cognitive architecture and programming environment (Rosenbloom, Laird and Newell, 1993)) have been extended to include episodic memory (Nuxoll and Laird, 2004). In Nuxoll and Laird's model the features of the episode are encoded and used in retrieval by matching. Encoding a representation for sensory input rather than using the raw data is common, except, notably in the continuous case-based reasoning model of Ram and Santamaria (1997), and in the architecture for interaction history presented in the present thesis. The extension of Soar to include episodic memory by Nuxoll and Laird has only been demonstrated on a grid world task, memories were perfect, non-modifiable and not deleted. Any extension of this model to robotic implementation would necessarily maintain the symbolic processing perspective that is characteristic of Soar.

Related work in the multi-agent domain (Arai, Sycara and Payne, 2000) has agents in a grid world acquiring coordination strategies, and uses a fixed-length episodic history expressly to counter the Markov Decision Process (MDP) assumption. However, that model is also state based and so uses a profit-sharing mechanism to assign credit to state-action pairs. Moreover, it does not compare episodes of history with previous ones, nor locate them in a metric space.

Other approaches include certain behaviour oriented control systems combined with learning (Matarić, 1992; Michaud and Matarić, 1998). Most behaviour-based models do not include learning from past experience, but of those that do the approach taken in this thesis differs in that the history is not specified in terms of the behaviour being selected (or indeed, the action being selected), but in terms of the sensorimotor history.

Ho et al. (2008), also describes an architecture for an agent in a virtual word that uses an episodic memory (that is symbolic but grounded in sensing and action), in a cognitive architecture for virtual agents. Ho's system is in fact an extension of behaviour-based subsumption architectures. The distinguishing feature of Ho's architecture though, is the communication and exchange of "autobiographic memories" as stories between agents, resulting in better adaptivity.

2.7 Interaction Histories in Other Fields

Interaction Histories appear in various other fields notably Human-Computer Interaction (HCI), intelligent virtual agents, student modelling and collaborative learning. In classical HCI, interaction histories, although not usually referred to as such, are used as a record of a user's interaction with an application or object for the purpose of providing the ability to revisit or replay previous actions or to provide an 'undo' facility. Intelligent software agents use histories of interaction to make suggestions, automatically complete tasks and improve future interactions (Rosson and Carroll, 2002, p332). Increasingly interaction history is considered as part of the design process itself, for example Jenifer Tidwell's design patterns for HCI systems design (Tidwell, 1999). The process of how people learn is the subject of "student modelling" and can benefit from taking into account the sequence of interactions of learners with a computer-based learning system thus increasing the likelihood that desirable learning events, specific to a particular learner, at a particular stage in their learning process, should occur (Akhras and Self, 2000).

Extending this concept, computer supported collaborative learning environments provide a framework (using interaction histories and sequences) for groups of people to learn and work on problems together (Dillenbourg, 1999) (du Boulay, 2000, p348). Such collaborative learning is thought to play a major role in constructive cognitive development (Vygotsky, 1978). A key element in such a system is a group memory of interaction between individuals, and it is suggested that interaction histories that take into account context play an important role in supporting learning (Siebra, Salgado and Tedesco, 2007). Perhaps the closest related field to interaction histories for robot ontogeny is that of virtual software agents. Virtual agents are often portrayed using a three-dimensional computer graphic representation and are increasingly being used in the interface between application and user, as well as to portray humans in virtual environments. A history of interaction with users here can be important in modelling the agent personality and making the agent believable (Romano and Wong, 2004; Tomlinson and Blumberg, 2003). A history of interaction also plays an important role in improving the "cognitive fit" between software and human users (Nehaniv, 1999*b*).

Chapter 3

Sensors and Measurement of Distance

3.1 Introduction

Any robot or embodied agent situated and acting in an environment will have many sensors through which the agent can receive data about itself and its environment. Some sense the external environment (*e.g.* visual sensors, infra-red distance sensors, sonar sensors), others sense the internal environment and body (*e.g.* motor position or proprioception sensors, internal temperature sensors, gyroscopic accelerometers) and others still sense internal variables (*e.g.* affective state). Some of these quantities are naturally discrete (*e.g.* buttons and switches). Generally though, the observed quantity is continuous and in current robotic systems the sensor maps the continuous values into discrete observations to some level of precision.

At any time, any of these sensory inputs can be modelled as a random variable, *i.e.* a variable with values taken from a given probability distribution. Note that these probability distributions may change over time, and with respect to each other. The changing distribution may indicate fundamental changes in either the agent (including the operation of the sensor), the environment, or in the agent-environment interaction. In this thesis, a central notion is that the changing nature of the sensory observations and their relationships to each other are material in describing the agent-environment interaction (see Hypothesis H1).

While the changing nature of individual sensors over time provides some level of measurement of the changing agent-environment interaction, it is clear that how these quantities vary with respect to each other and over time may provide a richer description of the changing interaction. In (te Boekhorst, Lungarella and Pfeifer, 2003), (Tarapore, Lungarella and Gómez, 2004) and (Tarapore, Lungarella and Gómez, 2006) statistical correlation, entropy and mutual information are used to segment and quantify (or fingerprint) the agent-environment interaction. Other papers, for example (Mirza, Nehaniv, Dautenhahn and te Boekhorst, 2005*a*, details of this work are also reported in Chapter 4) and (Kaplan and Hafner, 2005), explore information distance in behaviour categorization. Furthermore, these papers and many others including (Lungarella and Pfeifer, 2001), (Sporns and Pegors, 2004), support the notion that active sensory-motor coordination itself results in an increase in informational relationships between an embodied systems sensors and effectors, as well as in the control systems (*e.g.* nervous system and brain) of the agents. Behavioural characterization is explored in more detail in Chapter 4.

To make apparent the changing relation between sensory observations over time, a measure is required. Information distance was chosen as the primary measurement with which to compare sensors as it appears to capture both linear and non-linear relationships while also having the property of a true metric. Olsson, Nehaniv and Polani (2006*b*) compared the performance of the information distance with other measures on a sensory reconstruction task. The task required that sensory organization was recovered from the time-series data alone, and it was found that the information distance outperformed a range of measures including the correlation coefficient, Kullback-Leibler divergence, the 1-norm distance and other measures. In particular, where the sensors were of different modalities, only the information metric (used together with entropy maximizing adaptive binning, see Section 3.3.2) was successfully able to complete the task[1]. This is particularly important for our requirements as sensors will be compared from very different modalities.

The remainder of this chapter briefly overviews foundational mathematical concepts relevant to the metrics on sensorimotor experience presented in Chapters 4 and 5. Firstly, the information distance measure is presented, followed by a discussion of the essential concepts of how robotic sensors can be considered as random variables with time horizons, making such a measure applicable to them.

3.2 Information Distance

3.2.1 Information Theoretic Principles

Crutchfield (1990) takes the position that "information theory provides a quantitative and consistent framework with which to describe physical processes that admit only partial knowledge" and that information can be "a quantifier of behavioral[2] complexity". Seen from this perspective "information" may provide us with the tools with which to view physical processes such as the sensorimotor experience of robots and embodied agents.

In this section important results of information theory are presented, and then the Crutchfield-Rényi information metric is described, before this is related to sensor measurements in robots.

Shannon Information and Shannon Entropy

Consider[3] a random variable $\mathcal{X}$ taking values from the alphabet $\mathcal{A}_{\mathcal{X}} = \{a_1, a_2, \ldots, a_m\}$, with probability mass function $P(\mathcal{X})$.

[1] In this task the sensors were taken from a moving image, with the different modalities corresponding to different colours.

[2] Crutchfield was specifically referring to the *behaviour* of complex systems, but I believe that behaviour of embodied agents interacting with their environments can be considered in the same way as they are complex systems.

[3] The notation used generally follows (MacKay, 2003) except for the use of calligraphic letters to denote sensors modelled as random variables.

Shannon entropy is a measure of the uncertainty of a random variable. The Shannon entropy of $\mathcal{X}$ in the discrete domain is defined as

$$H(\mathcal{X}) \equiv \sum_{x \in \mathcal{A}_\mathcal{X}} P(x) \log_2 \frac{1}{P(x)} \tag{3.1}$$

with the convention that $0 \times \log \frac{1}{0} \equiv 0$ (MacKay, 2003). The units are *bits* for log base 2. The Shannon entropy is negatively related to the average information content that can be derived from observations on a random variable. A reduction in uncertainty (entropy) is an increase in information.

Given a second variable $\mathcal{Y}$ (which is not necessarily independent of $\mathcal{X}$), then two further quantities can be derived: the conditional entropy and the joint entropy. The *conditional entropy* is given by:

$$H(\mathcal{X}|\mathcal{Y}) = \sum_{(x,y) \in \mathcal{A}_\mathcal{X} \times \mathcal{A}_\mathcal{Y}} P(x,y) \log_2 \frac{1}{P(x|y)} \tag{3.2}$$

This measures the average remaining uncertainty about $\mathcal{X}$ when it is known that $\mathcal{Y} = y$. The probabilities $P(x,y)$ and $P(x|y)$ are the joint probability of outcomes $\mathcal{X} = x$ and $\mathcal{Y} = y$, and the conditional probability of outcome $\mathcal{X} = x$ given $\mathcal{Y} = y$, respectively. The *joint entropy* is given by:

$$H(\mathcal{X},\mathcal{Y}) = \sum_{(x,y) \in \mathcal{A}_\mathcal{X} \times \mathcal{A}_\mathcal{Y}} P(x,y) \log_2 \frac{1}{P(x,y)} \tag{3.3}$$

Note that entropy is additive only for *independent* random variables:

$$H(\mathcal{X},\mathcal{Y}) = H(\mathcal{X}) + H(\mathcal{Y}) \quad \text{iff} \quad P(x,y) = P(x)P(y) \, \forall \, x \in \mathcal{A}_\mathcal{X} \wedge y \in \mathcal{A}_\mathcal{Y} \tag{3.4}$$

Where $\mathcal{X}$ and $\mathcal{Y}$ are *dependent* random variables, then the joint entropy will be less than the sum of the individual entropies.

The relationship between the individual entropies, their joint entropy and their conditional entropies is shown conceptually in Figure 3.1. Note that when the circles representing the entropies do not overlap, the variables are independent,

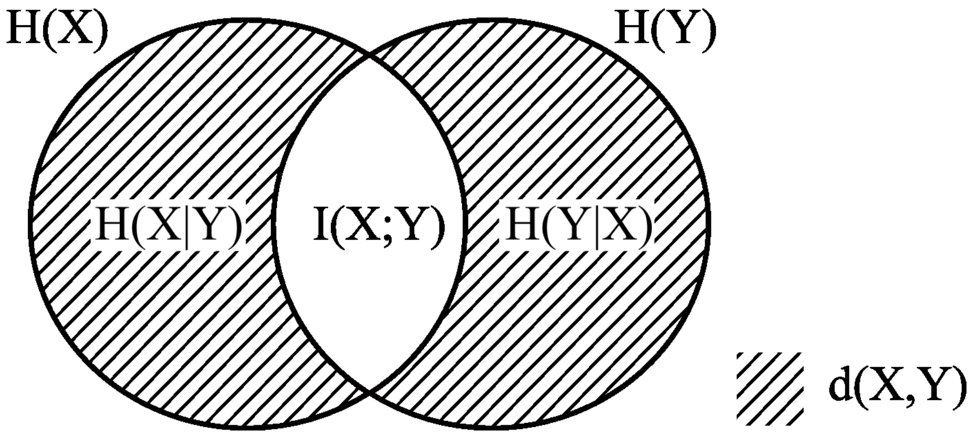


Figure 3.1: Conceptual diagram showing relationships between entropies of two random variables and their joint and conditional entropies. Also indicated are the information distance $d(\mathcal{X}, \mathcal{Y})$ and the mutual information $I(\mathcal{X}; \mathcal{Y})$.

and when the circles overlap exactly, the variables are *recoding equivalents*, *i.e.* knowledge of the value of one variable completely determines the value of the other.

3.2.2 Crutchfield-Rényi Information Metric

The Crutchfield-Rényi *information metric* (Crutchfield, 1990), in this thesis also referred to as the *information distance*, is given by:

$$d(\mathcal{X}, \mathcal{Y}) = H(\mathcal{X}|\mathcal{Y}) + H(\mathcal{Y}|\mathcal{X}) \tag{3.5}$$

Crutchfield (1990) shows that this satisfies the mathematical axioms of equivalence, symmetry and the triangle inequality and so is a metric. Specifically, for three random variables $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$, d is a *metric* if it satisfies the following:

1. $d(\mathcal{X}, \mathcal{Y}) = 0 \quad \text{iff} \quad \mathcal{X}$ and $\mathcal{Y}$ are equivalent (*i.e.* recoding equivalents)
2. $d(\mathcal{X}, \mathcal{Y}) = d(\mathcal{Y}, \mathcal{X})$ (symmetry)
3. $d(\mathcal{X}, \mathcal{Y}) + d(\mathcal{Y}, \mathcal{Z}) \geq d(\mathcal{X}, \mathcal{Z})$ (triangle inequality).

Thus d defines a geometric structure on any space of jointly distributed information sources. The information metric is also shown conceptually in Figure 3.1. Note that $d(\mathcal{X}, \mathcal{Y})$ has a minimum value of 0 when the variables are recoding equivalents, and a maximum value of $H(\mathcal{X}) + H(\mathcal{Y})$ when $\mathcal{X}$ and $\mathcal{Y}$ are independent.

Note that while the information metric is a true metric in the space of information sources, *mutual information* $I(\mathcal{X};\mathcal{Y}) = H(\mathcal{X}) - H(\mathcal{X}|\mathcal{Y})$, and the *relative entropy* or *Kullback-Liebler divergence* $D_{KL}(P \parallel Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$, are not metrics, failing on equivalence and symmetry respectively.

3.3 Robot Sensors as Information Sources

In the preceding review of the information-theoretic background, the random variables (or information sources), can be discrete or continuous, and come from any source physical or otherwise. In this thesis I am concerned with sensors from robots or other embodied agents. In general any sensor samples features of the internal or external environment at regular intervals, and can be modelled as a random variable $\mathcal{X}$ changing with time, taking values $x(t) \in \mathcal{A}_{\mathcal{X}}$, from a probability distribution $\mathcal{P}_{\mathcal{X}}$. $\mathcal{A}_{\mathcal{X}} = \{x_1, \ldots, x_m\}$ is the set of m possible values of $\mathcal{X}$, and time is taken to be discrete (i.e. t will denote a natural number).

In practise, different sensors will have different sample rates (and these may or may not be regular) and different sample resolutions. In the experiments conducted for this thesis though, a further sampling of all sensors at a universal rate and resolution, is operated, while also normalizing the result to common values for all sensors. This is achieved by the processes of *binning* and *normalization.*

3.3.1 Normalization of Sensor Values

A sensor S with values in the range $[S_{min}, S_{max}]$, can be normalized by re-mapping all samples S_t into the range $[0, 1]$ as follows:

$$norm(S_t) = \frac{S_t - S_{min}}{S_{max} - S_{min}} \tag{3.6}$$

3.3.2 Binning of Sensors

A sensor S, modelled as a discrete random variable, that takes values from an alphabet $\mathcal{A}_S$ can be re-mapped into a new sensor S' that takes values from a new alphabet $\mathcal{B}_S = \{0, 1, \ldots, (N-1)\}$, where $N > 0$ is the number of "bins", as follows:

$$S'_t = ceil(norm(S_t) * N) - 1 \tag{3.7}$$

where the $ceil(x)$ is a function that returns the smallest integer not less than x.

There are, of course, alternative schemes for binning sensor data. An important one is "adaptive binning using entropy maximization" as used by Olsson et al. (2006*b*). This scheme assigns bin boundaries during quantization so as to maximize the entropy across bins. While this was implemented during the research for this thesis, it was not used in the results presented as, for the experience metric, comparison is only made between time-series from the same sensor taken at different times, not cross-modally. Additionally, it is not clear how having different bin boundaries used to compare different experiences within a single space will affect its metric properties, and this is flagged as a subject for further research. For details about the experience metric see Chapter 5.

3.3.3 Sensorimotor Variables with Horizon

Any sensor or motor variable X, beginning from a particular moment in time t_0 until a later moment $t_0 + h$ $(h > 0)$, with the sequence of values

$$\mathcal{X}_{t_0,h} = x(t_0), x(t_0+1), \ldots, x(t_0+h-1) \tag{3.8}$$

can be considered as the time-series data from a new random variable $\mathcal{X}_{t_0,h}$, the *sensorimotor variable with temporal horizon* h *starting at time* t_0. Note that it is likely that a single robot "sensor" can be considered as many time-shifted sensors of a given horizon length and use the notation $S^n_{t,h}$ to refer to *sensor* n starting at time t with horizon h, and $\mathcal{S}^n_{t,h}$ the corresponding *random variable*.

3.3.4 Estimating the Probability Distribution Function

Consider a random variable $\mathcal{X}_{t,h}$. If the time-series is stationary (*i.e.* its mean, and statistical variance are constant over the length of the series), then its first-order probability distribution $p(x)$ can be directly estimated from its *frequency histogram* counting the occurrence of sensor values in each of a number of bins partitioning the data space. Likewise, a 2-dimensional frequency histogram can be used to estimate the second-order probability distribution $p(x, y)$ of two variables $\mathcal{X}$ and $\mathcal{Y}$ where the samples can be "lined-up".

This method is a naive way of estimating the probability distribution and is subject to certain systematic errors due to the number of samples used in the estimate, the rate of sampling, the choice of quantization level (number of bins) and the deviation from the underlying stationarity of the data. These issues and potential solutions, including using kernel density estimators, are considered by Lungarella et al. (2005). In (Mirza, Nehaniv, Dautenhahn and te Boekhorst, 2005*b*), the effect of different quantization levels is examined in relation to the estimation of the entropy of *regions* of a long wavelength sine-curve. It is found that quantization using a very small number of bins causes artificial peaks in the entropy which are smoothed out towards the idealized curve when using higher numbers of bins. However, as the number of bins is increased, so is the computational requirement and so a compromise has to be reached. An important result though is that such effects appear to be less marked for "real" data than for the smoothly varying sine-curve examined, and so real data may not need such a high quantization level for adequate estimation.

In the work conducted within this thesis, in the interest of keeping the computational complexity to a minimum and achieving performance in real-time, the general strategy is to use a short as possible time-series but sufficiently long enough to be able to assume local stationarity. The number of bins is kept to a minimum mainly to offset the choice of a short horizon (to avoid sparse population of bins), with the assumption that real data will reduce the artificial peaks in the entropy caused by smaller numbers of bins.

3.4 Chapter Summary

This chapter introduced the Crutchfield-Rényi information metric and related it to the information-theoretic comparison of sensors cast as information sources. The concept of sensorimotor variables with horizon was introduced permitting defined periods of sensor time-series to also be considered as information sources. Furthermore, issues of estimation of the probability distribution functions of these information sources from discrete sensor readings were discussed.

In later chapters these concepts will be used to permit comparison of groups of sensors both among each other and over time realizing measures such as the experience metric (See Chapter 5), eventually using these methods to create an interaction history for an artificial agent that can be used to direct future action based on past experience.

Chapter 4

Self-recognition of Grounded Sensorimotor Experience

4.1 Introduction

A first step towards a grounded sensorimotor interaction history for ontogeny in robots is to establish techniques that can be used for meaningful recognition of the robot experience from the robot's own perspective (*self-recognition*), as the robot interacts with its environment . This is necessary as we assume there are no externally imposed symbols or categories pregiven in ontogeny. Category formation and recognition must therefore be grounded in relationships between what is sensed by different sensors at different times.

Thus, the goal in this chapter is to explore and validate the information metric as a tool that an embodied agent (robot) can use to identify and categorize behaviour from the agent's perspective. Ideally, categorization should be grounded entirely in the sensorimotor data by which the robot is coupled with the environment without explicit externally imposed symbols or characterization.

A simple method is developed in this chapter (first reported in Mirza, Nehaniv, Dautenhahn and te Boekhorst, 2005*a*; Mirza, Nehaniv, te Boekhorst and Dautenhahn, 2005) whereby the sensory input of the robot is split into two groups (nominally "sensor" and "motor") and the average information distance between the sensors of these groups is plotted as a trajectory on a graph. The trajectory is then examined for its ability to organize the behaviour when compared to an external observer's own categorization of the same behaviour. Simple metrics describing the trajectory structure are used for objective measurement of the different traces that different behaviours produce. Finally, it is shown that such a mechanism can be used by a robot to identify its own behavioural interactions.

This chapter addresses Hypothesis 1 (regarding behaviour categorization) and Hypothesis 2 (regarding self-identification of behaviour). It is important to recognize that the basic underlying proposal that sensory data over a period of a time will be different for a robot executing different behaviours, is in itself trivial, but that recognizing, categorizing and identifying behaviour is not. To illustrate, consider a robot receiving sensory data from 10 sensors at a rate of 10Hz for 1 second. This represents a 100 dimensional data set, and if each sensor was an integer ranging from 1 to 4, then there are a staggering 1.6×10^{60} possible permutations of the data set, each potentially describing a different behaviour. Therefore, effective methods to reduce the complexity of the data set while retaining its character are required.

4.2 Sensor-Motor Average Information Distance (AID) Plots

4.2.1 Definition of Average Information Distance (AID)

The *Average Information Distance* (AID), $\langle d_{\mathbf{X}} \rangle_{t,h}$, of a collection of n random variables $\mathbf{X}_{t,h} = (\mathcal{X}^1_{t,h}, \mathcal{X}^2_{t,h}, ..., \mathcal{X}^n_{t,h})$ at time t with horizon h is defined as:

$$\begin{aligned} \langle d_{\mathbf{X}} \rangle_{t,h} &= \frac{1}{n^2} \sum_{a=1}^{n} \sum_{b=1}^{n} d(\mathcal{X}^a_{t,h}, \mathcal{X}^b_{t,h}) \\ &= \frac{1}{2n(n-1)} \sum_{1 \le a < b \le n} d(\mathcal{X}^a_{t,h}, \mathcal{X}^b_{t,h}) \end{aligned} \tag{4.1}$$

where the probability density functions are estimated for a window of h time-steps using Q bins, and the computational simplification follows from the metric symmetry of the information distance. As with information distance, the AID is measured in *bits*.

Observe that low values of the AID indicate a small information distance on average between all variables and imply a high degree of correlation between them. A situation where the AID would be expected to be zero would be when the variables were unchanging. The highest value of AID would occur between groups of completely uncorrelated random variables. Also note that the estimates of the information distance, and therefore the AID, are dependent on the chosen values of both the horizon length h and the number of bins Q used to estimate the probability densities. Section 3.3.4 discusses estimating probability densities for time-series and choosing suitable values of h and Q.

4.2.2 Groups of sensors

By grouping sensors and calculating the AID for each group, it becomes possible to describe the changing informational relationships between groups using a very small number of variables. Note however, that in taking the average of the possible

Table 4.1: AIBO Telemetry Collected.

Sensors	#	Motors	#
IR-Distance	1	Leg Joint Positions	12
Accelerometers	3	Head Joint Positions	4
Temperature/Battery	2	Tail Joint Positions	2
Buttons	8	Motor Force / Duties	18
Visual	27[a]		
Total Sensors	41	Total Motors	36

[a] R,G,B in a 3×3 grid, see Section 4.3.1

information distances between sensors, it is no longer possible to say which sensors contribute to the changes. However, this has a potential advantage in that it frees us, to some extent, from the details, and allows general relationships to emerge. Specifically, grouping in this way allows consideration of patterns of activity in one sensor as being equivalent to the same pattern of activity in any other sensor in the group.

While any number of groups of sensors could be used, in the experiments that follow, just two groups are considered, one that intuitively characterizes the "agent" and another that characterizes the "environment". Thus, all environmental sensoric inputs constitute one group, **S**, and all motor and internal variables constitute another group, **M** (see Table 4.1 and Section 4.3.1).

The AID for each group of sensors can be calculated and plotted in two dimensions to realize a representation of the relation between sensors and motors. Doing this for successive time-steps for a fixed-size moving window, results in a representation of how the sensor-motor relationship is changing with time. I call this plot a *Sensor-Motor* AID *vs.* AID *Plot*[1].

[1] In (Mirza, Nehaniv, Dautenhahn and te Boekhorst, 2005*a*) the term "Phase-plot" was used, however, in dynamical systems theory, that term specifically refers to the plot of a variable and its derivative, and so here, the term "AID *vs.* AID Plot" is used instead.

Table 4.2: Sensorimotor Behaviour Categorization: Key to Experimental Investigations.

Experiment		Section
1	Investigate the effect of the horizon length, h and number of bins, Q parameters.	4.3.4
2	Study the sensor-motor AID plots of some simple behaviours.	4.4
3	Consider potential metrics of the AID plots suitable for identifying simple behaviours among a series of behaviours.	4.5
4	Compare information distance to other distance measures in this context.	4.3.3

4.3 Experimental Investigations

Hypotheses 1 and 2 were tested in a series of experiments (see Table 4.2) using the robotic setup described in Section 4.3.1. Initially, the robot executed a number of simple behaviours in isolation, and for each of these, an AID plot produced. The plots were compared in terms of certain descriptive metrics. In the next experiment, the same robot executed some of the same simple behaviours, but this time as part of a continuous autonomous overall "wandering" behaviour. The previously discovered categories were then used to characterize (identify) each segment of that behaviour.

To begin with, I describe some preliminary investigations into the effect of horizon length, h and number of bins, Q on the AID plots, and also briefly compare some other distance measures with the AID in terms of the resulting graphs.

4.3.1 Experimental Setup - AIBO ERS220

Experiments were conducted on a real robot to avoid artifacts of simulation and to provide rich sensory-motor data. The commercial robot Sony AIBO[2] ERS220

[2] AIBO is a registered trademark of Sony Corporation. At the time of writing, the production of the AIBO series of robots is discontinued.

Figure 4.1: Sony AIBO™ ERS210 used in the experiments.

was used- see Figure 4.1. Behaviours were written using the Open Source software framework Tekkotsu (Touretzky and Tira-Thompson, 2004) and executed on the AIBO. Sensor/motor data was transmitted at regular intervals (on average 10 frames/sec.) to a workstation over wireless LAN where the data was processed in real-time. For experimental purposes, data was also reprocessed off-line with different parameter values.

Experiments were carried out in a low walled (50cm high) 2m×2m arena (dark wood walls, dark green speckled floor, over-head lights, cream coloured walls of lab beyond the walls), either empty or containing 2 white "A4-printer paper" boxes, and/or a pink ball.

Table 4.1 summarizes the variables available to the AIBO from which data was collected. The data was grouped into 36 motor variables and 14 sensor variables. Further sensor variables were constructed from the AIBO's camera located in its head which produced 3-component colour images, 88×72 pixels in size, received at regular intervals. These were partitioned into an $N \times N$ grid, and for each region, the pixel values for each colour (red, green and blue) were averaged, and the resulting numbers taken as the values of three pseudo-sensors for each region at that time. For the majority of experiments presented, 27 visual sensors were used, constructed by taking the red, green and blue pixel averages of regions in a 3×3 grid over the image. Tests were also conducted with 108 vision sensors (36 each of R,G,B in a 6×6 grid), and 36 sensors (red, or "Effective-Red" [3] only).

4.3.2 Preliminary Investigation

A two minute time-series of data (1195 timesteps, 1 timestep$\simeq 100ms$) was taken while the robot moved around the arena (in this case also containing white boxes as obstacles) interacting with a pink ball. This interaction involved random wandering behaviour coupled with object detection targeted to the pink ball. When the ball was seen, the AIBO moved towards the ball. If necessary, the ball was then moved to another location in the arena by the investigator. This data was used in the preliminary investigations of this section only.

AID *vs.* Time Plot

The graph of Figure 4.2 shows the AID (in *bits*) for each sensor group against timestep, as calculated for a horizon $h = 100$ timesteps and number of bins $Q = 8$. In the sequence shown, the robot approaches the ball twice and each time the ball is moved to a new location in the arena (at timesteps ~ 550 and ~ 850).

The AID seems to capture certain aspects of the general behaviour. The motor group AID increases to around $2.4 bits$ when the robot is in motion decreasing to around $1.5 bits$ when it stops in front of the ball. Note it does not reach zero as the stationary state does not extend over the whole horizon. The sensor group AID shows peaks of high and low corresponding to variability in the environment. The lowest values of $0.7 bits$ at timesteps ~ 200 and ~ 1000 correspond to times where the visual sensors show very little change, for example when the robot sees just the wall or the floor for some time.

[3] "Effective-Red" calculated as $R - \frac{G+B}{2}$, *i.e.* the amount of red compensating for the effect of green and blue on perception of red (Tarapore et al., 2004) (Varela et al., 1991, ch. 8)

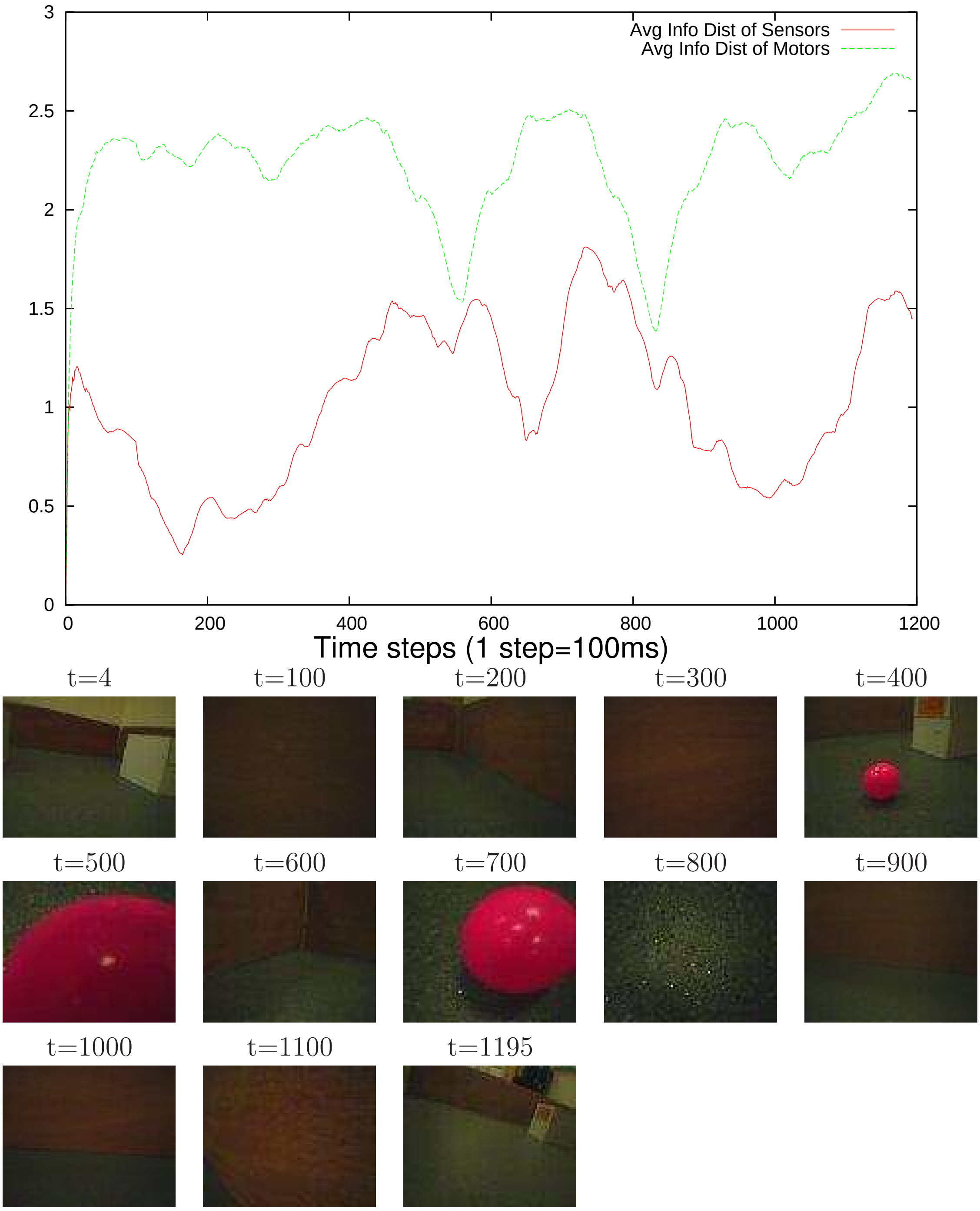


Figure 4.2: (Top) *AID for sensor and motor groups against time.* The robot acts for 2 minutes (1195 timesteps). AID is calculated using probability distributions estimated from a 100 timestep moving window ($h = 100$) using $Q = 8$ bins. Vertical axis is AID in *bits/sensor*. (Bottom) Images from AIBO head camera from selected timesteps.

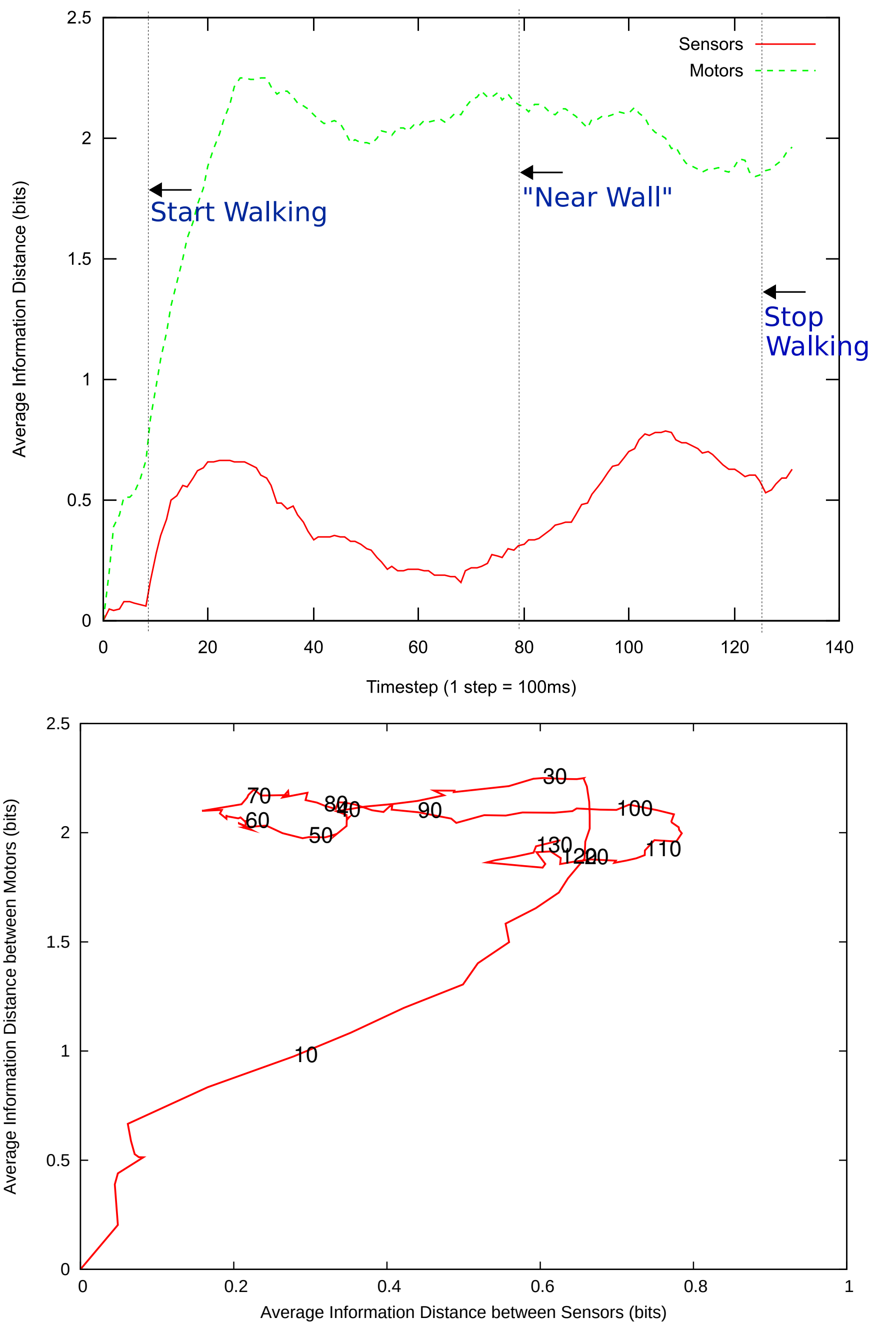


Figure 4.3: *Average Information Distance (AID) against time and sensor-motor* AID *vs.* AID *plot.* The data is from an AIBO walking from one end of the arena to the other. (Top) Sensors and Motors AID vs. time. Annotations mark *Start of walk*, *Near Wall* - which means that wall is in IR sensor range (i.e. 900mm) and *Stop Walking*. This illustrates delay in AID responding to change (*delay* $\simeq h$) (Bottom) Sensor-motor AID *vs.* AID plot. The plot starts at the origin, time steps are marked along path. Data has 130 time-steps, horizon $h = 20$ and $Q = 12$ bins.

AID *vs.* AID plot

If the AID for one group of sensors is plotted against the AID for another group of sensors, the resulting trajectory helps visualize their interaction. Such a plot for the robot walking from one end of the arena towards a wall is shown in Figure 4.3 along with a plot of the average information distance of sensors and motors against time. This illustrates the utility of the AID *vs.* AID method as it makes the changing relations between the two groups of sensors instantly clear. In this short sequence the trajectory of the plot shows interesting structure in terms of the position and vertical/horizontal extent of the plot as well as in the points and frequency of the crossing points. Later in this chapter metrics that can quantify such features are explored.

Note that there is a delay in the trajectory responding to changes, as illustrated by annotations on the figure. This is a result of estimating the probabilities over a window, and the effect increases as the horizon length is increased. See also Section 4.3.4 for an investigation into the effect of changing the horizon length. The effect is particularly noticeable in the start-up delay where the plot traces a line from the origin till approximately timestep 20 (the horizon length), and any measurement of trajectory structure should ignore this start-up anomaly. This effect needs to be considered when separating one behaviour from another.

Table 4.3: Alternative Measures on Sensory Groupings

Distance Measurement	Description
1 Simple average	Time average over binned sensor values: $$\frac{1}{hn}\sum_{t=t_0-(h-1)}^{t_0}\sum_{a=1}^{n}X_t^a$$
2 1-norm distance (Hamming distance)	Average of absolute numerical difference between binned value of pairs of sensors: $$\frac{1}{2n(n-1)h}\sum_{t=t_0-(h-1)}^{t_0}\sum_{1\leq a<b\leq n}\lvert X_t^a - X_t^b\rvert$$
3 Pairwise average of Pearson's Squared Correlation Distance	$$\frac{1}{2n(n-1)}\sum_{1\leq a<b\leq n} d_{pearson}(\mathcal{X}_{t_0,h}^a, \mathcal{X}_{t_0,h}^b)$$ where $d_{pearson} = 1 - r^2$ and r is Pearson's Correlation Coefficient (see Coolican, 1994)
Note: X_t^a is the value of sensor a at time t. Calculations given for time t_0 and must be repeated for consecutive timesteps to obtain the graphs in Figure 4.4.	

4.3.3 Comparison of AID to Other Measures

The data gathered in the first preliminary investigation was reanalysed using different distance measures (See Table 4.3) for comparison with the AID. Figure 4.4 shows the graphs produced as a result. Inspection of the relative variation in the sensor and motor traces in these graphs suggests that the information distance (A) may reveal more detail about the relationship between the sensors and motors than either the simple average (B) or the 1-norm distance (C). The statistical correlation (D) appears to be different, showing more peaks and troughs in the sensor graph than for information distance, but less so for the motor graph. The graphs suggest that the average information distance measure both finds interest-

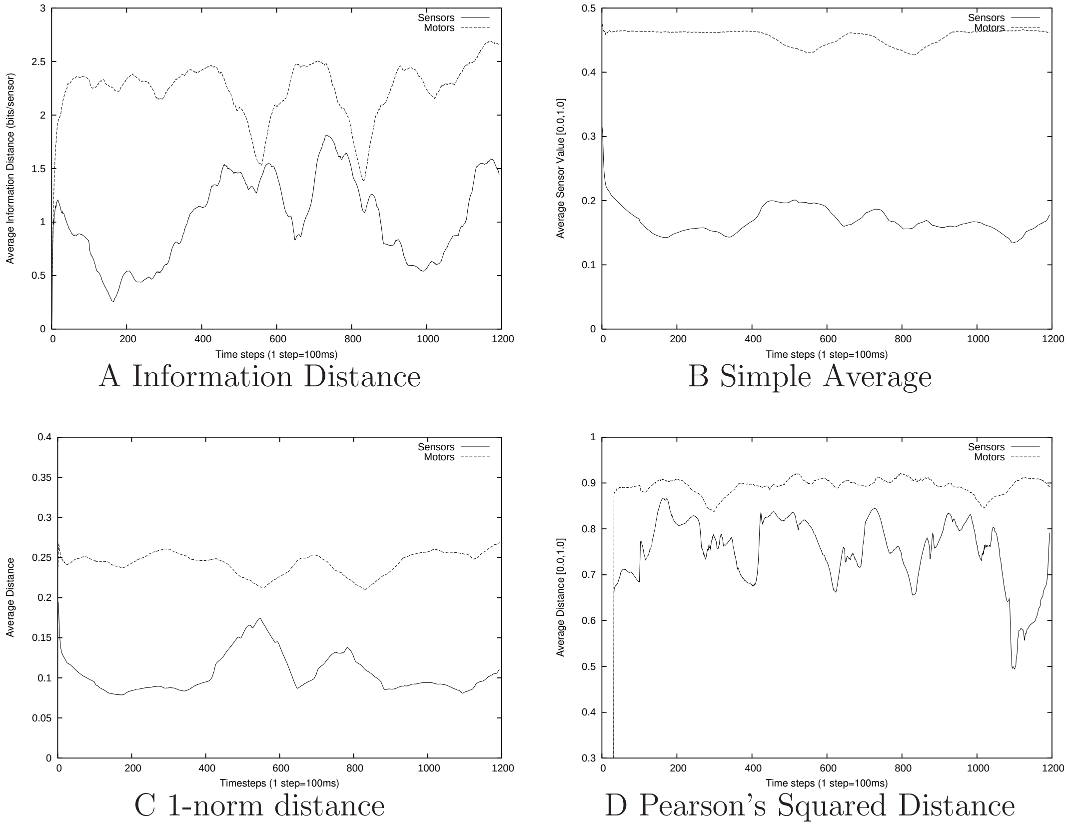


Figure 4.4: *Comparison of alternative AID measures.* The Information Distance distance measurement *(A)* is compared to the Simple Average *(B)*, 1-norm distance *(C)* and the Pearson's Squared Correlation Distance *(D)*. Calculations made for a horizon $h = 100$ over a time-series with 1195 timesteps. Entropy estimates for information distance used $Q = 8$ bins.

ing features of the sensor-motor relationship as well as finding detail not revealed by simpler measures, however this is likely to be highly task dependent and other statistical measures may be equally useful. The usefulness of information distance over other measures is supported in the results of (Olsson et al., 2006*b*) where the "sensory reconstruction method" is used as a test problem, and suggests that the information distance captures general relationships between sensors, not just linear relationships.

4.3.4 Investigation of effect of horizon length h and number of bins Q on the AID

Experiment

The choice of values for the horizon h (the moving window across which the probability density functions, and therefore the information distance, were estimated), and for the number of bins Q (which sets the resolution of the probability density functions), can be expected to affect the resulting plots. To investigate the effect, a time-series was chosen consisting of 450 data points taken from an experimental run where the robot was "exploring" an empty arena. This "wandering" behaviour consisted of walking forwards until an object (wall) was detected and then turning a random amount before repeating the behaviour.

Results and Discussion

The AID was calculated over a moving time-window for many different values of h and Q. A selection of the results are shown in Figure 4.5, showing AID *vs.* AID plots for varying values of h for a fixed Q and in Figure 4.6 where Q is varied instead. The resulting trajectories (disregarding the start-up sequence) are generally cyclic, and occupy a definite area within the AID *vs.* AID space.

The results show an overall similarity, but definite progression in structure of the trajectory. Figure 4.5 shows similarity in structure between the graphs for $h = 20$ and $h = 40$ as well as a similarity between $h = 60$, $h = 80$ and $h = 100$. The graph for $h = 10$ is also somewhat similar to $h = 20$ but is lower in the motor dimension in addition to being more "chaotic" in structure. However, there appears to be a smooth progression in structure that sees the chaotic nature of the path reducing as h increases, as well as information distance on average increasing with increasing h (to a maximum, in this case, of about 2.5 in the motor dimension and 1.0 in the sensor dimension). Increasing h also appears to reduce the detail and amount of "motion" in the trajectory, as well as the area occupied, as more of the preceding time-series is considered for every point of the

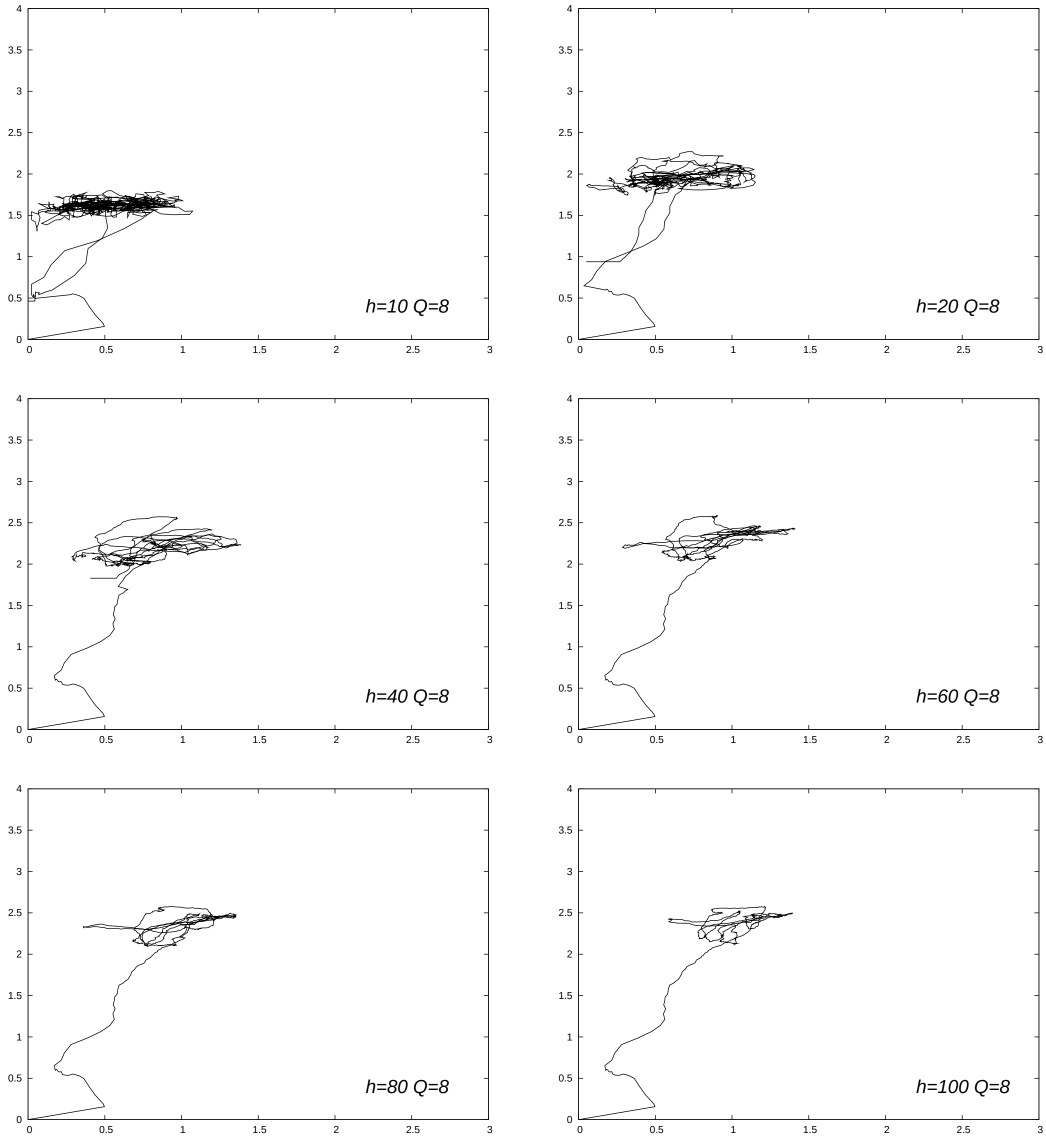


Figure 4.5: *Effect of horizon length h on Sensor-Motor* AID *vs.* AID *plots.* Figure shows a sample of results for window-size $h \in \{10, 20, 40, 60, 80, 100\}$ with fixed Q=8. In all cases horizontal and vertical axes are the moving window AID for sensors and motors respectively.

AID plot. At the limit h=*length of time-series*, this would result in a single value for the whole time-series.

In Figure 4.6 a similar story can be seen for the effect of increasing the variable Q. Apart from the smallest value of Q, the overall structure of the traces are similar, with the information distances increasing on average as Q is increased. Increasing Q seems to increase the overall size of the trajectories (in terms of

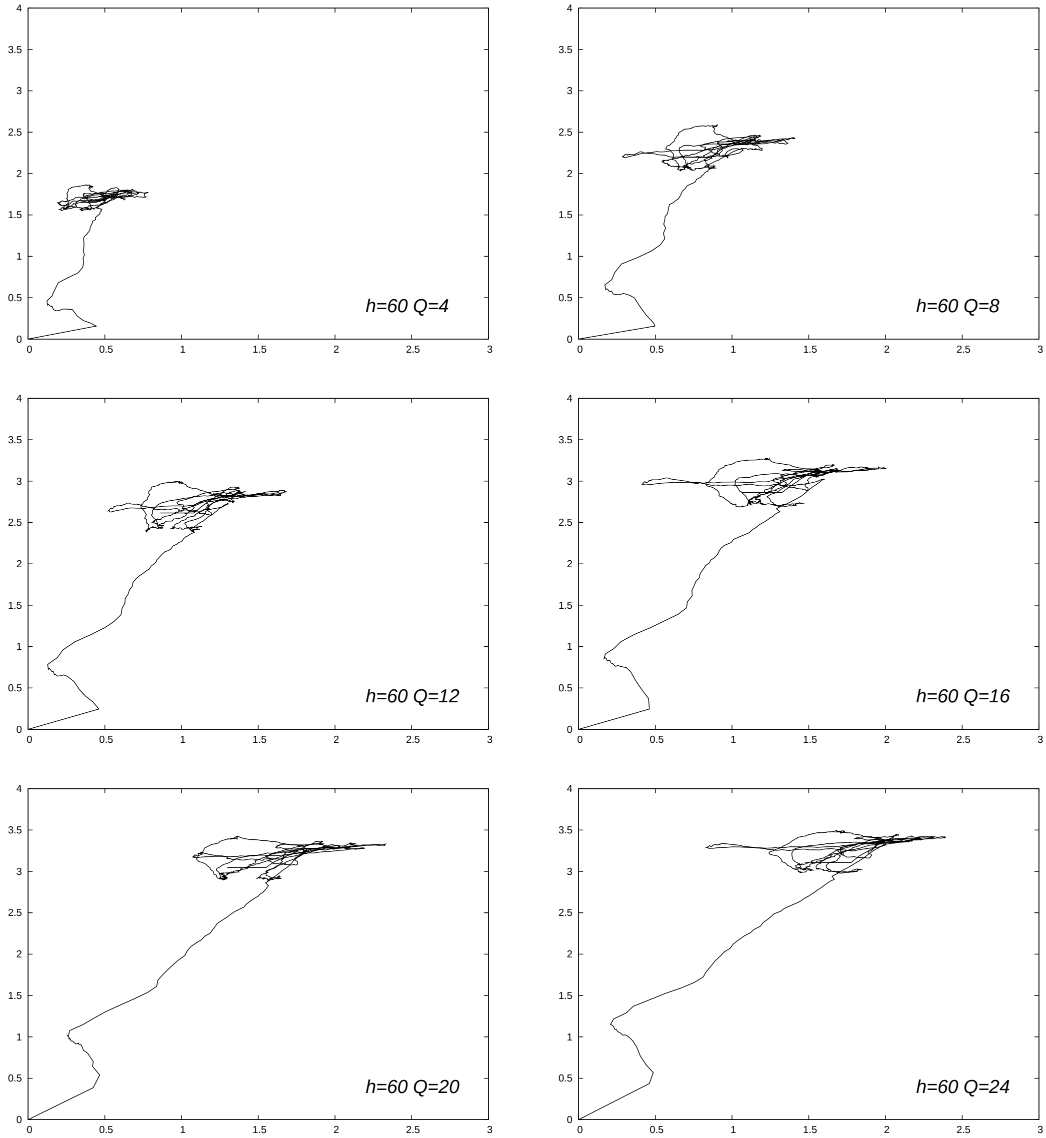


Figure 4.6: *Effect of number of bins* Q *on Sensor-Motor* AID *vs.* AID *plots.* Figure shows a sample of results for number of bins $Q \in \{4, 8, 12, 16, 20, 24\}$ with fixed h=60. In all cases horizontal and vertical axes are the moving window AID for sensors and motors respectively.

the number of bits separating the extremes of the trajectory), as well as moving the trajectory up and to the right (increasing AID for both groups). This would be expected as a finer grained estimation of the probability density would find more differences in the data. However, depending on the horizon h, and therefore the number of data points used in the estimate, a very large number of bins would be sparsely populated resulting in inaccurate estimates of the probabilities.

Thus there is a limit to the amount of detail to be found in the plots. Again, increasing the variable beyond a certain amount no longer results in an increase in information distance suggesting that there is only a certain amount of information that can be discerned from a given set of data.

Conclusion

The progressions identified as a result of varying the variables h and Q suggest that (1) if the variable is altered a relatively small amount, then no significant change can be expected in the AID *vs.* AID trace in terms of structure of trace or position in the AID *vs.* AID space (*i.e.* the method is fairly robust with respect to relatively small changes in h and Q), (2) that there is a limit to the amount of information that can be discerned between sensorimotor streams by increasing the quantization variables h and Q and (3) that very small values of either h or Q result in a distorted view of the informational relationships in the data.

Suggestions for the Choice of h and Q

These results suggest that a minimum of $h = 20$ and $Q = 8$ should be considered for quantization of robot data such as this, with values $h = 60$ and $Q = 16$ being closer to optimal choices. However, in choosing suitable values for quantization in the experiments conducted in this chapter, two further factors were considered. The first was computational tractability, *i.e.* although this analysis was carried out off-line, the method should be suitable for a robot to carry out self-characterization of behaviour and thus should be operable in "real-time". Higher values of either h, Q or both would result in increased computation time. The second factor was the length of behaviour that would be covered in the timeframe of the horizon length chosen. Clearly if the horizon covered many different behaviours, then the plots could potentially give mixed results. Shorter horizons were therefore desirable to ensure consistency of behaviour during a single horizon length (*i.e.* "stationarity" in terms of entropy). With these considerations, a value of $Q = 12$ was chosen with a view to maximizing the differentiability in the plots while keeping the

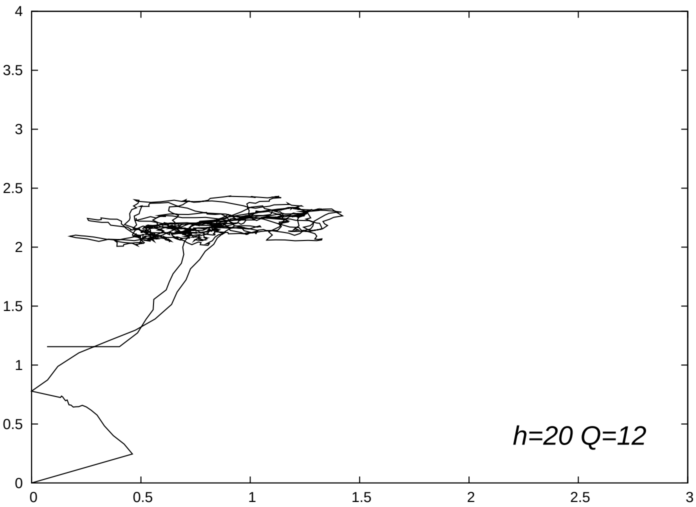


Figure 4.7: AID *vs.* AID *plot for selected values of horizon length* $h = 20$ *and number of bins* $Q = 12$*.* Horizontal and vertical axes are the moving window AID for sensors and motors respectively.

computation time to an amount reasonable for on-line computation at 10 frames of data per second and the horizon was kept small $h = 20$ to show a large amount of detail only smoothing out short term variations. The AID *vs.* AID plot for these selected values is shown in Figure 4.7.

Table 4.4: Simple Behaviours Executed

Behaviour	Description	Runs[a]
walking	Walking from one end of the arena to another	11
turning	Turning on the spot in either direction	7
observing	Robot stationary with activity in environment, e.g. ball or hand waved in front of visual field	5
stationary	Robot remains stationary (with motors powered) in a static environment	1

[a]See text for notes on number of runs.

4.4 Characterizing Simple Behaviours

In this investigation the Aibo robot was placed in the same 2m×2m arena described in Section 4.3.1 but without any obstacles. The robot was programmed to execute a single behaviour at a time and data from the robot analysed using the AID *vs.* AID method. The goal was to determine if it was possible for the robot to distinguish one class of its own behaviour from another by means of analysis of AID *vs.* AID traces.

Four behaviours were studied; *walking*, *turning*, *observing* and *stationary* (see Table 4.4). The number of repeated examples of each behaviour were different as in the initial data gathering phase behaviours were repeated according to the variation in possibilities of executing each behaviour. For instance there are more variations possible for the robot "walking" in an arena (towards a wall, away from a wall, near a wall, in the centre, oblique path, *etc.*) than, say, "observing" (wave ball, wave hand *etc.*). Indeed for the stationary example only one variation was recorded as the requirement was for neither the robot to move or there to be any movement in the environment. Further or repeat investigation should consider having more examples of all these behaviours introducing redundancy where necessary to arrive at equal numbers of runs, as well as to consider more possibilities of behaviours.

4.4.1 Morphometrics

A trajectory of informational relationships between groups of sensors has been demonstrated, but one of the goals of this investigation is to allow the agent or robot to be able to characterize and identify behaviours. Therefore, the agent needs to be able to compare AID *vs.* AID plots, and one mechanism for this is to compare details of the trajectory, such as its shape. Three "morphometrics" are discussed, that is, "measurements of shape" that can be used to objectively describe the shape of the trajectory in just a few numbers.

Centre of Gravity (CoG)

The *Centre of Gravity* (CoG) of the plot describes the overall position of the trajectory in the 2-dimensional space described by the AID axis for each sensor group. The CoG is calculated by assuming each position on the plot to be a point of unit mass, and summing over all points (disregarding the first h points).

Direction of Movement

Additionally, the overall movement of the AID *vs.* AID plot during a behaviour can be examined. A vector was calculated for every point with reference to the next point in the trajectory as:

$$\vec{\mathbf{v}}_t = \begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} - \begin{bmatrix} x_t \\ y_t \end{bmatrix}$$

The overall direction of movement or *Movement Vector* of the AID *vs.* AID plot is then the sum of these vectors for all points (disregarding the last point and the first $h-1$ points).

Fractal (Capacity) Dimension

While it is possible to measure the position (CoG) of the curve in the AID *vs.* AID space, and its overall direction of movement, this does not tell us about what the curve does in this region. Does it follow a simple path? Does it repeatedly cross over itself? How "jagged" is the line? To measure this, it is possible to use a measure of the *fractal dimension* to describe a trajectory in terms of its "convolutedness"[4].

Fractals (see Fig. 4.4.1) have the property of invariance under change of scale, known as self-similarity and the *fractal dimension* is a measure of the self-similarity of a set (Baker and Gollub, 1990). There are many ways to define the fractal dimension and I will use the *capacity dimension* which is defined as

$$d_c = \lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log(1/\varepsilon)} \tag{4.2}$$

where $N(\varepsilon)$ is the number of boxes of size ε that can "cover" the figure. The capacity dimension can be estimated by using the "box-counting" method from an image of the plotted path. The "box-counting" method is a general procedure that can be applied to any image and proceeds by counting how many of N^2 boxes dividing the image have pattern detail in them and then iterating over N. The slope of a line fitted to a log-log plot of the results gives the fractal dimension.

4.4.2 Results

CoG and Movement Vector Combined

Figures 4.9 and 4.10 show the CoG and movement vectors of all 24 experimental runs. Figure 4.11 summarizes all runs of each behaviour type. Note that the movement vector magnitude and direction is combined with the CoG position to place the vectors on the graphs.

[4]Note that the trajectory is not a true fractal, and does not reveal more detail at higher resolution. However, the box counting method only estimates the fractal dimension and does not *require* the curve to actually be a true fractal.

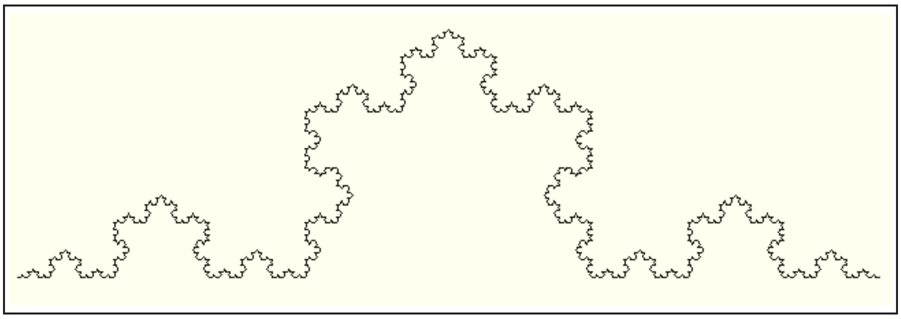


Figure 4.8: *Koch curve.* An example of a self-similar fractal shape. The curve has a theoretical fractal dimension of $\log 4/\log 3 = 1.262$. This compares to a value of 1.225 calculated by the simple box-counting method for the curve shown above.

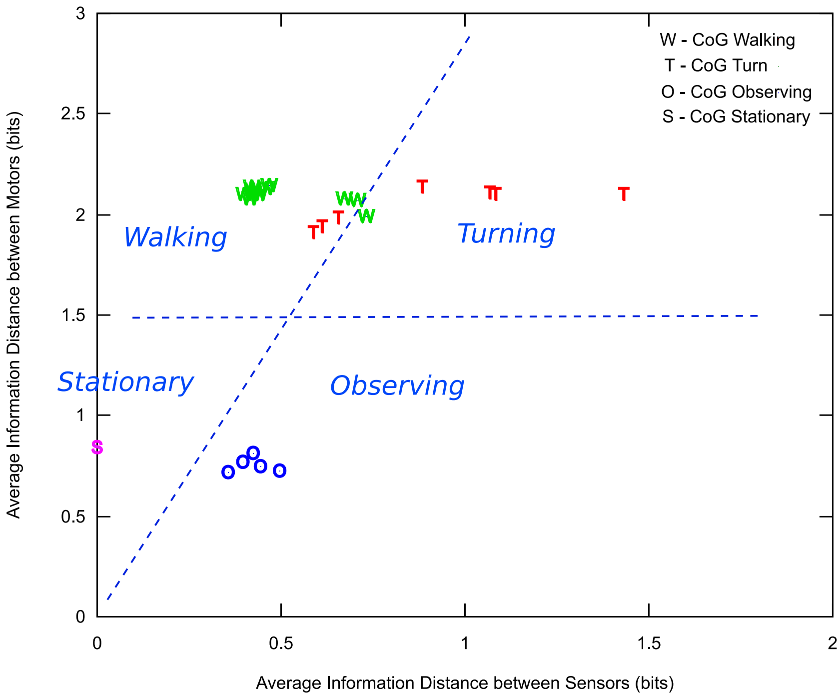


Figure 4.9: *CoG of* AID *vs.* AID *trajectories. Summary of 24 Simple Behaviour Experiments.* Showing Centre of Gravity of each trajectory. Experiments are in 4 categories *walking*, *turning*, *stationary* and *observing* (see Table 4.4). The 4 behaviours appear in 4 quadrants of the geometric space as indicated. Horizon h=20, number of bins Q=12.

It is clear from Figures 4.9 4.10 and 4.11 that *turning* and *walking* are very different from *stationary* and *observing*. This would be expected due to the activity of the motors. Moreover, the difference between being stationary in a quiescent and a changing environment is shown as a difference in the sensory AID, again as would be expected. This is can also be seen to some extent with *turning* and

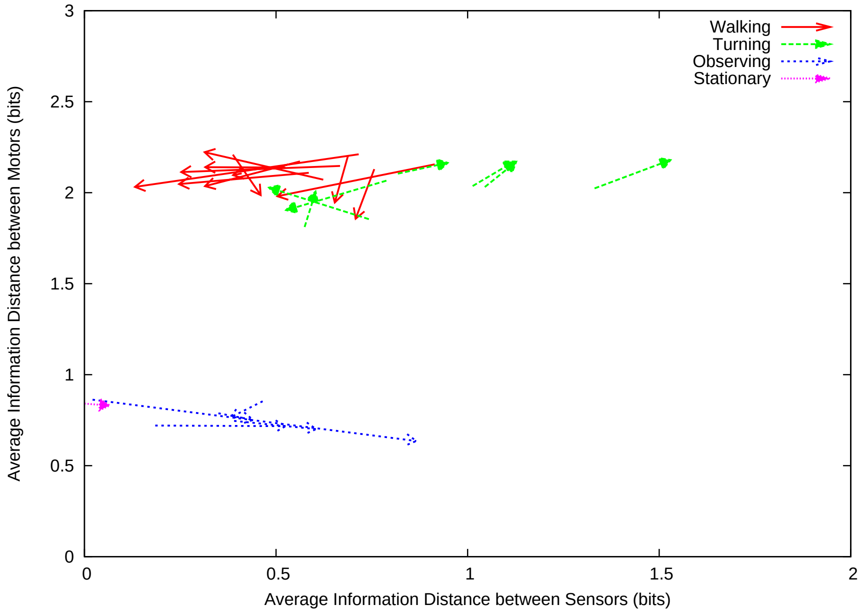


Figure 4.10: *Movement vectors of* AID *vs.* AID *trajectories. Summary of 24 Simple Behaviour Experiments.* Showing overall direction of movement of each trajectory as vectors. These show further distinction between behaviour types. Experiments are in 4 categories *walking*, *turning*, *stationary* and *observing* (see Table 4.4). Horizon h=20, number of bins Q=12.

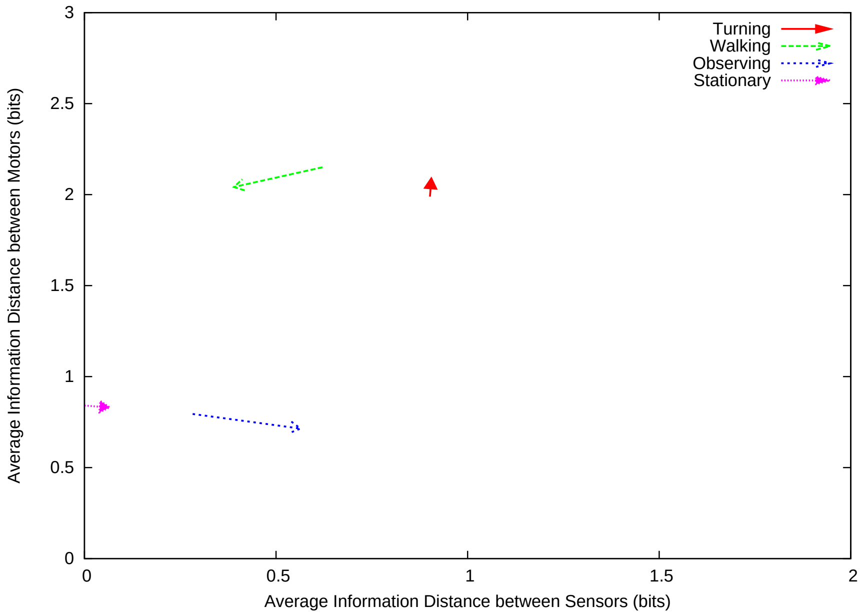


Figure 4.11: *Summary of the CoGs and Movement Vectors of the 4 Behaviour Types.* All vectors of each behaviour type are summed to give an overall vector for each behaviour, likewise for CoG. Behaviours are in 4 categories *walking*, *turning*, *stationary* and *observing* (see Table 4.4). Window size τ=20, bin size Q=12.

walking, the former being characterized by a far more rapidly changing sensory input from the vision sensors.

Furthermore, it is interesting to note that *turning* and *walking* are further distinguished by how their respective AID *vs.* AID plots change during the behaviour, as shown by their Movement Vectors; *walking* has sensory and motor AID reducing while *turning* has (for most of the examples) motor and sensor AID increasing.

Note, that even if the CoG and Vector direction were used in combination to characterize the robot behaviour, not all the *turning* and *walking* examples in this data-set can be distinguished.

Table 4.5: *Fractal dimension.* Summary of results for capacity (fractal) dimension calculated by the box-counting method for a robot conducting simple tasks.

Behaviour	mean d_c	min d_c	max d_c	Std Dev
walking	1.1886	1.1111	1.2521	0.04665
turning	1.3153	1.2600	1.3765	0.03985
observing	1.3133	1.2838	1.3450	0.02841
stationary	0.8663	0.8663	0.8663	N/A

Fractal Dimension

In order to better characterize the trajectories corresponding to robot behaviours, the fractal dimension estimated by box-counting is considered as an additional morphometric on the AID *vs.* AID trajectories. Typical plots resulting from the four types of behaviour conducted are shown in Figure 4.12 along with their fractal dimension. The results for all the experimental runs are summarized in Table 4.5.

The fractal dimensions calculated for all of the *turning* and *walking* experiments fell in non-overlapping ranges and indicates that this group of measurements are linearly separable into the respective types.

Observe that *turning* has a higher fractal dimension than *walking*, which would be expected as the path appears more convoluted. Conversely, *observing* was found to have a very similar fractal dimension to *turning*. I speculate that the waving of the hand and ball in front of the AIBO camera (*observing*) and *turning* provide a similarly high degree of rhythmic sensory activity whereas *walking* straight ahead provides a steady visual field along with a steadily reducing infra-red range reading. The *stationary* behaviour appears as a straight vertical line on the plot and consequently has a fractal dimension close to 1.0.

4.4.3 Combination of Morphometrics

It was expected that all three measures of the shape of the AID *vs.* AID plot trajectory when combined would give a superior objective separation of the sub-

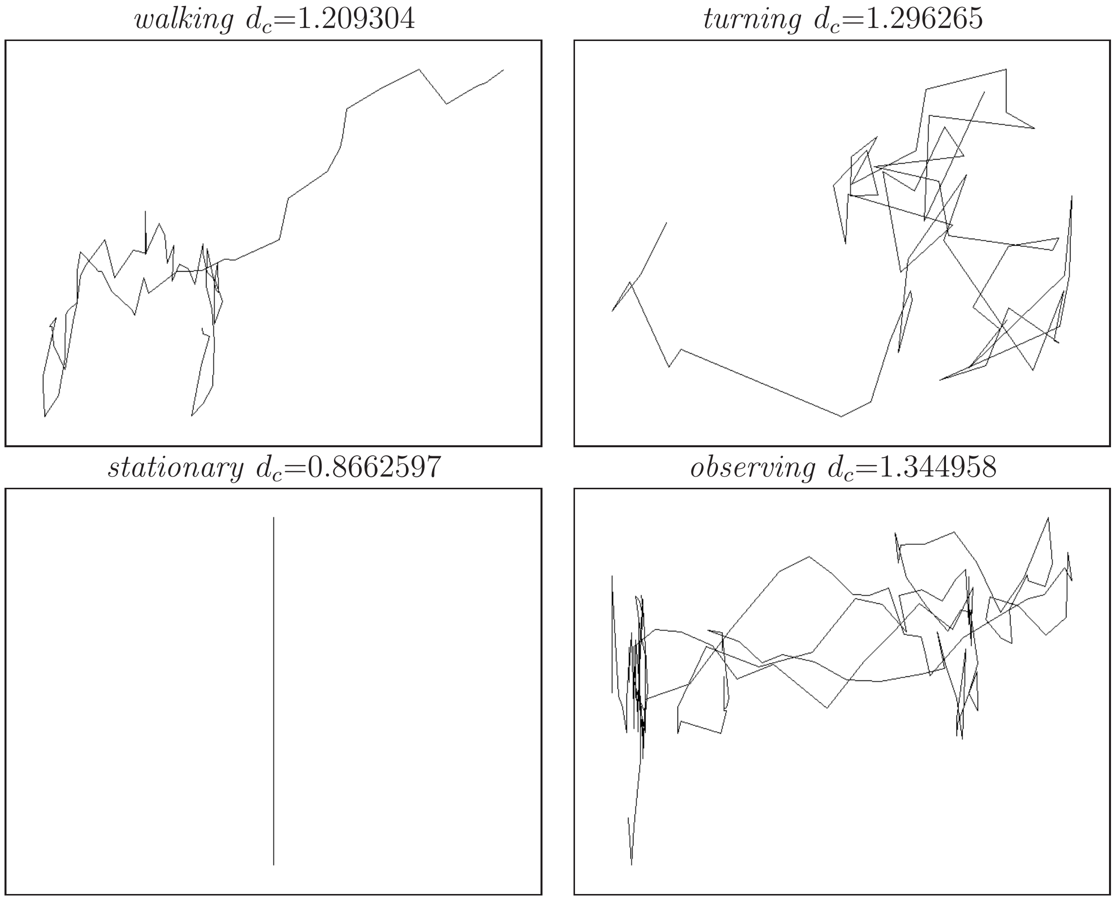


Figure 4.12: AID *vs.* AID *plots.* Typical trajectories for each type of behaviour studied along with the calculated box-counting fractal dimension for each figure.

jective behaviour categories. Combining the measurements, the trajectory can be represented in 5 dimensions: 2 for CoG, 2 for Movement Vector and 1 for Fractal Dimension. Figure 4.14 shows a hierarchical clustering of the simple behaviours based on the euclidean distances in the 5-dimensional space of the morphometric measurements. It can be seen that the behaviours separate well, except for Turn05 which clusters with the Walk behaviours. Turns 06 and 07, while clustering with the Walk examples, do form their own cluster at a separate hierarchical level from the Walk behaviours.

4.4.4 Discussion

It is clear by inspection that the trajectories produced for the different behaviours tested were different in terms of general shape. This was confirmed by using

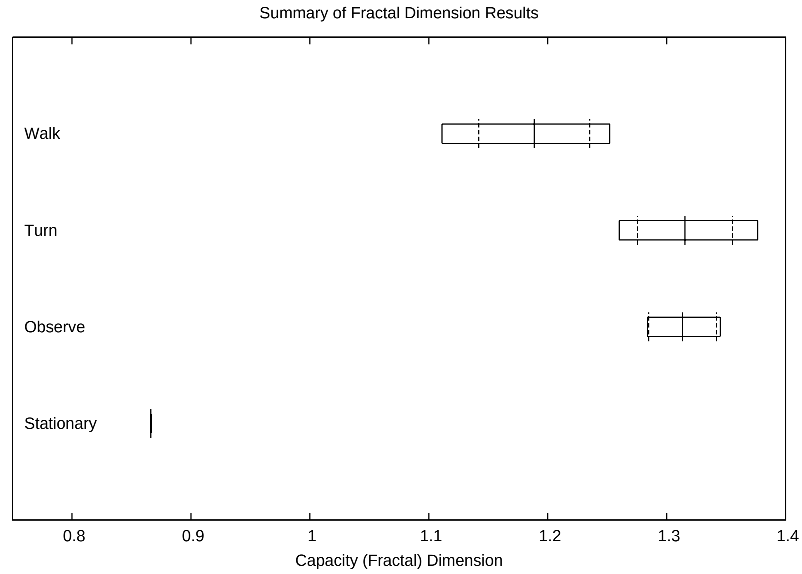


Figure 4.13: *Summary of Fractal Dimension Results.* Plot shows range of fractal dimensions for each behaviour from the experimental dataset (boxes), their means (solid vertical lines) and standard deviation (dotted lines).

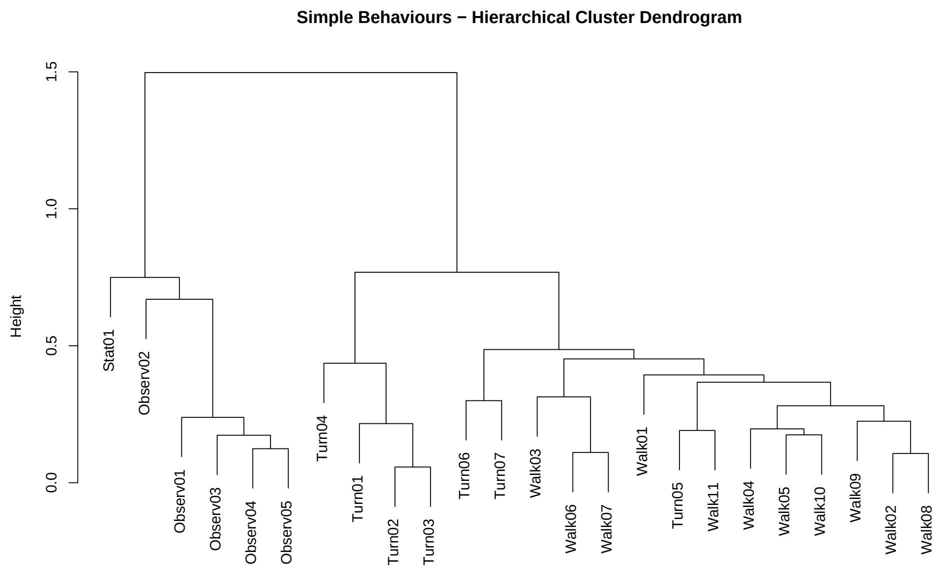


Figure 4.14: *Hierarchical Clustering of Simple Behaviours.* This dendrogram was produced using the morphometrics CoG (x and y), Movement Vector (x and y) and the Fractal Dimension, and used average cluster centre linkage. It can be seen that the behaviours separate reasonably well, except for Turn05 which clusters with the Walk behaviours, and turns 06/07 which are marginally closer to the walk behaviours than their fellow turning behaviours.

very simple measurements of the shape in terms of its position (CoG) and overall movement direction vector. Some individual examples of the *walking* and *turning* behaviours were not clearly separated using CoG and Movement Vector alone. However, a further measure of the shape of the trajectories - the fractal dimension - was able to make a clear distinction between all examples of *turning* and *walking*. A simple linear combination of all three metrics still however results in some overlap in the clustering of behaviours, although the groupings could suggest that either the subjective characterization of the behaviour by the observer was incorrect, or simply that *to the robot*, certain "turning" episodes (near a wall?), just "feel" like particular "walking" episodes.

It should be emphasized then, that clustering and characterization that is grounded in the robot's own experience may well be different to that which an observer may apply.

Clearly, the characterization presented here was done after the fact, however, to achieve this on-line, a neural network approach may be appropriate. One possibility is to train a feed-forward artificial neural network to recognize the categories based on a training set categorized by hand. However, a self-organizing map approach may be able to avoid the supervised training, and therefore would be more appropriate. This assumes that the behaviours can be segmented one from the next (for instance by using constant sized segments). Suggesting an approach to automatic segmentation is the subject of section 4.6.

4.5 Identifying Behaviours

The aim of this experiment was to see if the simple behaviours of section 4.4 could be identified within a sequential series of such behaviours. The overall behaviour executed was *exploring* as described in section 4.3.4 consisting of walking and turning behaviours.

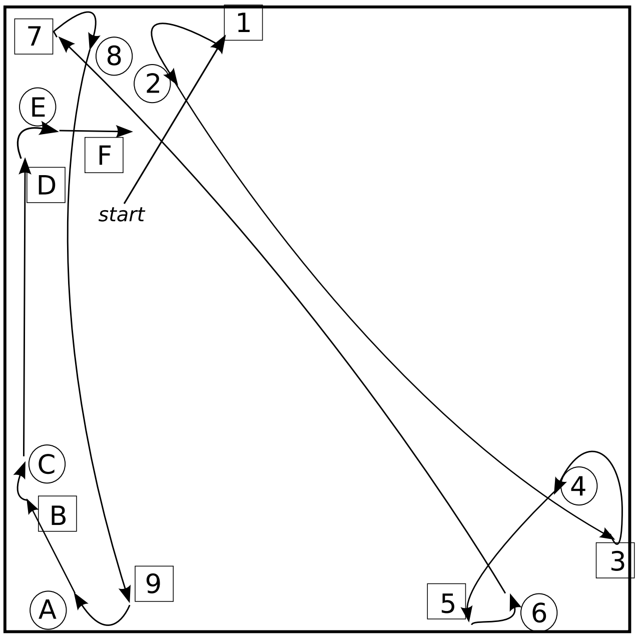


Figure 4.15: *Path traversed by AIBO during a particular "explore" experiment.* View is overhead of the 2m × 2m arena. Numbers are waypoints marking changes in behaviour. *end-of-turn* waypoints are circled, *end-of-walk* waypoints are enclosed in a square. `EXPL02` dataset.

The path traversed by the robot in the arena, estimated from an overhead video, is illustrated in Fig. 4.15 and annotated with numbered waypoints. The waypoints were chosen at points where behaviour changes from walk to turn or *visa versa*, and labelled such that they describe the just-completed behaviour (as determined by an observer). As this dataset is used in a number of subsequent experiments it will sometimes be referred to as the `EXPL02` dataset.

The CoG and Movement Vectors for the AID *vs.* AID plot describing the be-

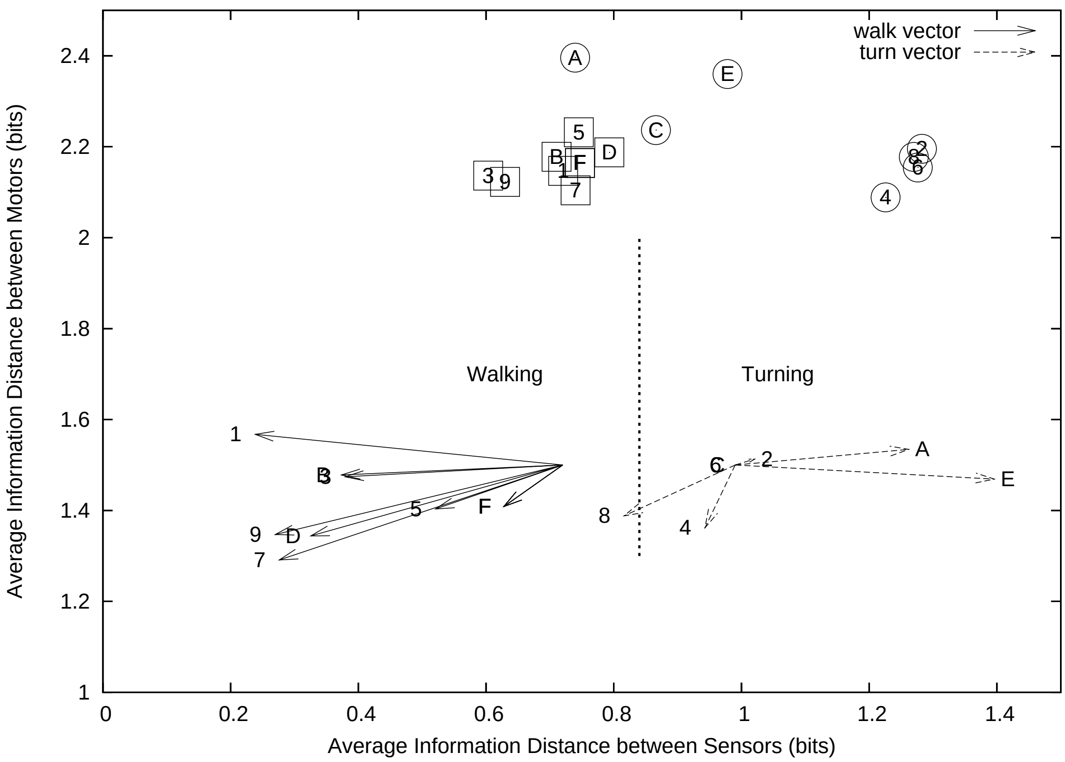


Figure 4.16: *CoG and movement vectors of sensor-motor* AID *vs.* AID *plot of waypoints of Fig. 4.15.* The behaviour sequence was separated into sections and labelled as *Walk* or *Turn*. The CoG and Movement Vectors for each segment were calculated and are shown separately in this combined plot. Note: trajectory itself not shown for clarity. CoG of *Walk* behaviours (1,3,5,7,9,B,D,F) appear on left side. CoG of *Turn* behaviours (2,4,6,8,A,C,E), while not as well grouped, appear further to the right. Vectors for *Walk* move to the left, those for *Turn* largely either are moving to the right or the CoG is already at the right side. $h = 20$ and $Q = 12$.

haviour were calculated and are shown in Figure 4.16. Note that only timesteps greater than the horizon h were counted in these summary morphometrics, as there is a delay effect due to the probability estimation over the moving window. In some cases, usually with the shorter duration Turn behaviours and in particular with behaviour numbers 4, 8 and 14, there are not enough samples to fill a whole window of horizon h, and so only the last timestep in that sequence was considered. This results in zero length movement vector, but a reasonable estimation of the position of the CoG for the plot.

4.5.1 Discussion

It can be seen from the plot of Figure 4.16 that the Walk behaviours are well grouped; they lie mainly to the left of the plot (lower AID between sensors in the Sensor group) and their respective movement vectors generally point to the left. The Turn behaviours are not so homogeneous, but the prevailing characteristic is of a greater AID between Motor group sensors and an increasing movement vector. *i.e.* the plot is moving towards the right.

Further characterization may be possible using additional morphometrics such as the fractal dimension, but for the short length behaviours of the Turn sections, this would result in plots that were inadequate for fractal dimension estimation using box counting. Self-identification of behaviour is thus possible, however, once again automatic segmentation would be required for an on-line solution.

4.6 Segmentation of Behaviour

The preceding sections have described a technique for characterizing behaviour from the robot's perspective, but using externally determined behaviour transition boundaries. As a step towards autonomous self-categorization, this section takes a tentative look at some possible techniques that automatically segment behaviour using the AID *vs.* AID plot as the starting point.

4.6.1 Segmenting Behaviour in the AID *vs.* AID Plot Using Transition Threshold

Investigation of the AID *vs.* AID plots reveals localized activity in different regions of the AID *vs.* AID space punctuated by transitions between regions, and the position and other characteristics of the localized activity was shown to characterize behaviour from the robot's perspective. These properties are now exploited to allow the robot to segment experience in terms of an AID *vs.* AID plot into frames of coherent activity.

Algorithm 4.1: `SEG_THR`: Incremental Threshold-based Segmentation using AID *vs.* AID

Input: threshold τ, window w
Output: segments $\mathbf{S}$

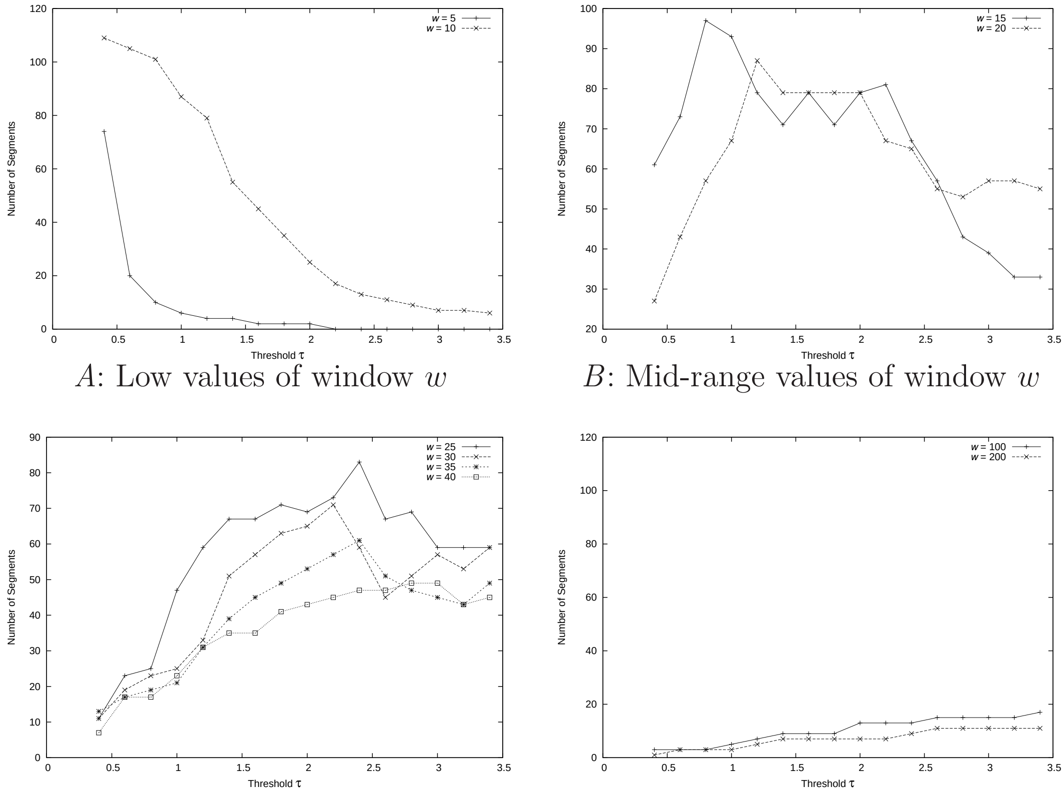
```
V: window,length w, vector of AID pairs
state ⟵ stasis
while A = new AID pair do
    push A to front of V, pop back
    calculate Σ_v = sum of V
    if Σ_v > τ and state = stasis then
        start new segment S_i, type motion
        state = motion
    end
    if Σ_v < τ and state = motion then
        start new segment S_i, type stasis
        state = stasis
    end
end
```

The magnitude of a vector, calculated over w time-steps, describing the current direction of the trajectory in the AID *vs.* AID space is used to indicate movement away from a localized region. When the magnitude is greater than a threshold τ, then a period of transition begins, ending when the magnitude falls below the threshold. This procedure is appropriate for on-line segmentation as new data arrives, see the Algorithm 4.1:`SEG_THR`. This procedure results in periods of "stasis" where the trajectory does not move very fast, and periods of "transition" where the trajectory is moving faster. Both types can be considered to be segments of behaviour and can be characterized using the morphometrics described.

Segmentation using this technique depends on the threshold τ, the length of the window over which the transition is estimated w and the environment. The effect of different values of both τ and w are examined, with the view that these parameters may be autonomously adapted to the conditions in the future.

4.6.2 Experimental Investigation

The same wandering walk used in Section 4.5 (`EXPL02` dataset), consisting of 15 individual walks and turns, was subjected to segmentation using the method



A: Low values of window w *B*: Mid-range values of window w

C: High values of window w *D*: Very High values of window w

Figure 4.17: *Threshold Effect on Segmentation* Number of frames produced for different values of the threshold τ and vector estimation window size w.

described above. The number of segments produced was found to depend on both τ and w. Figure 4.17 shows that for a small window w the number of frames reduces as the threshold is increased. This relationship is reversed for higher w.

To achieve a similar number of segments to the observed case (15) requires either a low threshold combined with a long window, or a short window with a medium sized threshold. Examples of segments achieved at selected window lengths and thresholds are shown in Figure 4.18. This figure also shows that as the window length increases, the bulk of the time-series is segmented as "motion", *i.e.* transitions, whereas for short windows, especially combined with higher thresholds, the segments that compose most of the time-series are of the "stasis" type.

None of the segments in any of the cases appear to accurately represent turning and walking as one observes them. This may be due to the choice of segmentation

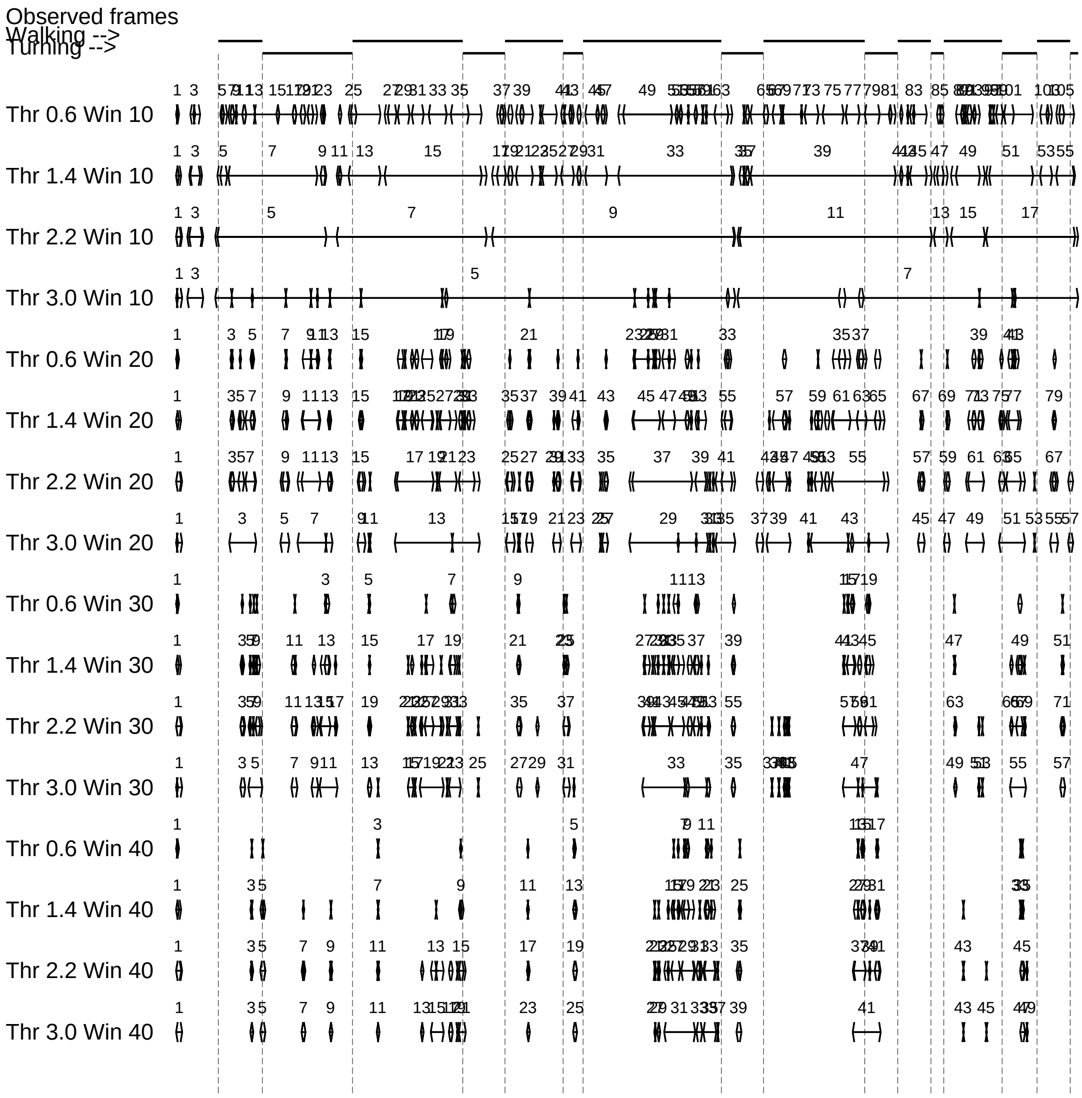


Figure 4.18: *Wandering behaviour segmented into time-frames for a range of threshold and window values.* Each line on the figure shows a different segmentation of the same data with the segments numbered consecutively. Periods of "stasis" (odd-frames) only are marked. Observed behaviour (walk/turn) is shown on the top part of the graph as a reference.

technique, alternatively they may represent "close-to-wall" and "far-from-wall" or some other subjective (from the robot's perspective) behaviour.

4.6.3 Identification of Behaviour Segments

The next step is to characterize the behaviour segments in terms of known behaviour. The CoG and Movement vector measures are used, leaving out the box-counting dimension due to its unreliability on short segments as noted above. A window size $w = 10$ and threshold $\tau = 1.8$ were chosen on the basis of the number of segments which resulted being of a similar order to the numbers of observed behaviours. The behaviour thus divided into 35 behaviour segments, with 90% of the time-series segmented as "stasis". To illustrate how the behaviours might be identified, Figure 4.19 shows hierarchical clustering applied to the segments (stasis1-35 and motion2-34), along with the simple behaviours (S01, O01-05, T01-07 and W01-11) of Section 4.4 for reference. The segments broadly segment into three groups (boxes on the dendrogram). On the top are the Observing and Stasis behaviours from the simple behaviours, clustered with segmented behaviours from the very start of the time-series. In the centre are Turn behaviours, and at the bottom are mainly walk behaviours along with Turns 5,6 and 7 which previously had clustered with Walk. The segments that wholly lie during turn behaviours (6, 7, 8, 12, 13, 14, 22, 27) all segment with the turn behaviours. Of the segments that wholly or mostly lie within walk sequences (9, 10, 11, 15, 16, 18, 19, 20, 21, 25, 28, 29, 30, 33, 34, 35), most segment with Walk behaviours with the exception of 11 and 21 which both *end* as turns, and 28.

This result suggests that, where the behaviour within an automatically generated segment is unambiguous, the morphometrics can be applied to successfully identify behaviour from the robot's perspective.

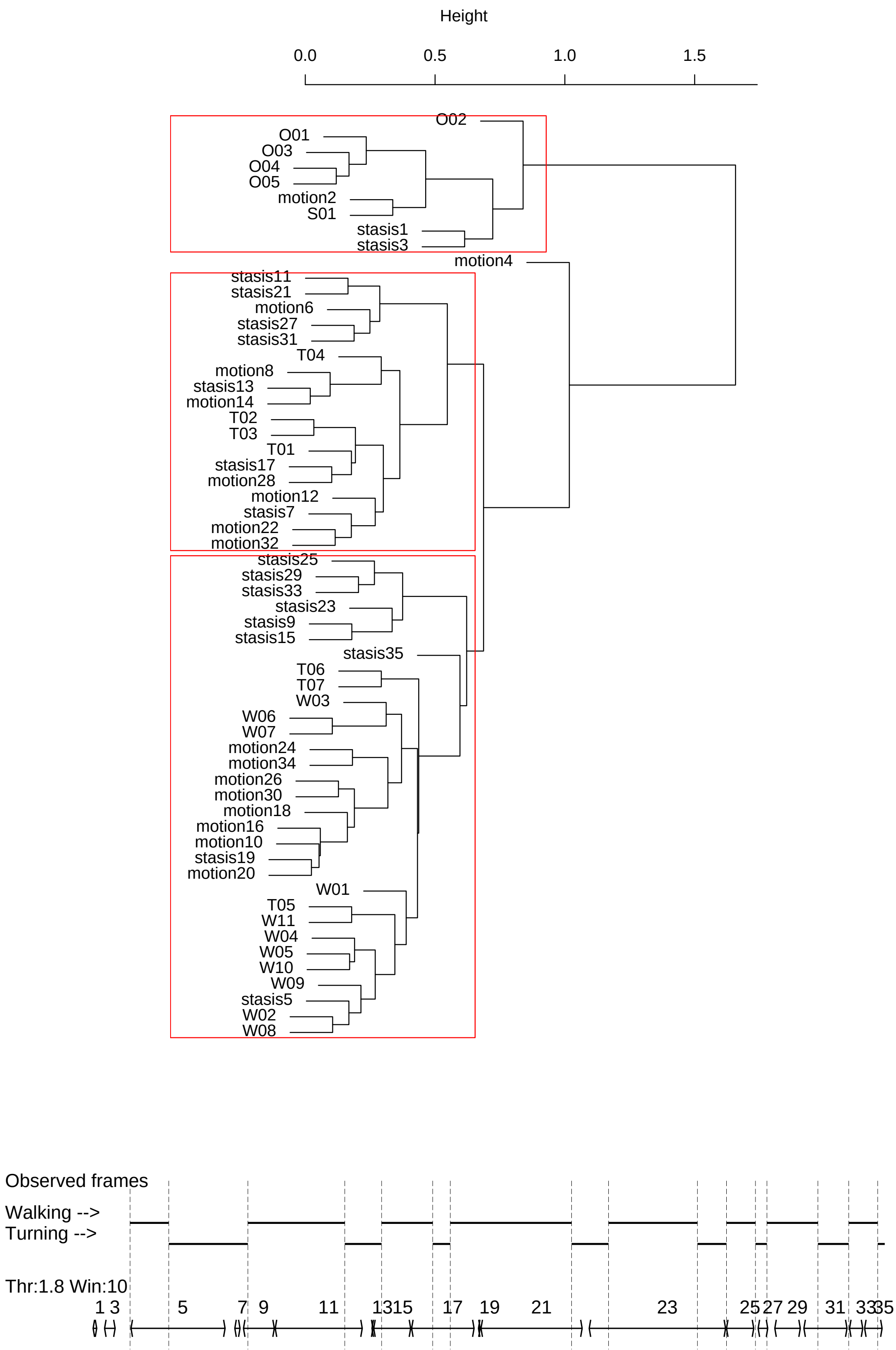


Figure 4.19: *Hierarchical clustering of autosegmented behaviours combined with simple behaviours for reference.* Segments produced using threshold τ=1.8 and window w=10. Morphometrics CoG (x and y) and Movement Vector (x and y). Average cluster centre linkage. Compare with Figure 4.14.

(Below) Segmentation of time-series, compared with observed behaviour.

4.6.4 Discussion

The literature on pattern identification, regression analysis and machine classification is vast (see for example Mitchell, 1997; Jain, Duin and Mao, 2000). When faced with very high dimensional data, these approaches either reduce dimensionality drastically using techniques such as linear regression analysis and principal components analysis, or instead they start with a lower dimensional data set by using "features" of the data, often hand-designed. With reduced dimensional data, classification can be performed using tools such as neural networks, linear discriminant analysis, nearest neighbour rules and template matching to name just a few.

With this in mind, I cannot claim that there are not better techniques for segmenting behaviour. However this approach was taken only with a view to seeing how far it is possible to go with the AID *vs.* AID plot. In the AID *vs.* AID plot, there is drastic dimensionality reduction, and there is good evidence that the plots can be used to characterize behaviour. However, the segmentation of behaviour seems to suffer from being performed on data that has already been reduced in dimension. Instead, better results may be achieved by using more traditional techniques from machine learning on the raw data.

As mentioned in Section 2.5, much of the existing research into classification in robotics is in object identification and localization. An example of the latter is (Nehmzow and Smithers, 1991) where the robot uses Kohonen self-organizing networks to build internal maps. Interestingly, from the perspective of this thesis, they required a history of previous encoded sensing to solve their localization task, and this again emphasizes the importance of a continually constructed interaction history of some kind in robots tackling complex, time-dependent, tasks.

4.7 Chapter Summary

This chapter has defined a measure, the Average Information Distance, that a robot could potentially use to characterize and recognize its own behavioural interaction with the environment. Using two groups of sensors representing the agent and environment, a trajectory in an AID *vs.* AID space further serves to characterize behaviour. Applying measurements of shape ("morphometrics") to the trajectories resulted in a quantitative method for characterization and identification. Furthermore, movement in the trajectory can serve as a method for automatic segmentation of one behaviour from the next.

However, there are limitations to the approach. Not all behaviours were easily separated, and the choice of parameters for segmentation seems critical. Additionally, I expect that taking a gross average of the distance between all sensors in a group removes too much information from the data. This may be resolved by using more, or automatically generated groups, that show internal consistency. The next chapters though, take a new approach, with the hope of solving some of these problems.

Chapter 5

Sensorimotor Experience and Metrics

5.1 Introduction

The preceding chapter explored the characterization of robot behaviour in terms of relationships between time-series of sensor readings and the subsequent dimensionally reduced trajectories in the AID *vs.* AID space. In this chapter an alternative approach is taken, applying the information distance to the same sensors at a different time. The notion of "temporally extended experience" is operationalized using the flow of values over the agent's sensorimotor variables during a particular interval of time (temporal horizon). Furthermore, clear mathematical relationships and measures between experiences are presented that provide potentially better characterization of robot behaviour and interactions.

Later, in Chapters 7,8 and 9, robotic sensorimotor experience and their relationships defined in this chapter are used to close the perception-action loop to create simple robotic "intelligence" operating on a broader temporal horizon based on grounded sensorimotor interaction histories. Experiments in this chapter (Section 5.4) lay the groundwork for this by using the robot's interaction histories to anticipate future sensorimotor experience.

5.2 Sensorimotor Experience and Metrics

A robot or other embodied agent's entire view of the world is experienced through its sensors, including those that measure internal factors such as temperature, motor positions, and other more general internal variables. As explained in Section 3.3 these can be modeled as random variables. A robot's experience, then, can be considered as the collection of all readings from all these variables over a given time period. This is a purely sensorimotor view of experience and says nothing about the quality or meaning of that experience.

Formally, an agent's *experience* from time t over a temporal horizon h can be defined as

$$E(t,h) = (\mathcal{X}^1_{t,h}, \dots, \mathcal{X}^N_{t,h}) \tag{5.1}$$

where $\mathcal{X}^1_{t,h}, \dots, \mathcal{X}^N_{t,h}$ is the set of random variables available to the agent constructed from time-series of sensorimotor readings from N sensors $(X^1, \dots, X^N)$ ending at time t with a horizon h timesteps (from time $t-(h-1)$ to t)[1].

Of course it is also possible to envisage sub-experiences within any experience made up of a subset of the sensors. Thus, for a subset $\mathbf{S}$ of the sensory variables, $E^{\mathbf{S}}(t,h) = (\mathcal{X}^k_{t,h})$ (where each $\mathcal{X}^k \in \mathbf{S}$) is a *sub-experience* of $E(t,h)$. This would correspond to the definition given in (Oates et al., 2000)[2]. Furthermore, it is possible to envisage a dimensionally reduced experience

$$F(t,h) = (\mathcal{Y}^1_{t,h}, \dots, \mathcal{Y}^M_{t,h}) \tag{5.2}$$

where $\mathcal{Y}^1, \dots, \mathcal{Y}^M$ are M remapped sensor variables with $M \le N$. Such dimension reduction could be performed with any standard method such as Multi-Dimensional Scaling (MDS), Principal Components Analysis (PCA) or Isomap (a non-linear multidimensional scaling algorithm).

[1]Note that for the definition it is equivalent to say starting at time t with horizon h, however, implementations must be consistent. In all my implemented code, the time of the sensor reading or experience is taken as the time at the end of the time-series, and h the length of the timeseries.

[2]Note that while the Oates paper predates our publications on Experience (see Appendix B), it was not known to me at the time of writing, and this formulation was arrived at independently.

5.2.1 Experience Metric

Given a definition of Sensorimotor Experience and the information metric, a formal measure of distance between experiences can be defined. This is useful as it allows a direct, scaled, comparison between different sets of sensorimotor readings of a robot or agent. A metric for comparison of sensorimotor experiences is important as it is then possible to talk of proximity and distance between different experiences in a quantitative way.

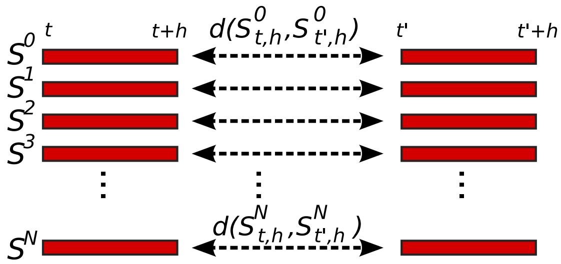


Figure 5.1: *Experience Metric.* A visual illustration of the experience metric. Each experience is shown as a collection of sensor readings of length h starting at time t and t'. The information distance between each respective sensor over time is summed to give the Experience Metric.

The *Experience Metric*, a metric on experiences of temporal horizon h, is defined as

$$D(E, E') = \sum_{k=1}^{N} d(\mathcal{X}^k_{t,h}, \mathcal{X}^k_{t',h}) \tag{5.3}$$

where $E = E(t, h)$ and $E' = E(t', h)$ are experiences of an agent at time t and t' over horizon h and d is the information distance (see Figure 5.1). D is measured in *bits*. That D is a metric follows from the fact that the metric axioms (equivalence, similarity and triangle inequality) hold for each of the components in the summation, since d is a metric (see Section 3.2.2).

Other Metrics on Experience

As before for the definition of experience (Equation 5.1) it is possible to extend the definition of experience to metrics between experiences consisting of subsets of sensorimotor variables or remapped (dimensionally reduced) sensors. However, for the definition of Equation 5.3 to apply, the experiences being matched must consist of equal numbers of sensors with equivalent horizon lengths. Where the sensors compared are different then the reformulated metric will have a different meaning (*i.e.* it will not indicate how given groups of sensors are directly related over time). Finally, it is also possible to make comparisons between different numbers of sensors with different horizon lengths. A few possible metrics are listed below.

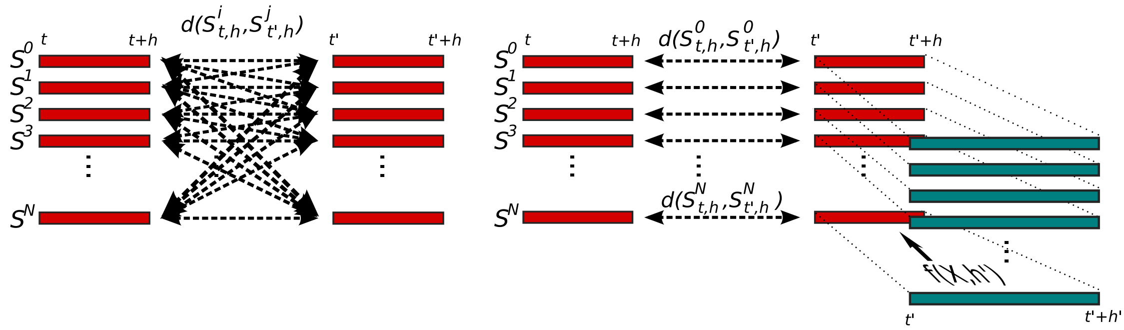


Figure 5.2: *Other Experience Metrics.* A visual illustration of two other possible experience metrics. (Left) The Intersensor Temporal Experience Metric, (Right) The Horizon Asymmetric Experience Metric: showing one possible mapping. In this case the current experience of length h' is mapped to an experience of length h, alternatively, the mapping could be done on the history experience. See text for details.

Intrasensor Temporal Experience Metric: For disambiguation this is the Experience Metric as defined in Equation 5.3.

Intersensor Metric: Informational relationship between sensors measured over the same time. As used by Olsson, Nehaniv and Polani (2006*a*) to create sensoritopic maps.

$$D_{Intersensor}(E(t,h), E(t,h)) = \sum_{i=1}^{N}\sum_{j=1}^{N} d(\mathcal{X}^i_{t,h}, \mathcal{X}^j_{t,h}) \tag{5.4}$$

"Cross-modal" or Intersensor Temporal Experience Metric: This metric compares sensors both over time and between themselves. It combines the ideas of the Temporal Experience Metric and the Intersensor Metric.

$$D_{IntersensorTemporal}(E(t,h),E(t',h)) = \sum_{i=1}^{N}\sum_{j=1}^{M} d(\mathcal{X}_{t,h}^{i},\mathcal{X}_{t',h}^{j}) \qquad (5.5)$$

One interesting point about this metric is that it can be sensory asymmetric where $N \neq M$. This may be important in a system where development increased the sensory array over time. Even where hardware did not develop, adding new physical sensors, this could be software controlled for instance by ignoring certain sensors early in development and gradually reintroducing them as development progressed. A further possibility is increased numbers of visual sensors derived from the camera images corresponding to an increase in visual acuity over time.

An alternative metric that can also be used to compare groups of sensors in a metric space is the Hausdorff experience metric (Nehaniv, 2005).

Dimensionally Reduced Experience Metric: This metric defines how experiences can be compared after dimension reduction. Note that correspondence between sensors is important, *i.e.* the same dimension reduction must be performed on both experiences. For example, if PCA is used, then the PCA should be calculated on either one of the experiences or a combination of both, then the sensors rescaled using the same principal components.

$$D_{DimensionReduced}(E(t,h),E(t',h),f()) = \sum_{k=1}^{M} d(\mathcal{Y}_{t,h}^{k},\mathcal{Y}_{t',h}^{k}) \qquad (5.6)$$

where there is a suitable function $f()$ (such as PCA) where $f() : (\mathcal{X}_{t,h}^{1},\ldots,\mathcal{X}_{t,h}^{N}) \mapsto (\mathcal{Y}_{t,h}^{1},\ldots,\mathcal{Y}_{t,h}^{M})$ with $M \leq N$.

Horizon Asymmetric Experience Metric: This metric opens the possibility of comparing experiences of different horizon lengths. A mapping function is used to convert a sensor of length h' to one of length h.

$$D_{HorizonAsymmetric}(E(t,h), E(t',h')) = \sum_{k=1}^{N} d(\mathcal{X}^k_{t,h}, \mathcal{Z}^k_{t',h}) \tag{5.7}$$

where there is a function: $f(\mathcal{X}_{t,h'}) = \mathcal{Z}_{t,h}$: where in general $h' \neq h$. That is, a mapping function that maps $x(t), x(t+1), \ldots, x(t+h'-1)$ into $z(t), z(t+1), \ldots, z(t+h-1)$.

A suggested mapping function for $h' > h$ is $z(t+j) = x(t+i)$ where $j = \lfloor i\frac{h'}{h} \rfloor$.

Alternative mapping functions include a simple cut-off, where the longer experience is truncated at the horizon length of the shorter experience, as suggested in (Nehaniv, 2005).

5.3 Metric Spaces of Experience

The mathematical definition of a *metric space* generally given is, the pair (M, d) where M is a set of objects and a $d(x, y)$ is a distance function on the elements of the set which satisfies the axioms of symmetry, equivalence and the triangle inequality. See *e.g.* Rosenlicht (1985, p33).

If the distance function is the information distance and the set of objects, time-series of sensors, the metric space describes informational relationships between sensors. Such a space and its projections into a small number of dimensions have been used by Olsson et al. (2006*a*) to create sensoritopic maps.

Instead, I am interested here in a metric space $(\mathbf{E}, D)$ that is defined on a set of *experiences* $\mathbf{E} = E^0, E^1, \ldots, E^k$, as defined in Equation 5.1, and the *experience metric* $D(E^n, E^m)$. I will refer to this space as the *metric space of experiences*.

One might be tempted to think of a metric space in the same way as Euclidean space, however that can be misleading as the topological properties of a Euclidean space do not necessarily follow from the basic definition of a metric space as given above. The derivation of the toplogical properties of metric spaces defined on the information distance and on the experience metric is beyond the scope of this thesis and is put forward as important future work. Thus, for correctness, I do not assume any geometric or topological properties of the space and rely on the metric properties of the distance measurement only.

Thus, to know the distances from any given experience to all others, all the distances can be measured. This 1-dimensional space is referred to as the *local view* from experience E. See Section 5.3.2 and Figure 5.4. By extension, the *global view* is given by the union of all local pictures of all experiences in the space (and determines the metric space).

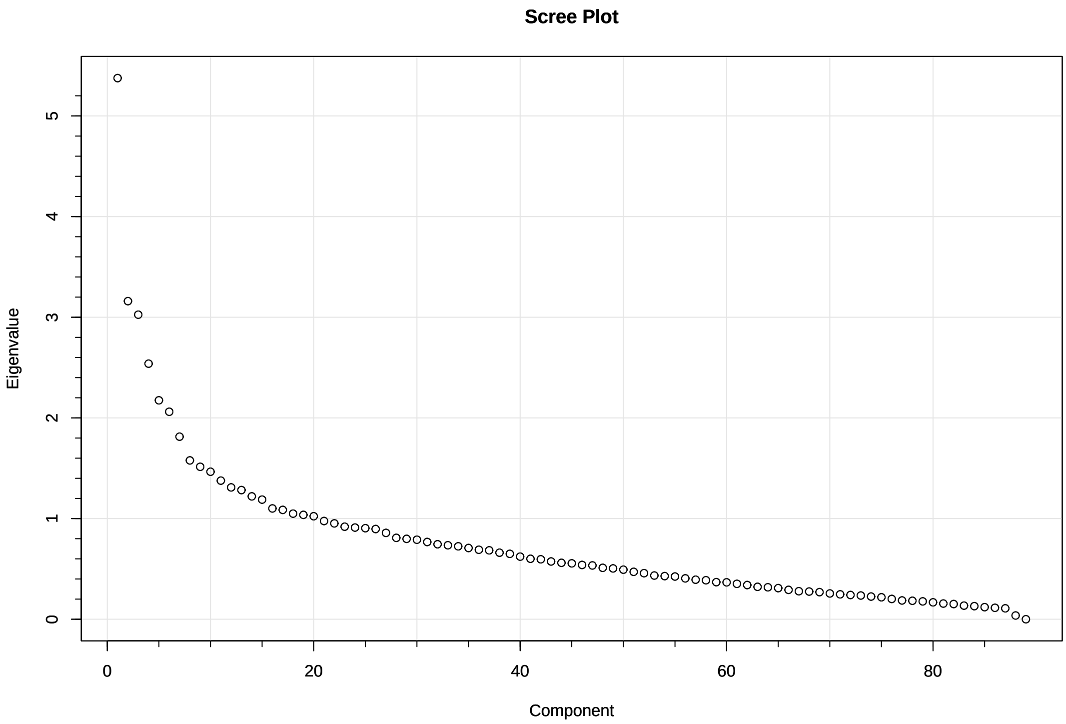


Figure 5.3: *Scree (Eigenvalue) Plot for 89 Experiences in a Metric Space.* Plot can be used for dimensionality estimation. In this case the estimated dimension is 8. Evenly spaced experiences from 911 timesteps of robot wandering (`EXPL02` dataset), $h = 20$ and $Q = 10$

.

5.3.1 Dimension of a Metric Space of Experience

Theoretically, the maximum dimensionality of the space is $N - 1$, where N is the number of experiences in the space, however, the actual dimension of the space is generally substantially less than that, and can be estimated by the "scree plot" method (*e.g.* Jolliffe, 2002, Section 6). Figure 5.3 shows such a plot for 89 evenly spaced experiences ($h = 20$, $Q = 10$) from the `EXPL02` dataset (see Section 4.5). The usual rule of thumb is that the dimension of the data can be reduced to be equivalent to the number of components at the "elbow" of the curve (Jolliffe, 2002), in this case 8. Alternative methods estimate the dimension to be equal to the component number where the eigenvalue becomes 1.0, in this case, approximately 20. In either case, it is clear that the dimensionality is a great deal less that the maximum, but this data may not be subject to linear dimension reduction (projection) into easily visualized two or three dimensional spaces.

The eigenvalues are estimated from the distance matrix using the method described by Gower (1966) using a mean-adjusted association matrix $\boldsymbol{\alpha}$: Given an $N \times N$ distance matrix $\mathbf{D}$ with elements d_{ij}, then the association matrix $\mathbf{A}$ has elements a_{ij} given by

$$a_{ij} = -\frac{1}{2} d_{ij}^2 \tag{5.8}$$

The mean-adjusted matrix $\boldsymbol{\alpha}$, then has elements

$$\alpha_{ij} = a_{ij} - \bar{a}_i - \bar{a}_j + \bar{a} \tag{5.9}$$

where $\bar{a}_i$ is the mean value of the ith row (or column) of $\mathbf{A}$ and $\bar{a}$ is the overall mean.

5.3.2 Views of the Metric Space of Experience

It is possible to visualize the metric space of experience by examining local views from selected experiences that show the experience distance from a particular experience to all others. Two horizon 20 length experiences from the `EXPL02` dataset (903 time-steps, 90.3 sec) were taken as comparison models; $E_{walk220}$ where the robot was walking forward (timesteps 200-220 corresponding to the early part of the path between waypoints 4 and 5 in Figure 4.15), and $E_{turn310}$ where the

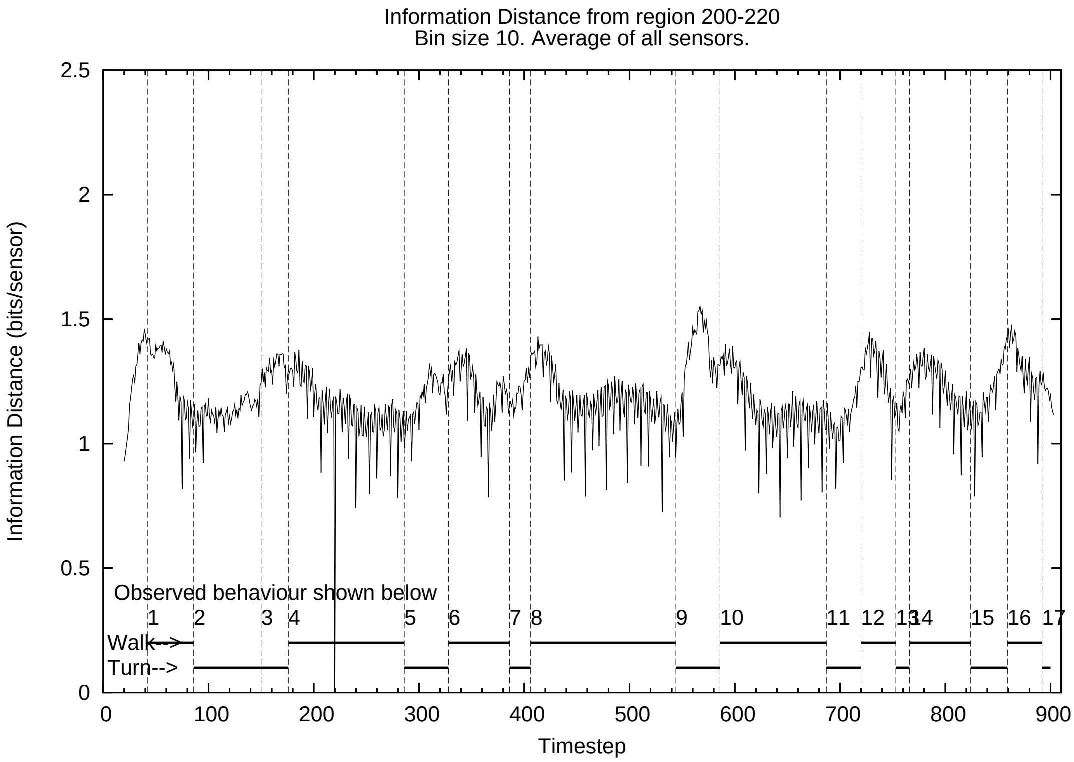


Figure 5.4: *Local Picture of Experience Metric Space From a "Walk" Experience.* The figure shows, in general a smaller distance to other walk experiences. All sensors considered. $h = 20$ and $Q = 10$

robot was turning near a wall (timesteps 290-310 corresponding to part of the turn between waypoints 5 and 6). These experiences were compared to all others using $Q = 10$ bins and the results are shown in Figures 5.4 and 5.5 alongside an indication of the transitions between observed behaviour.

Noting that a lower experience distance indicates similarity between the experiences, the results for the comparison of $E_{walk220}$ (Figure 5.4) show a reasonable agreement with the observed behaviour. For $E_{turn310}$ (Figure 5.5) this is less clear although the lowest information distances do correspond to other turning experiences.

Some of the experiences that *should* appear different to $E_{turn310}$ (*i.e.* all walking regions) do appear similar. On closer inspection it can be seen that the similarity tends to grow (*i.e.* distance falls) toward the end of a walk phase. This maybe because, as the AIBO approaches a wall, the experience (at least in visual and proximity terms) becomes more like that of a turn, which, owing to the reactive

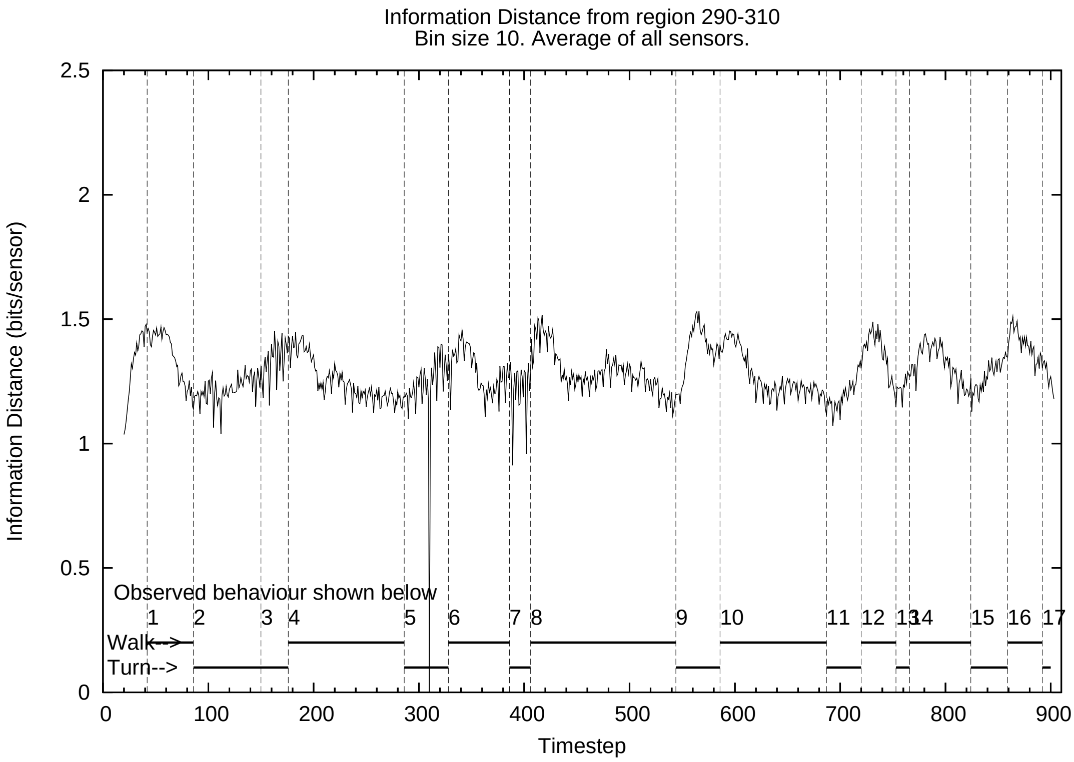


Figure 5.5: *Local Picture of Experience Metric Space From a "Turn" Experience.* The figure shows, in general a smaller distance to other turn experiences. All sensors considered. $h = 20$ and $Q = 10$

nature of the triggering of this behaviour, always occurs on approaching a wall.

To gauge how well experiences are matched to others of the same behavioural type, the experiences in the *neighbourhood* of $E_{walk220}$ and $E_{talk310}$ can be examined. Defining this neighbourhood as the collection of experiences that lie within a "sphere" of radius r bits centred on the experience E^t at time t:

$$B_r(E^t) = \{E^{t'} : \bar{D}(E^t, E^{t'}) \leq r\} \tag{5.10}$$

The experiences can be ranked in terms of their distance from a given experience. The 10 closest experiences to the $E_{walk220}$ for a window size of 20 and 10 bins, are shown in Table 5.1. Timestep 828 is 4 timesteps (0.4 seconds) into a turning phase and the error in classification is probably due to the time-window of data still containing more data from a walk action rather than a turn action. The sphere that contains these closest 10 experiences has a radius $r = 0.797$

Table 5.1: Nearest 10 Experiences to $E_{walk220}$ and $E_{turn3100}$

$E_{walk220}$

End Timestep	Behaviour Type
220	walk
643	walk
531	walk
240	walk
663	walk
280	walk
366	walk
458	walk
828	turn
253	walk

$E_{walk310}$

End Timestep	Behaviour Type
310	turn
389	turn
402	turn
112	turn
105	turn
693	turn
700	turn
290	turn
363	walk
694	turn

bits/sensor. For the experience $E_{walk220}$, the closest 10 experiences are mostly other turn experiences. Again, the only experience that is "misclassified" here is 363 which occurs 4 timesteps into a walking phase after a phase of turning. The radius of the sphere containing these experiences is $r = 1.1090$ bits/sensor.

5.3.3 Discussion

These results indicate that the closest experiences in terms of the experience metric, and thus from the robot's perspective, are of the same type as indicated by the external observer's classification (executed behaviour). This agrees with our hypothesis and encourages the use of the experience metric in classification and identification of behaviour. Many experiences that might have been expected to be closer, considering the executed behaviour, are actually farther away. However, this is likely to be because they differ in a manner not immediately apparent to an external observer. Such a situation was observed in these experiments: the "walk" behaviour when approaching a blank wall, appears from the robot perspective to be similar to "turn" when turning in front of a wall.

5.4 Robotic Experiments Using the Metric Space of Experience

This section presents a simple experiment, with two variants, designed to test the operation of the metric space (being both constructed and used for prospection) in real-time on a robotic platform. As new experiences are added to the metric space, the closest previous experience is used to construct a "predicted path" of a ball.

5.4.1 Static Path Prediction Experiment (SPPEXP)

In this experiment a ball is moved in front of a robot, *i.e.* in view of it's internal camera. The movements are simple and regular, for instance circular, vertical or horizontal movements. In the simplest version of the experiment, the robot is stationary. Sensorimotor data is collected, including those constructed from the camera images, and experiences are stored in a metric space. The order of experiences is retained, as well as an indication of the ball position at the end of the experience. The metric space of experiences is then used to construct a predicted path of the ball. This is done by finding the closest experience in the space to the current one; the ball position at the end of that experience and its subsequent experiences is then the predicted "path". Success would be indicated by correct prediction of the ball path, and this would only occur where the recent sensorimotor experience was correctly matched to a similar preceding experience.

It is important to note that, the robot is not matching current ball position with previous ball position, rather all sensory and motor[3] variables are used as information sources to detect similarity between experiences, and then the stored ball position is used to give the experimenter an indication as to how well the experience was chosen. For verification purposes a path is drawn on the display of the robot's visual field during operation, indicating the predicted future path.

[3]Even though in this first part of the experiment, there is no motor movement.

Implementation and Experimental Setup (SPPEXP)

The robot used was a Sony Aibo ERS-7 and the control and sensory collection software implemented using using URBI (Baillie, 2005). URBI provides the robot control layer and a full-featured event based parallel scripting system. The URBI software runs directly on the robot where actions and background behaviours are executed, URBI receives and processes events and controls motors every 35ms. Telemetry data (sensor values) is sent over wireless to a personal computer approximately every 80-120ms. Reception of each frame of data defines a *timestep*, so the time between timesteps varies and is approximately 80-120ms. Video images were received from the robot head camera approximately every 400ms, however visual sensors were computed at the rate of the sensor data frame using the most recent image from the camera. Experiences were formed from data streams from 33 internal sensors (including proprioceptive motor positions and infrared distance measurements, see Appendix A) and 9 sensors formed from average pixel values in a 3×3 grid over the image.

URBI also provides a ball-detection algorithm tuned to the "pink ball" that is shipped with the Aibo robot, and this was used for determining the path for evaluation of suitable matching of experiences. The metric space creation and prediction was implemented in Java and ran on-line in real-time.

Figure 5.6: *Sony Aibo ERS-7, and Pink Ball*

The robot was stationary in a "sitting" position, with the head pointed forward (Figure 5.6). A pink ball was moved in the air in view of the robot's head camera at a distance of approximately 30cm. No particular effort was made to "sanitize"

Table 5.2: Path Prediction Experiment - Part 1: Sequences of Movements

Start TS	End TS	Movement Type	Iterations
1	210	Ball in visual field	
211	323	Horizontal, Right to Left	2 full
324	671	Circular, Clockwise	6 $\frac{1}{2}$
680	969	Vertical, Top to Bottom	4 $\frac{1}{2}$
970	1032	Horizontal, Left to Right	1 full

the environment to aid ball-detection against the background (a normal "office" environment with a louvred window in view). Thus, it is likely that other items in the environment provided potentially useful information about any interaction.

The ball was moved either vertically, horizontally, or in a circle. Simple, smooth motions of the ball were chosen so that the data would be amenable to analysis based on the type of motion. Also, discontinuous and fast motions were avoided due to the relatively slow rate of image capture (approximately 10 frames/second).[4] The ball was moved such that the time for the ball to describe a circle (or to move horizontally or vertically for a complete cycle) was 6-7 seconds, Thus the horizon length was shorter than, but of the same order of magnitude as, a single cycle of the repeated behaviour and the experiences would comprise approximately a half of a cycle.

The horizon length of the experiences was $h = 40$ timesteps or approximately 3400ms (a single timestep was approximately 85ms long). The data was quantized into 5 bins in the probability distribution estimation algorithm. Experiences were created only every $G = 4$ timesteps, where G is referred to as the *granularity.* A value of $G = 4$ was chosen to reduce the number of experiences that would be generated to a rate that would allow for real-time processing of the experience data on a typical office desktop computer.

[4]However, there is no reason to assume that more complex or discontinuous motions would not provide appropriate experiences for the history of a robot.

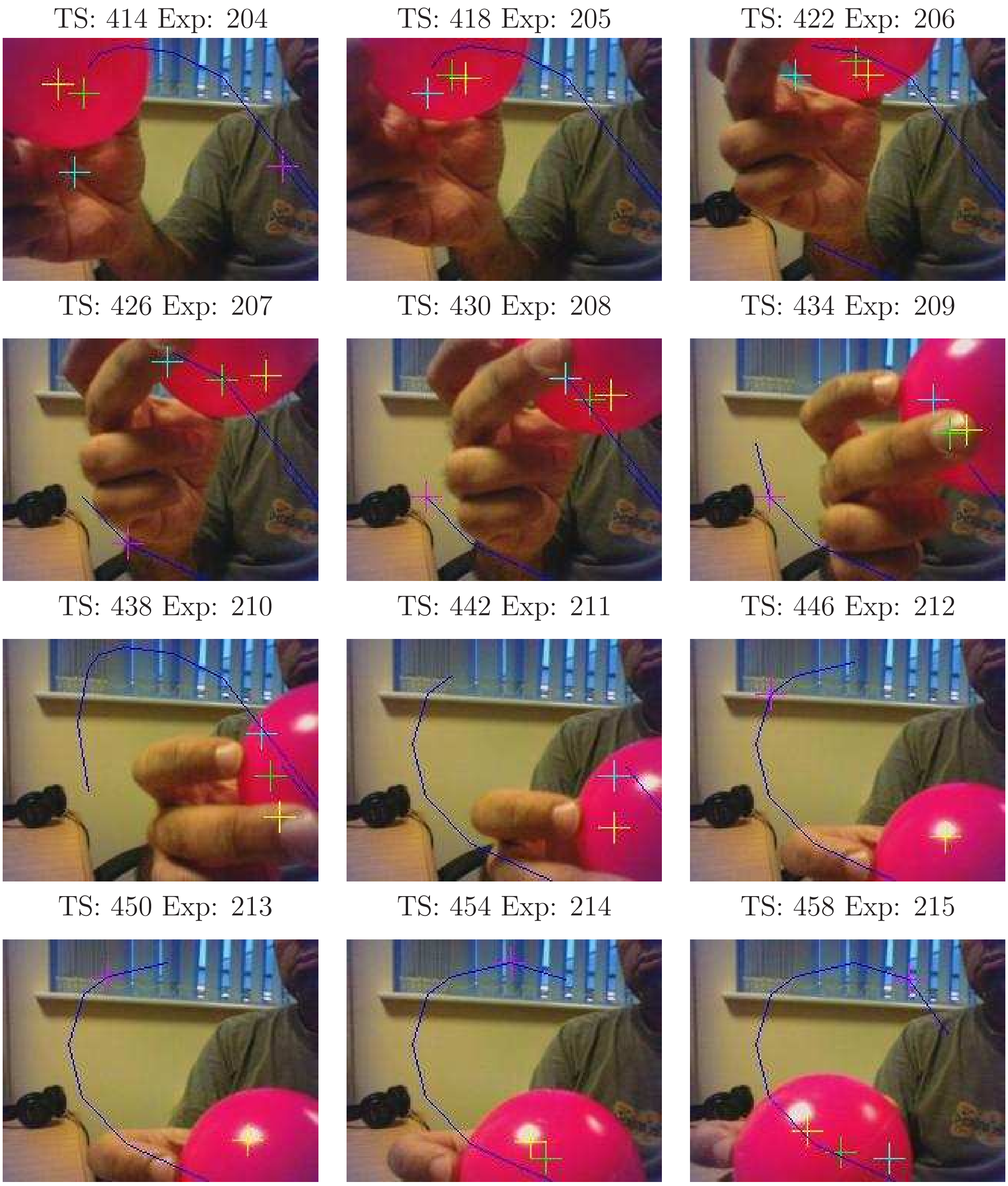


Figure 5.7: *Series of 12 consecutive images from the Aibo camera showing ball path prediction using a sensorimotor experience space.* The robot does not move its head in this sequence. Images are sequential left to right and top to bottom. The sequence lasts approx. 4 seconds (44 timesteps or 12 experiences) and is taken after 37 seconds of activity. The line shows the path prediction for 10 experiences ahead. The crosses are from various methods for ball detection, only one of these was actually used as sensory input. $h = 40$, $Q = 5$, Experience granularity $G = 4$ timesteps. One image shown per experience.

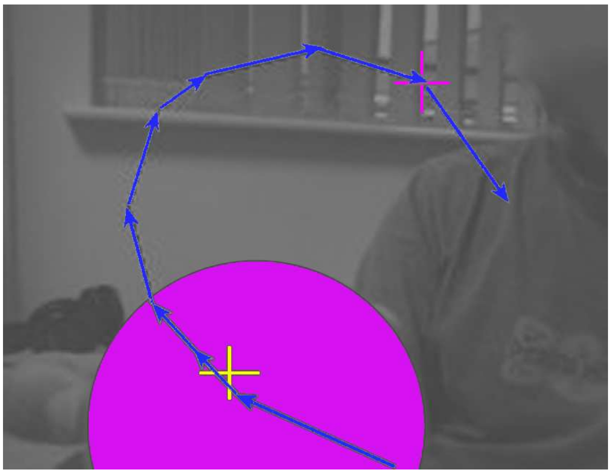


Figure 5.8: *Single image from the Aibo camera taken during ball prediction experiment.* The predicted path has been highlighted with arrows, starting from the position of the ball during the matched experience, and ending with the position of the ball during the 10th experience after the matched one. The lower cross-hair is detected ball position, the upper cross-hair is predicted ball position.

Results and Analysis (SPPEXP)

Figure 5.7 shows a sequence of images from one trial with one image shown per experience. The sequence shown in Figure 5.7 lasts just over 4 seconds and consists of approximately 50 timesteps (1 timestep $\sim$ 85ms) and 12 experiences (experience granularity $G = 4$ timesteps). There were 112 overlapping experiences (about 39 seconds of activity) before those shown, during which the ball was moved from left to right four times and in a circle once (see Table 5.2). Each image shows the robot's camera view during an experience with the predicted path overlaid (at run-time). For clarity a single image from the sequence is reproduced in Figure 5.8 with the position of the ball and the predicted path highlighted.

In the sequence shown and others, the robot required very few examples of a sequence (usually one) before the appropriately predictive experience could be located. This demonstrates that the information distance measure is capable of placing subjectively similar experiences (to an external observer) near to each other in the experience space (of the agent). However, it was found that while the path of the ball could be predicted fairly well early on in the sequence, later on, as the choice of experiences grew, the candidate experience chosen was not always the most appropriate.

Occasionally subjectively inappropriate experiences were matched. As an example, consider the seventh image in Figure 5.7 (Experience 210), here the predicted path inferred from the sequence of experiences following the candidate experience corresponds to the half circle that the ball has just been through (rather than the half-circle it is just about to go through, as in the other images). The candidate experience chosen is informationally close to another experience half a cycle back in time that may have been more appropriate. These two possible experiences that could have been matched correspond to motions of the ball from opposite sides of a circle. As the experience distance measure is the sum of information distances between variables, then a symmetric error such as this is likely, especially as phase-shifted periodic variables can have a small or zero[5] information distance.

This particular test scenario does not make use of the motor sensors of the robot in constructing, and therefore matching experiences. In the next experiment, this is addressed.

5.4.2 Interactive Path Prediction Experiment (IPPEXP)

In this second experiment the robot follows the motion of the ball, moved in front of it, by using a simple reactive behaviour to adjust its head motors to attempt to centre the ball in its field of vision. This presents a very different situation to the previous experiment in that, in addition to the visually derived sensors, there is also information about the experience of the robot in its own proprioceptive sense of its movement arising through interaction with the environment. The robot, as before, continually builds a metric space of experiences from its ongoing sensorimotor experience, including its own proprioceptive sense of movement arising through interaction with the environment. Experiences temporally following the historically closest experience then provide a model for anticipation

[5]Variables that have a zero information distance are *recoding equivalent* and are not necessarily identical (see Crutchfield, 1990).

Table 5.3: Path Prediction Experiment - Part 2: Sequences of Movements

Start TS	End TS	Movement Type	Iterations
91	185	Horizontal, Left to Right	2 full
201	272	Vertical movements, Top to Bottom	2 full
283	361	Horizontal, Right to Left	1 full
376	453	Vertical, Top to Bottom	2 full
463	534	Horizontal, Right to Left	1 full
548	593	Vertical, Top to Bottom	1 full
607	852	Circular, Clockwise	4 full
866	929	Vertical, Bottom to Top	2 full

of future experience. How good this model is depends on both the predictability and consistency of the environmental interaction as well as how "good" the historical matching is. Thus, the analysis of the experiment focuses on measuring how well matched the historical experience is to the current one. Note that predicting the trajectory of the tracked object corresponds to prospection regarding part of a future temporally extended interval of sensorimotor experience.

Implementation and Experimental Setup (IPPEXP)

The implementation and experimental setup are as for SPPEXP. In addition the robot executes a continuous reactive behaviour to follow the motion of a ball with its head. The algorithm is simple, making appropriate incremental adjustments to the neck, headTilt and headPan motors (see Table A.1), such that the position of the ball is brought closer to the centre. In this experiment, the horizon, binning and experience creation granularity are set as follows: $h = 20$, $Q = 10$ and $G = 1$.

The full interaction sequence lasted 965 timesteps ($\sim$ 84 seconds) constituting 945 experiences of horizon length 20. The movements of the ball consisted of a number of horizontal and vertical movements, and a number of clockwise circles;

see Table 5.3. Path prediction in this experiment operates in the same way as for SPPEXP, in that the predicted path is that traced by experiences subsequent to the most "similar" (*i.e.* closest in terms of distance) previous experience. However, in these results I focus on measuring how well the path traced during the current experience matches the path traced during the selected nearest experience, as this is a good indicator of how accurate the predicted path might be.

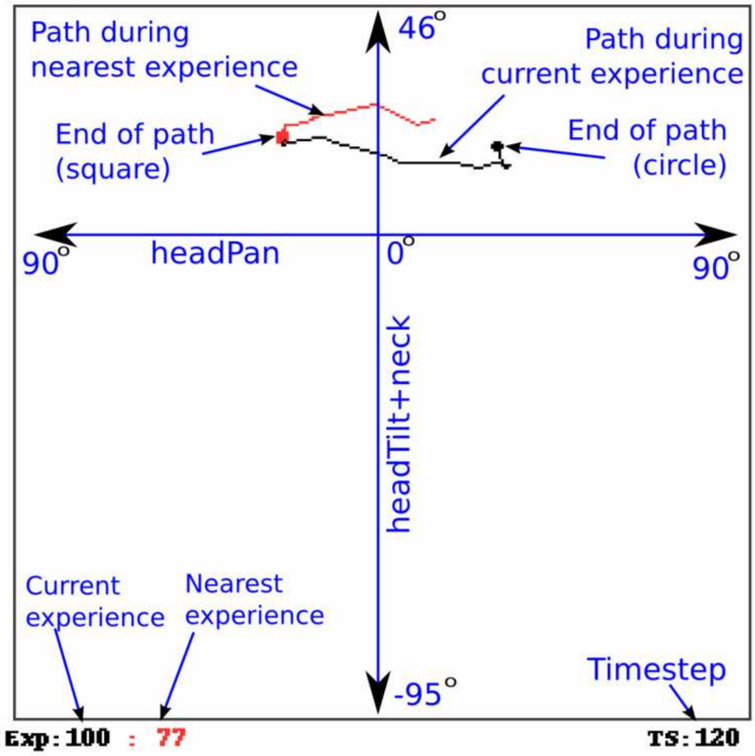


Figure 5.9: *Key to Ball Path Diagrams.* The diagram shows the parts of the ball path diagrams used to visually analyse the traces of the ball in a neck-centred coordinate system derived from motor positions. This serves as a key to Figures 5.12 and 5.13.

Visualizing Ball Path: The robot follows the motion of the ball with its head, so it is not possible to directly plot the path of the ball in terms of the camera images. Instead, it is possible to plot the direction in which the head is pointed estimated from three motors contributing to head motion. The path is plotted in two dimensions with the coordinates given by:

$$(x, y) = (W \times headPan, H \times (headTilt + neck)/2)$$

where W and H are the image width and height, and $headPan, headTilt$ and $neck$ are the motor values at any instant normalized into the range $(0, 1)$. See the explanatory diagram of Figure 5.9. Note that the plots are created for analysis of the experiments, and this abstraction of the sensorimotor flow is *not* available to the robot. Instead it allows an external observer to gain insight into what the robot 'expects' will happen in an interval of the near future based on its own previous experiences, and how accurate these expectations are (again to an external observer).

Error Measurements: Two different measurements of path error were used. The first measured the sum of the Euclidean distance between each corresponding point of the paths. The second calculated a vector direction for each path and returned the angular difference in radians between the vectors as the error.

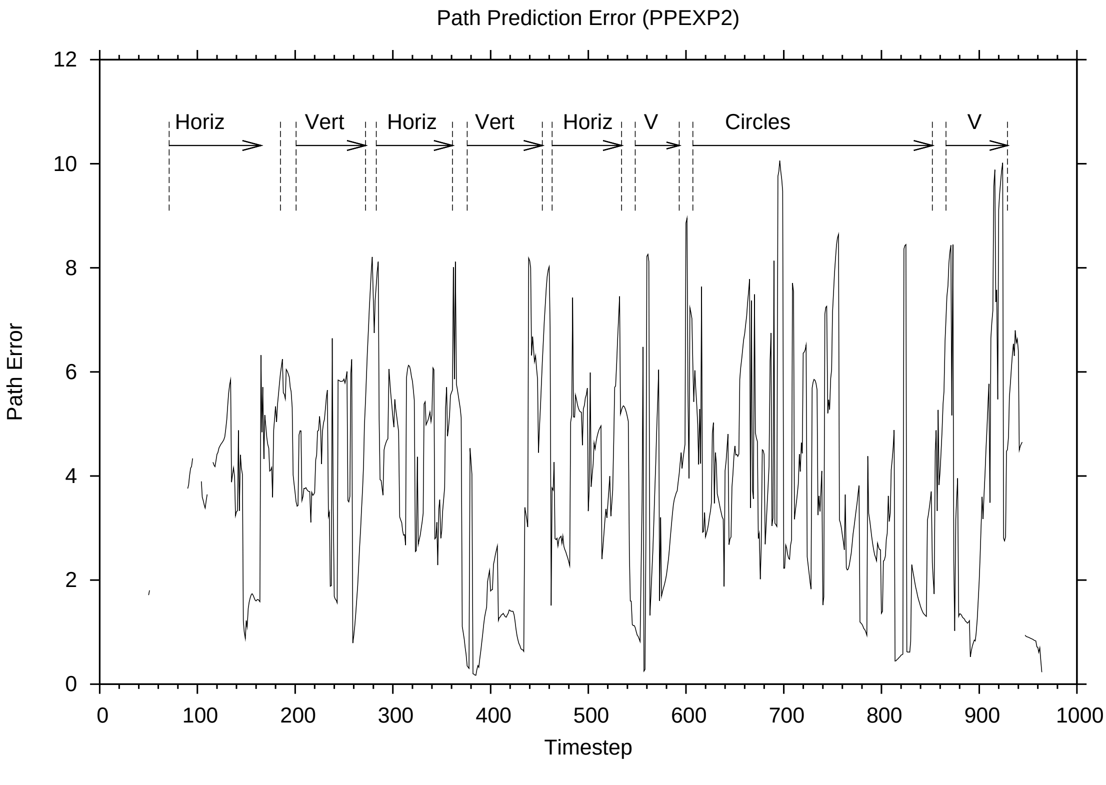


Figure 5.10: *Euclidean Distance (Error) Between Current and Nearest Experience Path Traces.* Graph shows the difference (error) between the path of the ball during the current experience and the path during the closest previous experience. The top part of the graph shows the behaviour (See Table 5.3). The *error* in this case is the sum of the Euclidean distance between corresponding points. Temporal horizon $h = 20$, number of bins $Q = 5$.

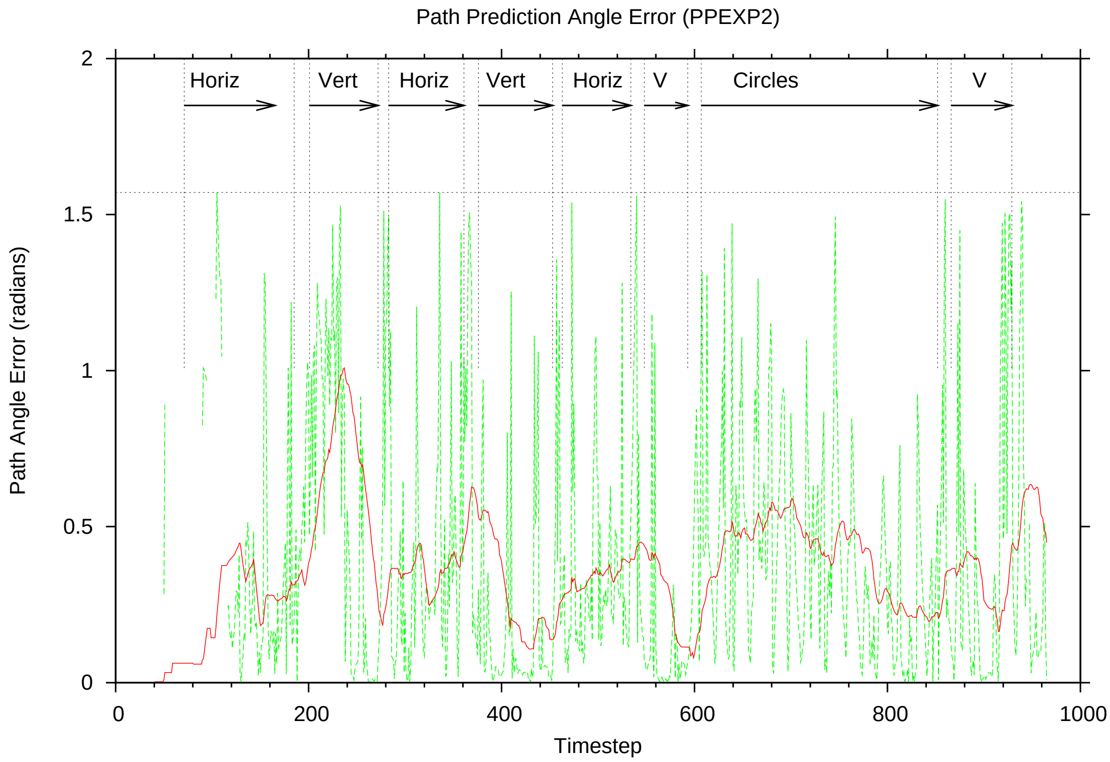


Figure 5.11: *Angle error and its running average (over the last 40 timesteps) between the current and nearest previous experience path traces.* The graph (*red solid line*) shows the error reducing, on average, *within* a given behaviour sequence. The top part of the graph shows the behaviour (See Table 5.3). The *angle error* (*green dashed line*) is the difference in radians between the vector directions of each path. For errors$> \pi/2$, $\pi - error$ is shown (reflection about $\pi/2$). Temporal horizon $h = 20$, number of bins $Q = 5$.

Results and Analysis (IPPEXP)

Figures 5.10 and 5.11 show, using different error estimations, the error between the current path and the path corresponding to the nearest previous experience in terms of information distance. Figures 5.12 and 5.13 show traces of the paths from experiences in regions where horizontal and vertical movements were taking place. As can be seen from the traces, which are selected from regular intervals, it is often the case that the paths are similar and so the experiences are well matched. However, the objective measure of error indicates that the actual path is not exactly the same. This is to be expected as there do not exist any *precisely* identical experiences in a real situation.

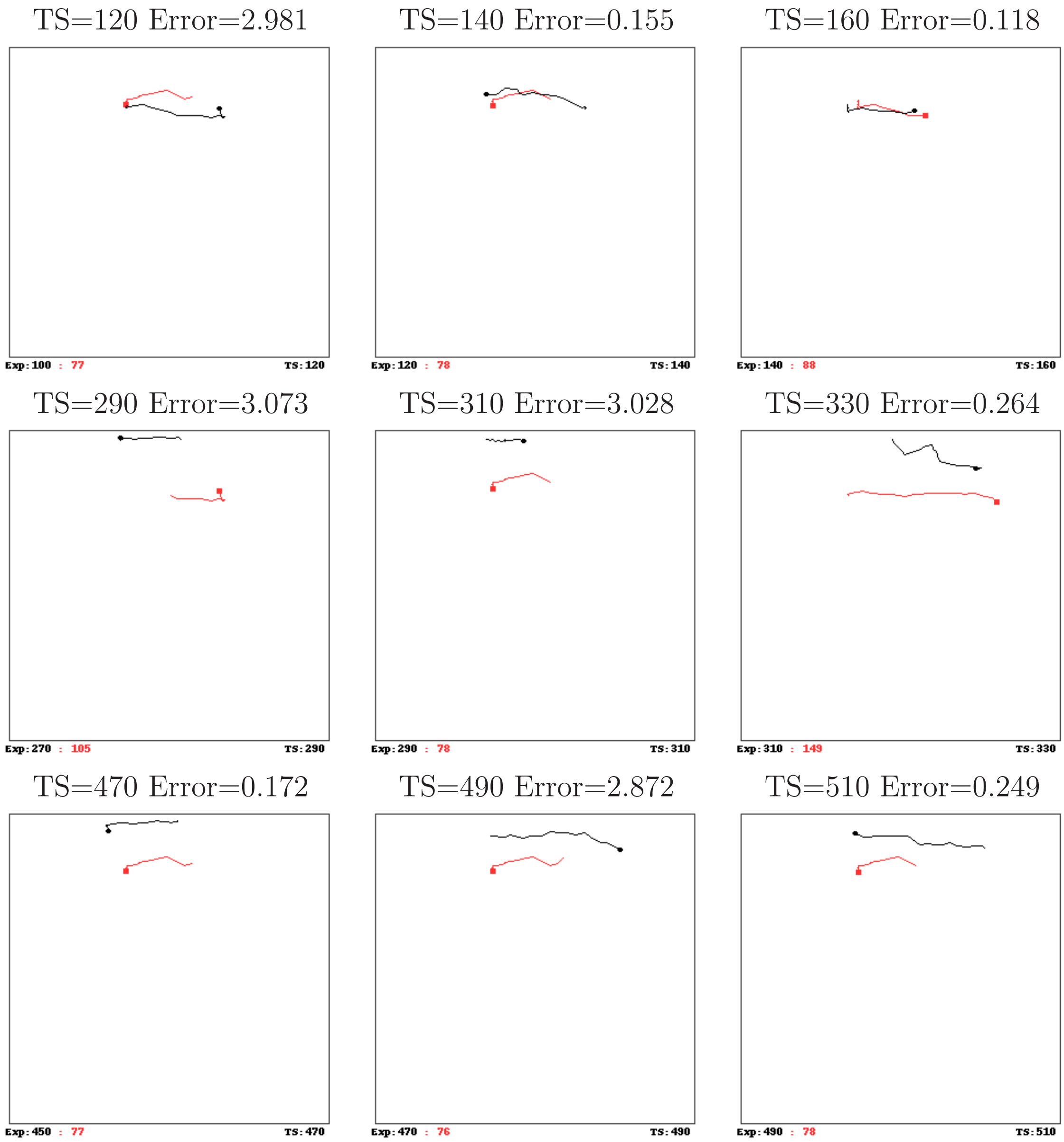


Figure 5.12: *Head Movement Traces and Matched Path.* A selection of path traces from horizontal head movements. Each diagram shows the path of the ball, as determined by robot head movements, for both the current experience at that timestep (dark line, circle end) and for the matched (nearest previous) experience (red line, square end). Path direction indicated by circle/square at the end of the path. (See Figure 5.9). The closeness of the paths is measured by determining the vector error between the path directions. Images are from evenly spaced timesteps from three separate horizontal movement regions. $h = 20$, $Q = 5$, path length$= 20$.

The error graphs also show that the opposite direction path is regularly matched

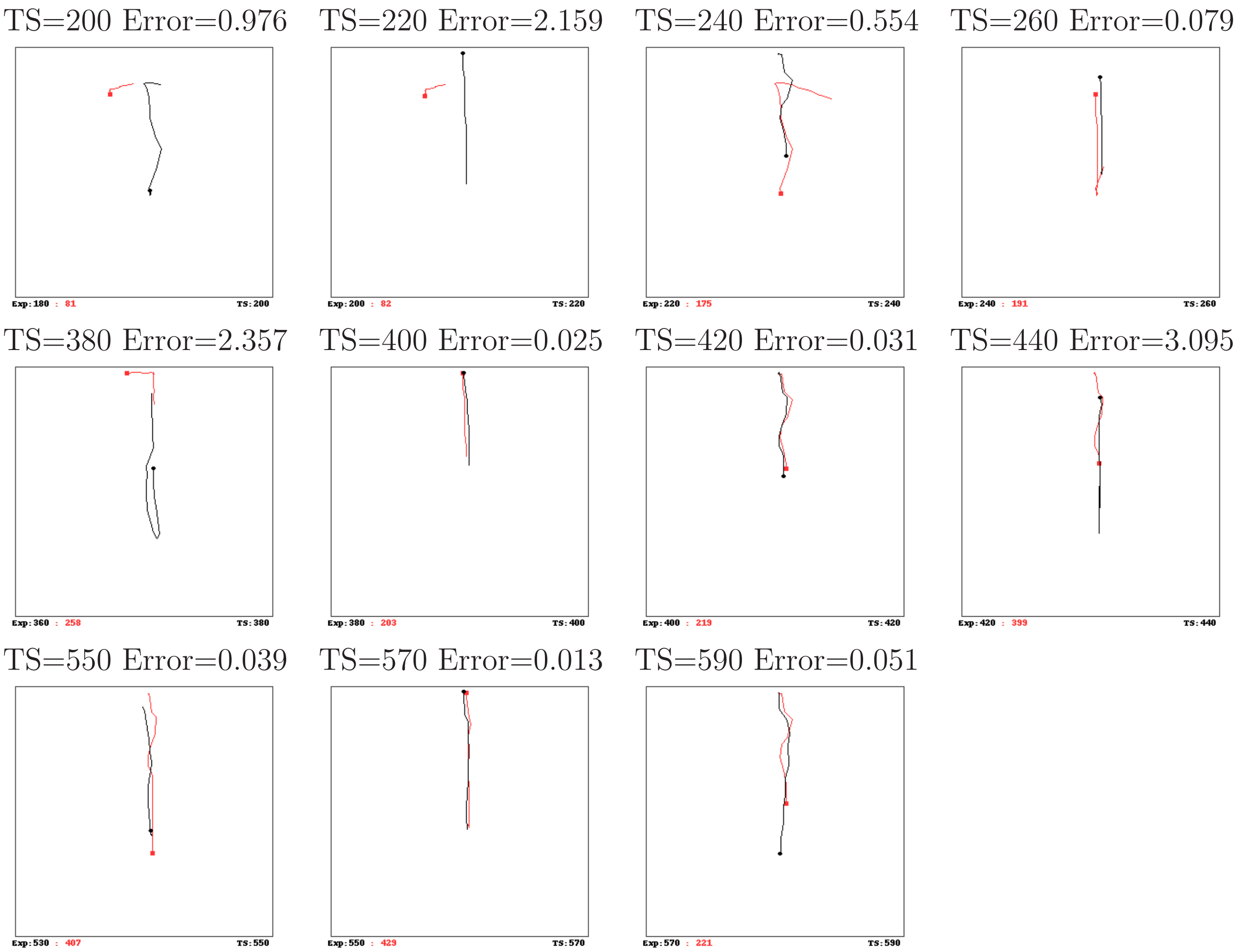


Figure 5.13: *Head Movement Traces and Matched Path.* A selection of path traces from vertical head movements. Each diagram shows the path of the ball, as determined by robot head movements, for both the current experience at that timestep (dark line, circle end) and for the matched (nearest previous) experience (red line, square end). Path direction indicated by circle/square at the end of the path. (See Figure5.9). The closeness of the paths is measured by determining the vector error between the path directions. Images are from evenly spaced timesteps from three separate vertical movement regions. $h = 20$, $Q = 5$, path length$= 20$.

as was the case in the first experiment (SPPEXP). As the sensors are not biased left or right, and the experience distance measure is the sum of information distances between variables, then a symmetric error such as this is likely. Indeed, such experiences are *informationally* very close to their 'opposites'. Out-of-phase periodic variables can have a small or zero[6] information distance.

[6]Variables that have a zero information distance are *recoding equivalent* and are not necessarily identical (see Crutchfield, 1990).

Table 5.4: Improvement of Experience Matching Over Time

Type	Iteration	Number $< \pi/4$	Total Number	Percentage $< \pi/4$
HORIZ	1	0	41	0.0%
HORIZ	2	27	73	37.0%
HORIZ	3	25	75	33.3%
HORIZ	4	27	72	37.5%
VERT	1	0	34	0.0%
VERT	2	8	51	15.7%
VERT	3	15	30	50.0%
VERT	4	42	61	68.9%
VERT	5	32	52	61.5%
VERT	6	27	49	55.1%
CIRCLE	1	9	65	13.8%
CIRCLE	2	13	54	24.1%
CIRCLE	3	27	66	40.9%
CIRCLE	4	31	63	49.2%

In terms of angle, the error is less than $\pi/4$ (*i.e.* closer to parallel than orthogonal) 55.13% of the time and is greater than $3\pi/2$ (*i.e.* closer to opposite than orthogonal) 29.21% of the time. This indicates that the path and therefore the experience is generally well matched, however due to the nature of the measure, experiences from the opposite phase in a cycle are often selected. It is interesting to note the opposite phase corresponds to time-reversed motion, and that the present metric relies on probability distributions constructed from sensorimotor flow and that these distributions do not encode the directionality of time.

Examining the progression of the error over time in these data, one would expect to see an improvement as the same kinds of behavioural interaction are re-experienced. How the matching of experiences improves over time is examined, referring to Table 5.4. During the horizontal motions after one full cycle, 37% of experiences can be matched to similar ones in the history. Vertical motions show that the success rate peaks at 68.9% with the 4th presentation, with a slight drop in success rate thereafter. The Circle movements also show marked improvement as experience grows. The initial 13.8% success rate of the very first circular motion reflects the fact that parts of the circular motion are being matched with previous horizontal and vertical experiences, with some limited success, even before any such motions had been observed.

The reason for the slight drop in rate of success for vertical motions as more experiences are added is not immediately clear and would be an important area to explore in future work. However, it is likely that a combination of factors contributed to the fall in success rate seen in the vertical motion sequences. Firstly, the repeatability of paths generated by human motion probably deteriorated over time so that later motions became less and less like the earlier motions. Secondly, a situation where there are a only a few experiences that are similar to the target experience would naturally result in a higher matching success than where there are many examples of varying quality of similarity. Thus, as more examples of varying quality are presented, it is likely that a poorer match (in terms of the objective angle error measure) may be chosen more often.

5.5 Chapter Summary

This chapter has described a mathematically rigorous formulation of temporally extended robotic sensorimotor experience and measures between experiences over time. The experience metric in particular is important for describing the changing nature of the robot's interaction with its environment over time. Having a rigorous and consistent measure between experiences is regarded as a step towards being able to directly drive development and learning in the robot by accumulation and organization of experience.

The construction and use of experience metrics for the comparison of robot behaviour is novel and demonstrates achievement of a degree of temporally extended prospection by an embodied agent, based on its raw sensorimotor experience. The experience metric was first described in (Mirza, Nehaniv, Dautenhahn and te Boekhorst, 2005*b*) and with mathematical discussion of the metric properties along with some alternative metrics on experience in (Nehaniv, 2005). As mentioned in Section 5.2 and 2.5, an operational formulation of experience (but not of the metric) was previously described in (Oates et al., 2000). A measure of distance between experiences was described there that used the area between time-warped experience curves. The fact that independent research groups both developed essentially the same notion operationalizing an agent-centred definition of experience suggests that this definition is a natural one.

Experiments were described that use fairly large numbers of robotic sensors to describe robotic experience such that a simple sort of prediction can be achieved by the matching of present experience with experiences in the history and extrapolating forward from the matched past experience. It was found that *proximity in terms of experience metric corresponds well with an external observer's notion of similarity of experience.*

The sensorimotor variables were treated by the autonomous robot in an uninterpreted "agnostic" manner, that is, no sensor is regarded as being different from any another or special in any way, in terms of finding close experiences. This performance was achieved despite many of the sensors not providing any seemingly useful information about the current experience as could be seen in the case of the first, static, ball path prediction experiment (Section 5.4.1). In the second, interactive ball path prediction experiment (Section 5.4.2), proprioceptive motor experience was important in this experiment in determining the experience and matching it to the appropriate past experience.

The capability of the experience metric to find suitable matching experiences was found to increase as more examples of a particular type of behaviour were presented. Table 5.4 shows that this reduces somewhat for the VERTICAL motion as more examples were presented (although this may be due to the quality of repeated motions presented as discussed). Further experiments that both control the generated ball motion and have more examples of each motion type may be necessary to investigate this effect.

Another important aspect of the experience metric is that it appears to confuse a behaviour with its 'opposite' (phase-shifted or time-reversed counterparts), as these are informationally nearly identical. This can be seen clearly in both the simple and interactive ball-path prediction experiments as opposite direction of path.

Chapter 6

Construction of Metric Spaces and Emergent Classes of Experience

6.1 Introduction

In this chapter I consider the practical aspects of constructing a metric space of experience on-line for an embodied agent. These are important as, for a metric space to be useful, it should be able to be constructed and used as the experiences arrive, and be able to continue working for the developmental lifetime of the agent. The goal, therefore, is that computation characteristics should allow for "real-time" operation within a finitely bounded storage. Achieving this is a great challenge as there is an enormous amount of data arriving at the sensory surfaces of any embodied agent, however, in pursuing this challenge there are also great advantages to be gained. Of primary importance is the possibility of grounded category formation, which can lead to important developmental advances for the agent.

I begin by discussing the scalability over time of the interaction history architecture in terms of computation time and memory requirement. Specifically the time to place a new experience in a metric space (for the purpose of returning a list of nearest neighbours) is investigated. Secondly I examine the storage of the metric space of experiences, leading onto the requirement for "forgetting" and "merging" of experiences being an integral part of an experience space. Of course as experiences are merged, so it becomes possible to consider emergent categories as groups of experiences.

6.2 Incremental Construction of Metric Spaces of Experience

As the agent acts in the environment and experiences are collected using estimation of entropies from a time window of binned sensor readings, they can be "placed" in a metric space by finding all distances between the new experiences and all experiences already in the space. This is incremental construction as the experience distances are not all calculated at the same time, but are only calculated as required upon arrival of a new experience. By far the most computationally expensive task in this process is the calculation of experience distance between any two experiences. Clearly, as each time a new experience is placed, there is one more experience to compare than the previous time, then the time to place an experience increases linearly with the number of experiences already in the space.

Given that new experiences arrive regularly, it is inevitable that as the number of experiences in the space grows it will not be possible to place an experience in the metric space before another one is available for processing. Further, given that the time to make a single comparison is constant, the only way to reduce computation time is to reduce the number of comparisons. This can be done either by reducing the number of items to compare (see Section 6.3) or by not explicitly computing all distances (see Section 6.2.1).

6.2.1 Reducing Comparisons

One way to reduce required comparisons is to use the distances between experiences in the space to infer distances to any new experience. However, as the metric space of experiences is a non-euclidean space, then this becomes more difficult.

Note, however, that knowing the distance to all other experiences is not necessary for the correct operation of the interaction history architecture. It is only necessary to know the *nearest* neighbours; *i.e.* the nearest N experiences, or all experiences within a "ball" of radius r.

Finding Nearest Neighbours

Say one is interested in finding all nearest neighbours of an experience E^{new} within a "ball" of radius r, then the triangle inequality can be employed to reduce the number of distances that need to be measured. Specifically:

Theorem 6.1 *Given an experience E^k that is distance $d(E^{new}, E^k) \leq r$ from E^{new}, then any neighbours of E^k that are further away than $2r$ are not within distance r of E^{new}.*

Proof 6.1 *Consider 2 experiences Y, Z near X; near is defined to mean within distance r, thus: $d(X, Y) \leq r$ and $d(X, Z) \leq r$. Then, by the triangle inequality ($d(Y, Z) \leq d(X, Y) + d(X, Z)$), $d(Y, Z) \leq 2r$. Therefore, if any 2 experiences are further apart than $2r$, then they cannot both be within radius r of any one particular experience.*

This fact can be used discard experiences from consideration when finding nearest neighbours within a specified radius. Of course, this requires first finding an experience with radius r of the new experience. One approach to this problem is to simply randomly sample the experience space until one is found. Other strategies exist, for example: using the continuous nature of the environment to start the search for near experiences (in terms of information distance) with those experiences near in terms of time.

Algorithm 6.1: `B2R_NN`: Populate Metric Space Distances for Nearest Neighbours

Input: r, radius
Input: E^{new}, new experience
Output: $newDistances$, empty list

$toTestList \Leftarrow$ all experiences in metric space
while *toTestList is not empty* **do**
    remove a random experience from $toTestList$, assign to E^K
    calculate $d(E^{new}, E^k)$, add to $newDistances$
    if $D \leq R$ **then**
        remove all experiences further than $2r$ from E^k in $toTestList$
    end
end

An algorithm to find the nearest neighbours of a new experience from a metric space of experiences is given in Algorithm 6.1 which guarantees that all experiences within r of the new experience E^k will be in the list $newDistances$. There may also be some other experiences not with r in that list which will have been checked as a consequence of the random sampling.

An important issue is that any strategy that does not fully populate all distances in a metric space is potentially degenerative. That is, when another experience arrives, it may not be possible to make the same guarantees as the existing metric space is not fully populated. In practical use however, the algorithm given should still find all neighbours as it excludes only experiences which clearly do not fall within radius r. The result instead is that potentially more comparisons will have to be made. This however in turn results in a better populated space.

A question remains: by how much this might reduce the space of experience to be searched? The answer is largely dependant on r (as shown in the tests below, see Section 6.2.1) and on the nature of the space. At one extreme, if experiences are clustered tightly together with no experience further than $2r$ from any other, then all experiences must be searched. Due to the nature of the algorithm, the computation time would actually be greater than if all experiences were checked in turn. At the other extreme, if the radius was smaller than any distance between two experiences, then once again all experiences would have to be checked because no near neighbour would be found.

Happily, the situation is likely to be somewhere between the two. If the experiences are clustered around many centres further apart than $2r$, or are evenly spaced with the minimum distance much less than r but the maximum distance much greater than $2r$, then it is likely that a near experience will be found fairly quickly and consequently, many experiences will be discarded, reducing the computation time significantly.

Finding a suitable radius r

With the strategy given above, an important question is: what value should r take? This clearly depends on the nature of the space and how many nearest neighbours are needed. (The latter quantity is important in the choice of next action within the interaction history architecture action selection presented in Section 7.) Thus, r is likely to change as the robot interacts in the environment and so should be adaptive.

A strategy to adapt r suitably to the current metric space is to instead take K, the number of nearest neighbours desired, as a reference point. Starting with r at an initial value, for every new experience, find all neighbours within radius r. If this number is greater than K, adjust r downwards and *visa-versa*.

Test of `B2R_NN` algorithm in artificial and real metric spaces

To quantify the computational saving that can be achieved by the `B2R_NN` algorithm, two tests were conducted. Firstly, an artificial euclidean metric space with evenly spaced random points was used to investigate the relationship between the density of the points in the space, N, and the radius, r. Secondly, a real metric space of experience taken from an Aibo interacting with a human partner was used to investigate the effect of varying the radius, r.

In figure 6.2 the results from the artificial space are shown. The metric space was 3-dimensional euclidean and contained randomly placed points. The maximum possible distance in the space was 17.32 *(no units)*, with an observed average

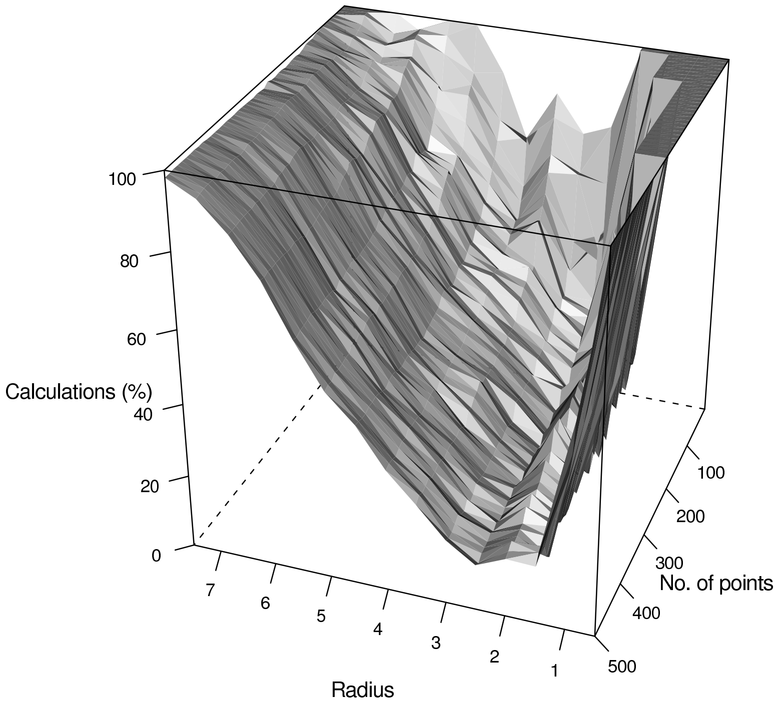


Figure 6.1: Graphic showing relationship between r and N in determining the number of calculations made by the `B2R_NN` algorithm in an artificial metric space.

distance between any 2 points of approx. 7.2 and a minimum distance to any neighbour between 2.3 and 0.7 depending on the number of points.

The results show that when the radius r is relatively small (in this case $r \leq 1.0$) then there is no or very little reduction in the number of calculations required to find the neighbours in a ball of radius r. As the radius increases, less than 20% of the calculations are needed, However, this saving of 80% is lessened as the radius grows until it eventually comes back down to 0. While these observations are true to some extent for any number of points, the certainty of gaining such a speed-up is increased with the density of points in the space.

Figure 6.3 shows the results when the algorithm was tested in a metric space that resulted from an Aibo interacting with a human partner. The Aibo variously looked at the partner's face, hid its face with it's forearm (peekaboo) and looked at the pink ball. The space had a total of 372 experiences in it. The distances for the 373^{rd} experience were pre-calculated for the purposes of the test, and used as a look-up table in the tests of the `B2R_NN` algorithm.

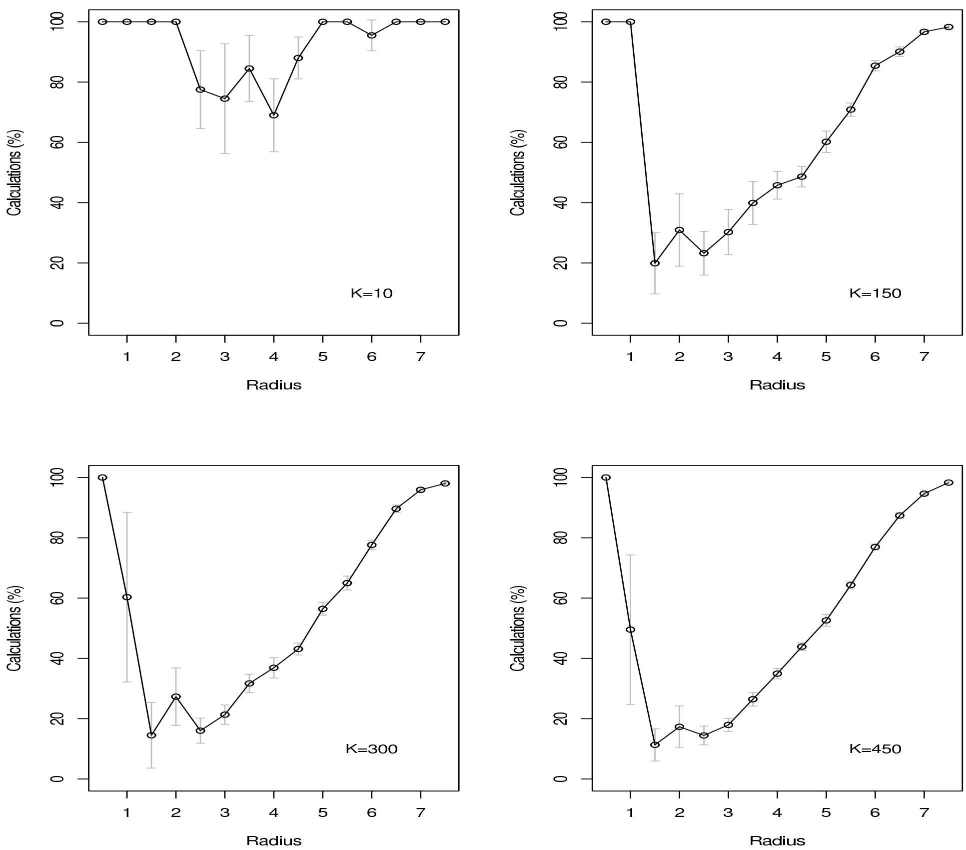


Figure 6.2: Graphs showing relationship between r and N in determining the number of calculations made by the B2R_NN algorithm in an artificial metric space for selected N. Each point is the mean of 20 runs, error bars show 1 *Std. Dev.*

A similar shaped curve is again observed indicating that, with a good choice of r, significant saving in number of calculations can be achieved.

6.3 Storage Requirements: Merging, Forgetting and Emergent Classes of Experience

Another strategy for reducing the number of computations of distance between experiences is to reduce the number of experiences in the space in the first place.

The memory storage required to maintain a experience space consists of: the storage of the experience[1], plus that of the metric space itself (*i.e.* distances), plus

[1]In storing an experience, all that is required are the binned values of the sensors, not the actual values of the sensors. In addition meta-information will be stored with the experience

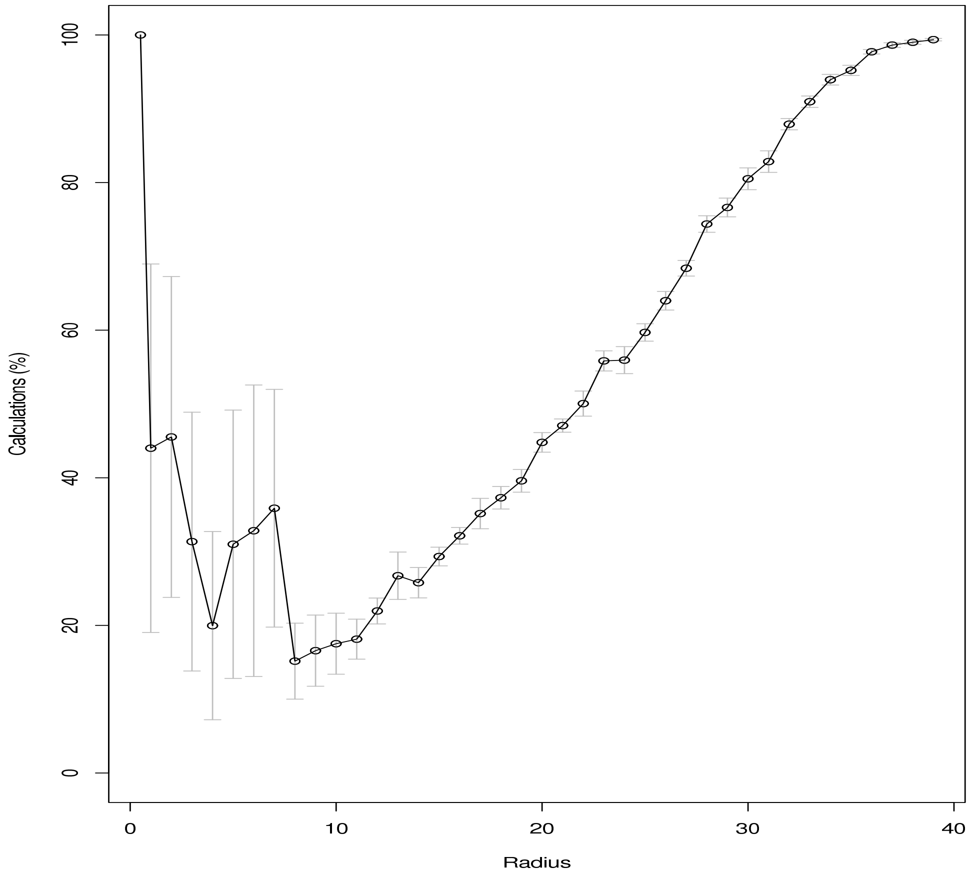


Figure 6.3: Graph showing effect of r in determining the number of calculations made by the `B2R_NN` algorithm in a real metric space taken from an Aibo interacting with a human partner. Each point is the mean of 20 runs, error bars show 1 *Std. Dev.*

constant factors. The storage of fixed length experiences grows linearly with the number of experiences. The non-constant storage of the distances increases faster. At any time it is proportional to P_2^N, the number of permutations of 2 items from N items, where N is the number of experiences. In terms of complexity this is order $O(n \log n)$.

Thus, it is not possible to store all experiences and all distances indefinitely for a metric space that is growing. At some point it will exceed the storage available. Also, many calculations in the space are dependant on the number of experiences and so computational complexity is also affected. Therefore, two strategies are examined that may reduce the number of experiences within a metric space as it

e.g. next action, quality, weight *etc.*See Section 7.2.1.

is growing: *forgetting* and *merging*. The later strategy is particularly important as it also leads to the emergence of grounded categories.

6.3.1 Forgetting

In terms of a metric space of experiences, *forgetting* corresponds to removing individual experiences from the space, including all meta-information and distances to other experiences. This is a useful way to reduce both computational complexity in maintaining the space as well as reducing storage requirement.

The question is: how should experiences be chosen for removal? Of course, it could be random, however, it seems to make more sense to base removal on some quality of the experience itself. For instance, *time*. *i.e.* how often the experience has been "accessed" or "used" or how long ago it was last "accessed". This would correspond to natural, intuitive ideas of forgetting. An alternative measure could be the *quality* of the experience in terms of reward signals. In this scheme, experiences that were neither "very good" nor "very bad" might be candidates for forgetting. In terms of the metric space of experiences, another measure might be how *isolated* an experience is from others.

6.3.2 Merging Experiences in a Growing Metric Space

This strategy is based on the idea that if two experiences are very similar, then they could potentially be treated as the same experience for the purposes of comparison with other experiences. Intuitively, this is what happens as we experience the world. As we engage in an activity that we do many times, such as drinking a mug of tea at our desk, we do not notice that it is similar to any one particular time we engaged in that activity in the past, only that it is similar to a generalized activity: *i.e.* past experiences have been merged into a single experience (for the purpose of comparison at least).

The general strategy is to replace two experiences in the space by a single experience that has features taken from one or other of the experiences or both.

Individual strategies are distinguished by how the two experiences are chosen, *e.g.* by using a threshold T^{merge}, and by what features of the experiences are retained or discarded. Two strategies are discussed: calculation of an intermediate experience; and, merging by deletion, where one of the experiences is deleted.

Calculating an Intermediate Experience

Merging two experiences E^a and E^b, and replacing them with one that is some-way between the two, can be considered as a problem of finding an *intermediate* experience E^{ab}, such that: $d(E^{ab}, E^a) \leq d(E^a, E^b) \wedge d(E^{ab}, E^b) \leq d(E^a, E^b)$.

Ideally the intermediate experience would be half-way between the two, *i.e.* $d(E^{ab}, E^a) = d(E^{ab}, E^b)$. This calculation is not mathematically straightforward due to the non-euclidean nature of the space and may take quite long to compute. One possibility is to find a combination of binned sensor readings that is approximately half the hamming distance between the two sets of values, however, this possibility is not explored further here.

Alternatively, one or other of the distances $d(E^{ab}, E^a)$ or $d(E^{ab}, E^b)$ can be zero, which amounts to keeping one of the experiences and removing the other. Clearly, however, the merged experience is closer to one experience than the other. This becomes less of a problem as $d(E^a, E^b)$ approaches zero.

Merging by deletion

Due to the difficulty of mathematically merging two experiences an alternative strategy is to remove one of the experiences entirely. This may not be satisfactory as that experience probably had important information that may be useful. The fact of its existence is one such, *i.e.* the fact that it occurred and was similar to other experiences gives a sense of *familiarity* and may be important in choosing a list of N nearest neighbours. Another important piece of information is the subsequent action that was taken after that particular experience, which may or may not have been different from the other experience. Finally, the distance information may also be important.

A modified strategy would be to remove one of experiences from the space, but retain other information such as number of merged experiences and subsequent actions with the remaining experience. This is in fact the preferred strategy in the Interaction History Architecture. See Section 7.2.5 and Algorithm 7.1 for further details.

An obvious choice for merging criteria is to merge any two experiences closer than a threshold T^{merge}. A fixed threshold can be used but that raises the problem of finding a suitable value. Alternatively it could be an adaptive threshold responding to some other criteria such as maximum number of experiences in the space. For the special case $T^{merge} = 0$ no information is lost in the merge of the sensorimotor experiences themselves.

With reference to the Interaction History Architecture (introduced in the next chapter) where the experiences in the metric space are augmented with other information such as reward feedback from environmental interaction, then an alternative way of choosing experiences to merge would be to use those other features[2] of the experiences as merging criteria. Of course, these other features could be combined with the distance threshold to refine the choice.

Distances to a merged experience are estimated by measuring the experience distance to the remaining experience in the merged pair.

Retaining Distances

An alternative to the complete removal of an experience from the metric space, is to delete only the sensorimotor experience data and retain only the existing distance information. This will result in a reduction of memory requirement while retaining important structural information about the metric space. The space would then contain *parent* experiences about which everthing is known, and *child* experiences having only distance and meta information. Any new experiences would only be able to be directly compared to parents and distances to child

[2]Candidates are *next action* and assigned *quality* (See Section 7.2.1). Interestingly, merging according to quality provides a powerful method of adapting the experience space to the changing feedback reward from the environment for any given emergent category of experience.

experiences only inferred.

This strategy has the advantage that only sensorimotor information is lost, and that a natural hierarchy within the metric space can easily be built. The disadvantage is in that the distances from child experiences cannot be known, and that the complexity and storage requirements are not reduced significantly.

6.3.3 Grounded Categories

A direct consequence of merging of experiences in the metric space is that natural categories are formed along with "representative" experiences of those emergent categories. The merged experiences can be thought of as a grounded representation of a class of experiences. Importantly, as experiences are directly associated with action, then the category is grounded not only in the sensory domain - which would leave it as an abstract representation without meaning - but also in how the agent responds to that class of experiences, closing the loop and grounding meaning too.

However, there are limitations to the kinds of categories formed by the type of merging discussed here. Firstly, categories cannot be split after they are formed, and they can only become larger. This leads to less and less resolution between experiences as development proceeds, whereas one might expect general categories to be refined with further experience. Secondly the resulting experience after a merge may not be representative of all the experiences that have been merged. This is also affected by the order of merging. If *new* experiences are always *merged into old* then experiences will cluster within a radius of T^{merge}, and so the "true" cluster centre can never be very far from the merged experience. However, if the *old* experiences are *merged into new* ones, then the cluster centre can be "dragged" arbitrarily far from the "true" cluster centre.

An approach to resolve this problem would be to continually calculate a true cluster centre and retain the closest experience to that as the new merged experience. Splitting of clusters, would however require the retention of more experiences within a single cluster. One implementation would be to delay merging. That is, to create cluster structure, for instance in a tree, and only merge (or split) depending on some further criteria of time, or depth of tree. A heuristic approach to this may be possible using retained distances and a parent/child experience tree structure discussed previously.

In (Weng et al., 1999), hierarchies of "brain states" are built automatically using a "classification and regression tree" that combines dimension reduction using principal components analysis and linear discriminant analysis. Merging is done on a group of neighbourhood states in the tree (*i.e.* not a pair), although the authors do not address the issue of splitting categories after they are merged.

6.4 Chapter Summary

An experience space that is manageable in terms of the resources available for computation is important in any practical implementation. Therefore this chapter explores techniques that can both reduce computation time to find the nearest neighbours of an experience in a metric space of experiences, and reduce the storage space required to hold a metric space. The first technique presented makes use of the metric nature of the space to reduce the number of calculations needed when searching for neighbours within a given distance of a given experience. The next approach uses both deletion of experiences in the space (forgetting) as well as merging of experiences that are a short distance apart. Merging and forgetting not only reduce storage space and computation time, but also provide a method whereby classes of experience, *i.e.* categories, may emerge through the natural relationships between experiences resulting from embodied interaction. Moreover, as experiences are merged and forgotten, the metric space will be continually changing and so be able to adapt and change in response to the new experiences of the agent.

Merging and forgetting are used in these ways in the full architecture for robot ontogeny based on experiential interaction histories described in the next chapter (7), and in the implementation in Chapter 9.

Chapter 7

Interaction History Architecture

7.1 Introduction

Referring back to the definition of an Interaction History given in Section 2.2, this chapter proceeds by addressing the challenge of "shaping current and future action" based on the history of interaction as embodied in the metric space of experience. Moreover, the challenge is not to merely control action, but also provide a framework upon which a robot can build its ontogenetic development through interaction with its environment.

This chapter builds on the properties of the metric space of experience established in the preceding chapters, and adds two aspects: an action or behaviour selection mechanism and a method of combining the robot experiences with information from the environment that enables the robot to judge success or failure of its actions. The Interaction History Architecture that is presented is then tested in a simple iterated scenario using a simulated wheeled robot ("The Road-Sign Problem") that is used to asses the ability of the architecture to select appropriate actions based on its past experience.

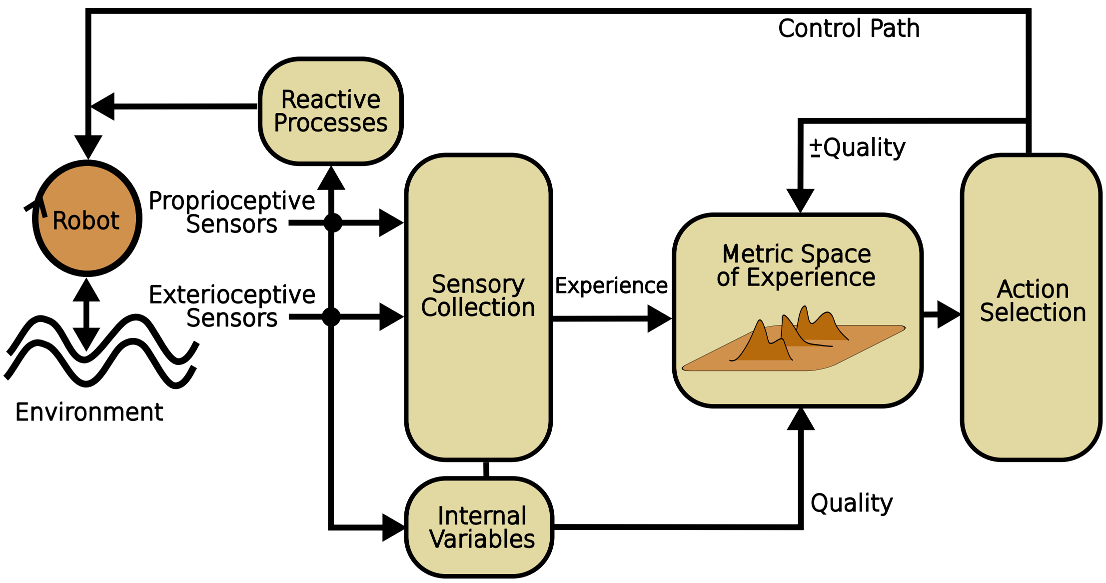


Figure 7.1: A Schematic of the Main Components of the Interaction History Architecture

7.2 An Interaction History Architecture

The Interaction History Architecture is shown schematically in Figure 7.2. The approach is as follows:

1. to continually gather sensorimotor data and find "suitable" episodes of sensorimotor experience in the history *near* (in terms of the experience metric) to the current episode;

2. depending on the course of subsequent experience, to choose from among actions that were executed when these episodes were previously encountered;

3. where no suitable experiences are found, to choose random actions.

There are two key aspects of this architecture. The first is the *metric space of experience* whereby new experiences appear as points in a growing and changing metric space. In this architecture the metric space is enhanced with *quality* information from the environment, internal drives or affective state. Each experience is also associated with actions executed during the experience. The second is the *action selection* system. This "closes the perception-action loop" and also closes an internal loop feeding back and modifying the experience space. The quality associated with each experience combined with proximity in the metric space is used to select experiences from the history and select actions associated with those experiences.

7.2.1 Metric Spaces of Experience

The metric space is constructed continuously as the robot experiences its environment. A new experience is created every *Granularity* G timesteps, and consists of Horizon h timesteps counting back from the current timestep. Where $h > G$ the experiences will overlap. Each sensor reading is quantized into Q evenly-sized bins. Each new quantized experience is compared to other experiences in order to determine its neighbours. This process, if all experiences are compared, results in a distance matrix between experiences which defines the structure of the metric space as it is experienced by an individual robot. The mechanisms for constructing the nearest neighbour list are examined in Chapter 6. A *quality* value is assigned to the quantized experience, determined by factors such as environmental reward/punishment, internal drive and affective state. The actual formula for calculation of *quality* is specific to the application and goal and can be a determining factor in the eventual behaviour and course of development, although it can be fairly general and thus applicable to a wide range of situations. Finally, the last action executed during the experience is also noted and stored with the quantized experience.

Thus the metric space of experience in the Interaction History Architecture, the *interaction history space*, can be described by the tuple $(\boldsymbol{\epsilon}, \boldsymbol{D}, \boldsymbol{q}, \boldsymbol{a})$, where $\boldsymbol{\epsilon}$ is a collection of quantized "experiences", $\boldsymbol{D}$ is the a matrix of distances between elements of $\boldsymbol{\epsilon}$, $\boldsymbol{q}$ is a vector of quality values and $\boldsymbol{a}$ a vector of actions. This description is extended for "clustered" experiences later in this chapter.

7.2.2 Action Selection

A simple mechanism is adopted for action selection whereby the robot can execute one of a number of "atomic" actions (or no action) at any timestep. This is seen as a tractable first-step, and a more sophisticated action or behaviour generation capability would allow for more open-ended development.

The actual action selected will either be a random selection of one of the atomic actions, or will be an action that was previously executed after an experience in the history that is *near* to the current episode. An advantage of this approach is that behaviour can be bootstrapped from early random activity, and later behaviour built on previous experience.

The process of action selection is as follows:

1. up to K *candidate experiences* from the experience space within a given information distance *radius*[1] r_0 of the current experience $E_{current}$ are initially selected;

2. these K experiences are ranked as $E_1, \ldots, E_K$ according to how close they are to $E_{current}$;

3. then, *sequentially*, experience E_i is chosen with probability a linear function of the *quality* of E_i until either an experience is chosen or the ranked list is exhausted;

4. if an experience is chosen from the candidate list, then the particular action that was executed following the chosen experience is then chosen as the action to be executed next, otherwise a random action is chosen.

The linear mapping from quality to probability ensures that, with small probability, the robot may still choose a random action as this may potentially help to discover new, more salient experiences. This has the advantage of emulating body-babbling, i.e. apparently random body movements that have the (hypothesized) purpose of learning the capabilities of the body in an environment (Meltzoff

[1]The radius can be fixed in this formulation, but, may instead be adapted on-line.

and Moore, 1997). Early in development, there are fewer, more widely spread experiences in the space, so random actions would be chosen more often. Later in development, it is more likely that an the action selected will come from past experience.

Roulette-Wheel Action Selection

In later implementations (including the T-Maze implementation described in Section 7.3), the process was improved to use a *roulette-wheel selection* from a probability list. The chance of random action selection is also represented in that list. The probabilities are calculated using a "gravitational model" where each experience is represented as a point mass a particular distance from the $E_{current}$. The probability of selecting an experience E_i from $E_1, \ldots, E_K$ is:

$$p_i = C_h \frac{m_i q_i}{D(E_{current}, E_i)^2} \tag{7.1}$$

where q_i is the *quality value* of E_i, m_i is the mass (*i.e.* how many experiences have been merged into this experience) and $D(E_{current}, E_i)$ is the experience distance. C_h is an optional quantity that is used to adjust for "horizon effect" when considering experiences of different horizon length together (see Section 7.3.3), and is given by

$$C_h = \frac{\sqrt{h}}{\sqrt{H_{max}}} \tag{7.2}$$

The chance of random is added to the list as:

$$p_0 = \frac{\sum_{i=1}^{K} p_i}{(r_{max}/\tau)^2} \tag{7.3}$$

where r_{max} is the radius of the ball that includes the ranked experiences and τ is a *temperature* factor, that controls the chance of random action selection.

Then the weighting on the "roulette wheel" is given by:

$$w_i = \frac{p_i}{\sum_{i=0}^{K} p_i} \tag{7.4}$$

7.2.3 Update of Environmental Reward

Each experience in the interaction history space is associated with a quality value q, see Section 7.2.1. This value has bearing on the selection of the experience, and in turn on the action-selection process. The quality value is intended to reflect how useful the experience is in terms of positive or negative environmental feedback, and is derived directly from the internal reward function or an external reward measured by the robot's sensors.

In the simplest case, the immediate (instantaneous) reward received from the environment is associated with the current experience. An alternative scheme is for the quality associated with an experience to be dependent not only on the current reward, but also on the future reward. The *future reward* for an experience $E_{t,h}$ for some given horizon h_{future} is a function $\mathcal{F}()$ on all reward values received for h_{future} timesteps after time t. Of course, this value cannot be known completely until at least h_{future} timesteps have passed, but it is estimated until that point. Two functions have been used in the implementations in this thesis. The first, $\mathcal{F}_{min\_max}()$, returns the most proximal maximum or minimum reward. The second, $\mathcal{F}_{max}$ simply returns the maximum reward over the horizon.

7.2.4 Feedback Loop

Finally, a feedback process evaluates the result of any action taken in terms of whether there was an *increase in quality* after the action was executed, and then adjusts the quality of the candidate experience, from which the action was derived, up or down accordingly. By this mechanism, the metric space is effectively altered from the point of view of the action-selection system. Closing of the perception-action loop in this way with feedback together with growth of the experiential metric space, results in the construction of modified behaviour patterns over time. This can be viewed as a form of ontogenetic development and adaptation, that is, a process of change in structure and skills through embodied, structurally coupled interaction.

7.2.5 Merging and Deletion of Experiences in the Interaction History Space

As discussed in Chapter 6, it is necessary to employ strategies such as *merging* and *forgetting*, if storage and computation requirements are to be controlled. The merging strategy in the Interaction History Architecture is to merge any two experiences closer than a threshold T_{merge} (see Algorithm 7.1). T_{merge} was fixed for the most part, however alternative strategies were trialled during development of the algorithm, including adapting the threshold such that the maximum number of experiences in the space remained constant.

Algorithm 7.1: Algorithm `IHA_MERGET`: Choose and Merge 2 experiences using a threshold

for E^i ***in*** *all experiences* **do**
  for E^j ***in*** *neighbours of* E^i **do**
    if $d(E^i, E^j) \leq T_{merge}$ **then**
      $actions(E^i) = actions(E^i) + actions(E^j)$
      $quality(E^i) = (quality(E^i) + quality(E^j))\,/2$
      $weight(E^i) = weight(E^i) + weight(E^j)$
      delete all distances to and from E^j in the metric space
      delete E^j
    end
  end
end

Algorithm 7.1 shows how the meta-information associated with experiences that are merged are also assimilated. Actions from both merged experiences are accumulated, resulting in an action probability distribution; the quality values are averaged; and, a weight value, indicating the number of experiences that have been merged together, set to the sum of the weights of the merged experiences.

Experiences may also be deleted, that is, forgotten. This serves two particular purposes in the present architecture. The first is to provide a mechanism where the interaction history space can be continually modified and so be adaptive to changes in the environmental interaction. The second, more practically, is to reduce the number of experiences in the space and so reduce computational complexity in estimating distances to new experiences inserted into the space (see also Chapter 6). There are a number of different strategies to decide which experiences should be forgotten, and the one used here is to forget those experiences which have lower quality values and thus will have little or no impact on future action selection. Specifically, experiences older than h_{future} with a quality less than or equal to T_{purge} will be deleted.

7.3 Experiment - The Road-Sign Problem (RSP-EXP)

In this section the capabilities of the Interaction History Architecture are explored using a simple simulated test-bed - The Road-Sign Problem, which is an extention of the T-Maze task.

In the classic T-Maze task, an agent (*e.g.* rat or wheeled robot) is required to navigate a simple maze with a reward at the end of one arm of the T. Also known as a *delayed response task*, this is a popular test-bed for reinforcement learning as the reward is given at sometime after the decision to turn left or right at the junction is taken. The Road-Sign problem is an extension of a simple T-Maze learning environment where an indication of the reward position is given by an earlier disconnected event. Thus the agent can make use of its experience in making the decision to turn left or right. This problem provides a benchmark test-bed for autonomous agents with some kind of short-term memory.

The aim is to find how well the system performs in this simple task, *i.e.* is it able to associate the signal and reward over a series of runs through the maze? Also investigated is the possibility of using multiple metric spaces with different horizon lengths and find if the system is able to choose actions from experiences with appropriate horizon length.

This section continues by detailing the implementation and experimental scenarios. The following setion presents the results.

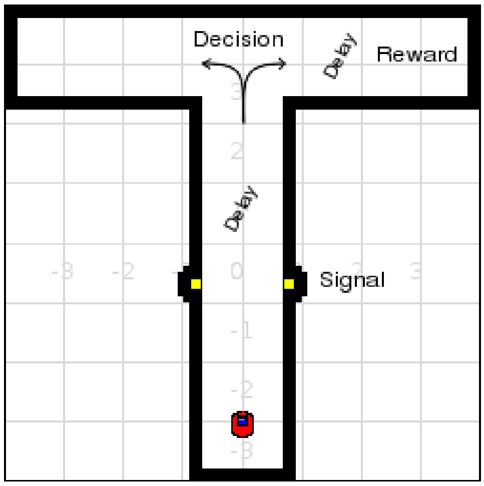


Figure 7.2: The T-Maze task

7.3.1 Implementation and Experimental Setup - (RSP-EXP)

Player/Stage was chosen to simulate a robot and the maze itself. The control and interaction history software was written in C++ using the YARP framework for interprocess communication. The simulation uses a pioneer robot model with a SICK laser scanner for localization, and a CMU camera with colour blob detection in the place of vision. See Table A.2.

The robot collects sensorimotor data continually at a rate of approximately 10 frames per second, creating experiences and placing them in a metric space. In this implementation it is possible for the robot to construct multiple spaces each of different horizon lengths on-line simultaneously.

The agent is in a "T-Maze" (Figure 7.2) at the bottom end of the T. The basic task is to travel to the junction and turn either right or left (a single action choice per iteration). A reward is placed at one arm of the T the choice of which is a variable of the experiment. While travelling to the junction, the agent encounters a signal in the form of a light (detected by the CMU blob detector) on either the left-hand side or the right-hand side. In the simplest version of the task this faithfully indicates the position of the reward in the T. Of course more complex relationships between signal and reward can be devised.

The experimental runs consist of multiple iterations of a maze with different positions of lights and reward with the robot being placed back at the start with its history intact after it has reached one or other end of the T. The initial heading is also slightly randomized to ensure that the routes taken by the robot on different iterations are not always the same. Details follow:

Reward: The motivational system is a simple reward signal and returns 1 when the robot reaches the end of the correct arm of the T and 0 at all other times. Alternative schemes can have negative rewards for reaching the wrong end of the arm, as well as returning an intermediate value while the robot is traversing the maze.

In this experiment, reward is updated from all future rewards received over a horizon h_{future} using the $\mathcal{F}_{min\_max}$ function to update the reward - see Section 7.2.3.

Actions: In order to study the effect of the interaction history in detail the robot is constrained to make a single action selection decision (turn left or right) at the junction of the T. In exploratory trials the system was less constrained, but this led to difficulties interpreting the results so the situation was simplified to have a single decision point that could be compared across trials. Additonally, the robot has a basic wall-avoidance reactive capability.

Common History: In order to compare decisions made across all runs of any particular trial, each run was preceded by an interaction history pre-populated with experience common to all runs. The common history was gathered during a single run where the robot was constrained to make the correct decisions and contained 219 experiences over 4 iterations of the maze, with the reward alternating between left and right over the 4 scenarios. Figure 7.3 shows the local view from the experience ending at timestep 18 (E_{18}), which is the timestep at which the first of the four action decisions takes place. This shows that the nearest experiences (distance 0.11731 *bits*) are those around E_{136} in the 3^{rd} iteration (the other left turn scenario). The distances to the other two decision points, E_{77} and E_{193} are 0.30491 and 0.31401 *bits* respectively, and are not as close.

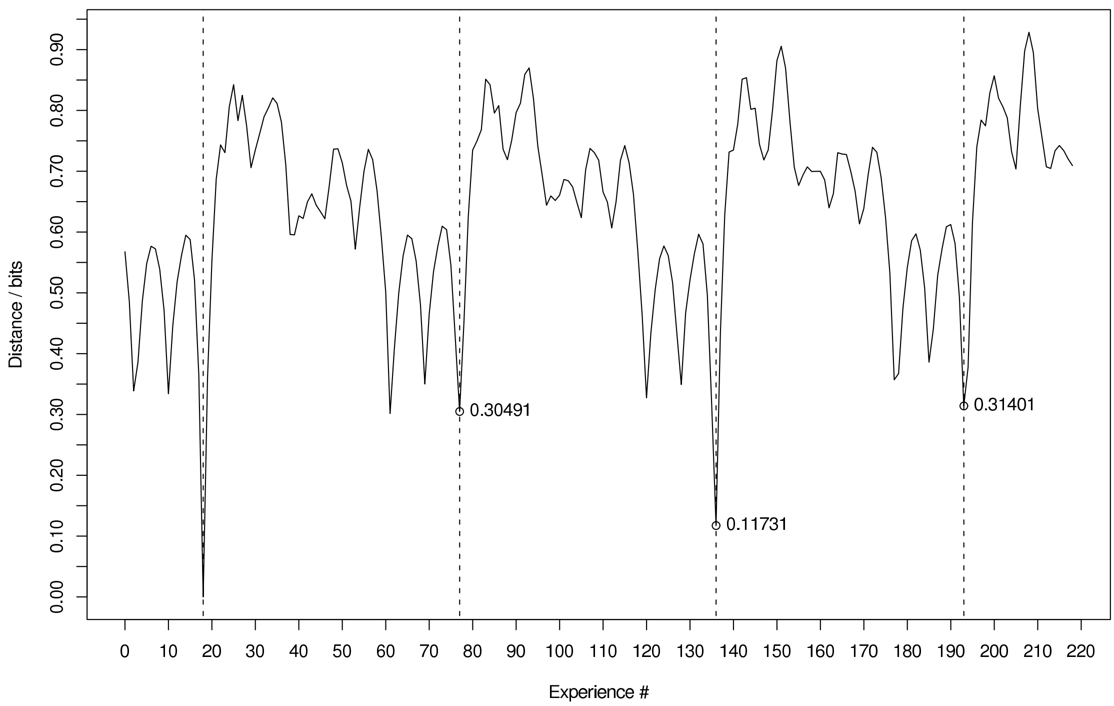


Figure 7.3: Distances from experience E_{18} (1st decision point) in common history. Action decisions are made at E_{18} (*turn left*), E_{77} (*turn right*), E_{136} (*turn left*) and E_{193} (*turn right*) marked by vertical dotted lines.

7.3.2 Experimental Trials

The following experiments were conducted:

1. *Action Selection:* Each trial consisted of 10 runs. Each run started with the interaction history space populated with the same starting experiences from a "common history" (see below). Then the robot completed 2 iterations of the T-Maze. Thus each trial presents 20 decision opportunities. No random selection chance was permitted in these trials, all decisions were made on the history information only.

2. *Action Selection with Body-babbling:* In this experiment the robot starts with an interaction history empty of experiences. It then undergoes 100 iterations of the T-maze. In this experiment, the robot uses random actions to explore the possible outcomes of its actions.

3. *Multiple Horizon Lengths:* An exploratory trial was carried out using an interaction history space that included experiences of three different horizon

Table 7.1: Summary of Results from 5 Trials

Trial	Correct (L)	Correct (R)	Total	% Correct	Parameters
1	9/10	5/10	14/20	85%	h=64 Q=5 Neighbours=20
2	3/10	4/10	7/20	35%	h=64 Q=5 Neighbours=20
3	7/10	8/10	15/20	75%	h=64 Q=5 Neighbours=4
4	10/10	9/10	19/20	95%	h=64 Q=5 Neighbours=2
5	4/10	5/10	9/20	45%	h=16 Q=5 Neighbours=2

lengths (16, 64, 128). The architecture selects the experience of shortest distance from among all three horizons at any point. This result assessed in terms of which horizons, if any, provided usable history information.

4. *Alternative Distance Measures:* The common history was constructed for certain alternative distance measures and compared to that constructed using the information distance measure.

7.3.3 Results - (RSPEXP)

Experiment 1: Test of Action-Selection Mechanism

The operation of the action-selection system given a known, favourable history was examined in this series of exploratory trials. Table 7.1 shows the results from five of the trials. In the first two, the results vary, but in total are not much better than chance (total correct over trials 1 and 2, 21/40, 52.5%). The reason seems to be that up to 20 nearest neighbour experiences are chosen for roulette wheel selection. As there are only two good examples of a similar turn (left or right) in the common history, then there is a large likelihood that an inappropriate experience, and therefore incorrect action, is chosen. To illustrate, consider Table 7.2, a list of selection probabilities from Trial 1 taken from the output of the action-selection process. A correct decision (action 2) has a probability of 44.80% of being chosen. However, if only the top two experiences are considered for selection, then a correct decision would be taken 100% of the time.

Trial 4 shows such a situation where the nearest neighbour list was reduced to

Table 7.2: Example of roulette wheel choice of experiences and associated actions ordered by weighted distance.

Exp	Hor	Weighted %	Distance	Mass	Value	Action Freq.			
						0	1	2	3
136	64	17.421436%	0.173884	1	1.0			1.0	
19	64	12.480776%	0.205438	1	1.0			1.0	
18	64	9.317441%	0.237768	1	1.0			1.0	
194	64	6.884041%	0.276618	1	1.0				1.0
137	64	5.582502%	0.307176	1	1.0			1.0	
120	64	5.166194%	0.319313	1	1.0		1.0		
178	64	5.056741%	0.322750	1	1.0		1.0		
77	64	4.804414%	0.331117	1	1.0				1.0
78	64	4.593748%	0.338624	1	1.0				1.0
3	64	4.492898%	0.342404	1	1.0		1.0		
128	64	3.812116%	0.371722	1	1.0		1.0		
62	64	3.770775%	0.373755	1	1.0		1.0		
61	64	3.516556%	0.387029	1	1.0		1.0		
186	64	3.487553%	0.388635	1	1.0		1.0		
121	64	3.224787%	0.404158	1	1.0		1.0		
2	64	3.222349%	0.404311	1	1.0		1.0		
11	64	3.165672%	0.407914	1	1.0		1.0		
251	64	00000%	0.374959	1		1.0			
244	64	00000%	0.337201	1			1.0		
236	64	00000%	0.272928	1			1.0		

Columns: *Exp*: experience number, *Hor*: horizon length, *Weighted %*: chance of selection of experience based on distance, value and weight, *Distance*: experience distance from current experience, *Mass*: number of merged experiences, *Value*: future expected reward, *Action Freq*: a frequency distribution of next actions from this experience. (Actions are 0=none, 1=Forward, 2=Left, 3=Right)

2. In this case, the history selection was correct 95% of the time. In this trial the horizon length was 64 timesteps which was long enough to include the experience of the light at the point that the decision was to be made. When the horizon is too short, as in Trial 5, then the robot again operates no better than chance.

Experiment 2: Action Selection with Body-Babbling

It has been established in the experiments so far that, given a history of experience where the robot executes the appropriate actions at the correct time, it is possible to use the interaction history architecture to correctly select actions. However, it will not always be possible to have such a perfect history on which to scaffold further learning and development. Consequently, random actions are used here, in an initial exploratory phase, to find appropriate actions and their environmental effects in given situations.

In this architecture, randomness is used in a number of different ways. Firstly it is used to select from a given set of experiences (and their associated actions) based on proximity in the metric space and other factors. Secondly, random actions can be selected instead of an action associated with one of those experiences. The chance of using random selection in this case is dependent on the relative proximity of experiences in the neighbourhood of the current experience. Finally, the relative chance of selecting a random action can be varied as development progresses. This can be viewed as a process of balancing exploration (high randomness) with exploitation (low randomness), and is a process widely used in machine learning. The process is often referred to as "simulated annealing" (Kirkpatrick, Gelatt and Vecchi, 1983) whereby an analogy with the process of working metal using controlled temperature reduction is used. In this architecture, the *temperature* adjusts the chance of randomness (see Equation 7.3). The temperature is adaptively modified from a high start, being reduced after high reward and increased after low reward.

In this experiment, the robot starts with no history, progressively building an interaction history as repeated instances of the maze are presented. A total of 100 instances are presented, alternating between two left positioned and two right positioned lights and rewards. Instead of creating regularly spaced experiences, experiences having an horizon length of $h = 100$ timesteps are created each time either the action or reward changes. Merging of "near experiences" takes place for experiences closer than $0.1 bits$; this value was chosen as a small number in relation to the radius of experiences considered to be neighbours ($1.0 bits$) which itself is small in relation to the potential extent of the metric space ($> 10 bits$). Experiences with a quality value of 0 ($T_{purge} \leq 0$) after a future horizon $h_{future} =$ 200 timesteps has passed, are deleted (forgotten) as they can no longer affect the choice of experience and thus action. The two nearest neighbours are presented for action selection at each decision point, along with a chance of random. The temperature coefficient starts at 2.0 and is adaptively adjusted as the rewards are received, to a minimum value of 0.5 and a maximum of 2.0. These values for temperature where chosen to provide sufficient randomness (high exploration) while allowing reasonable exploitation of experience.

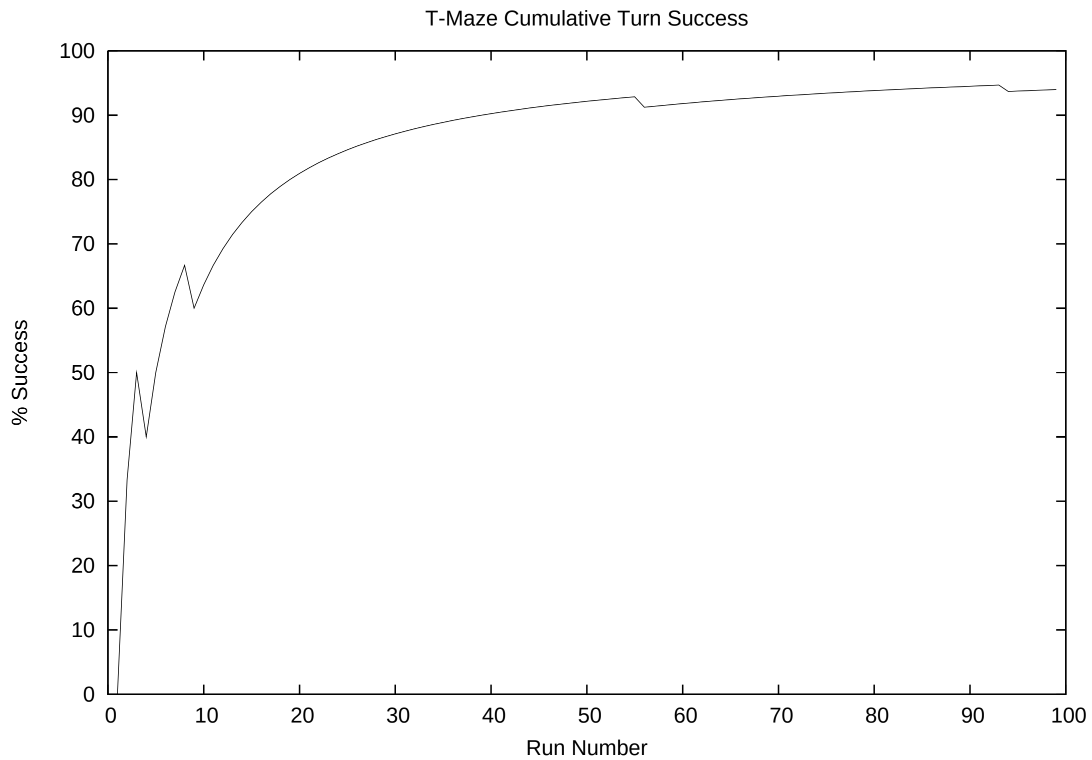


Figure 7.4: *Cumulative success rate for 100 cycles of the Road-Sign problem.* History starts empty, and random actions are used to find appropriate actions. $h = 100$, $Q = 5$, merge threshold 0.1, adaptive "temperature" reduction.

The results show that despite two unsuccessful random turns that an overall success rate of 60% is achieved after 10 cycles, 80% after 20 cycles and 90% after 40. Overall, of 100 cycles, 50 each of Left and Right; 94 were successful, with 45 of those being Left turns and 49 Right. The results are summarized in Figure 7.4. A scree plot analysis of the dimensionality of the experiences remaining in the metric space after the 100 cycles, reveals that they can be adequately represented in three dimensions, and Figure 7.5 shows a plot of the relative positions of experiences in the experience space projected into 3 dimensions. The 100 experiences on which a turning decision were made were examined further. Looking at the local picture

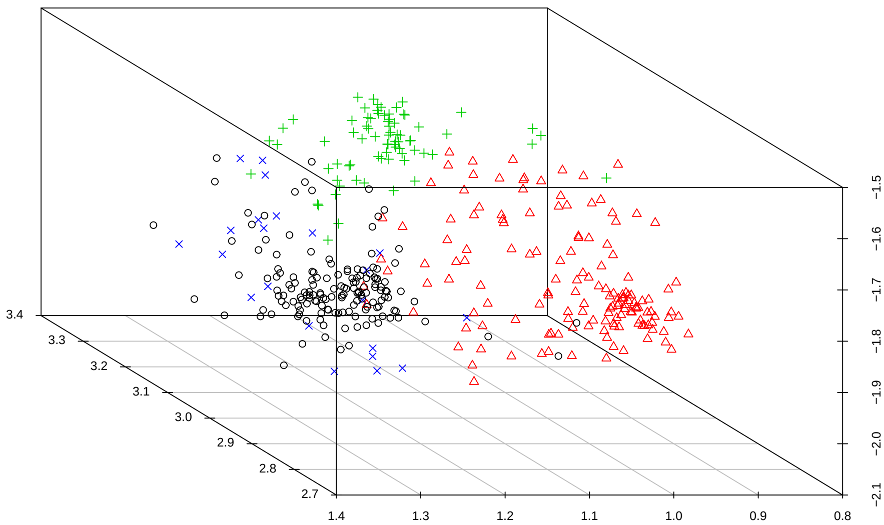


Figure 7.5: *Experiences projected into 3-dimensions.* Clusters shown (colours and symbols distinguish the clusters) were created using K-means with 4 initial cluster centres. Data clusters into 3 main groups. See text for discussion.

from two experiences from late cycles, one of each turn type, it can be seen that near experiences were also turning decision experiences of the same type. The closest 22 experiences to Exp540 (the last but one Right turn cycle) were of the same type and within a ball of radius $0.253 bits$, while the closest 17 experiences to Exp534 (the last Left turn cycle) were also of the same type and within radius $0.193 bits$. This is typical of the turning experiences.

Experiment 3: Multiple Horizon Lengths

Following the experiments with a fixed single horizon length for experiences, trials were carried out using multiple simultaneous metric spaces of different horizon length experiences. At any action selection point, the system could choose from similar experiences both within a single space as well as from other spaces. It was expected that the choice of experience would reflect the ideal horizon length for the problem at hand. However, instead it was found that the nearest neighbours were consistently of shorter horizon lengths as there is naturally less variation in shorter samples. Thus, when a set of horizons included a horizon length too short to learn the task, the system tended to choose experiences from that metric space and so failed to learn the task. We refer to this phenomena as the "horizon effect". See Figure 7.6.

In order for this strategy to succeed, it may be necessary to bias the experience choice to favour longer horizons over shorter ones. This is achieved by introducing a further term in the selection probabilities in Equation 7.1, Section 7.2.2 that balances the probabilities of selection of experiences in favour of those from longer horizons. Further testing of this problem was not carried out though, and is left as a direction for future research.

Experiment 4: Alternative Distance Measures

As discussed in Chapter 4, it is possible to use alternative distance measures in place of the information distance. Here, the information distance measure was compared with two other measures of distance, the Hamming metric and the Pearson's Squared Correlation distance in the creation of the metric space. See table 4.3.

Figures 7.8 and 7.7 show the Hamming and Pearson's distances from experience $Exp:18$ to all others in the common history as was shown for the information metric in Figure 7.3. All the measures clearly show most similarity between equivalent experiences (*i.e.* $Exp:136$ the other turn-left experience). They also show similarity to experiences at the same point in the maze but with the light on the opposite side of the wall. The Pearson's and information metric also show marked similarity between $Exp:18$ and certain others in the history, showing that they both reveal correlations in the experience beyond the obvious.

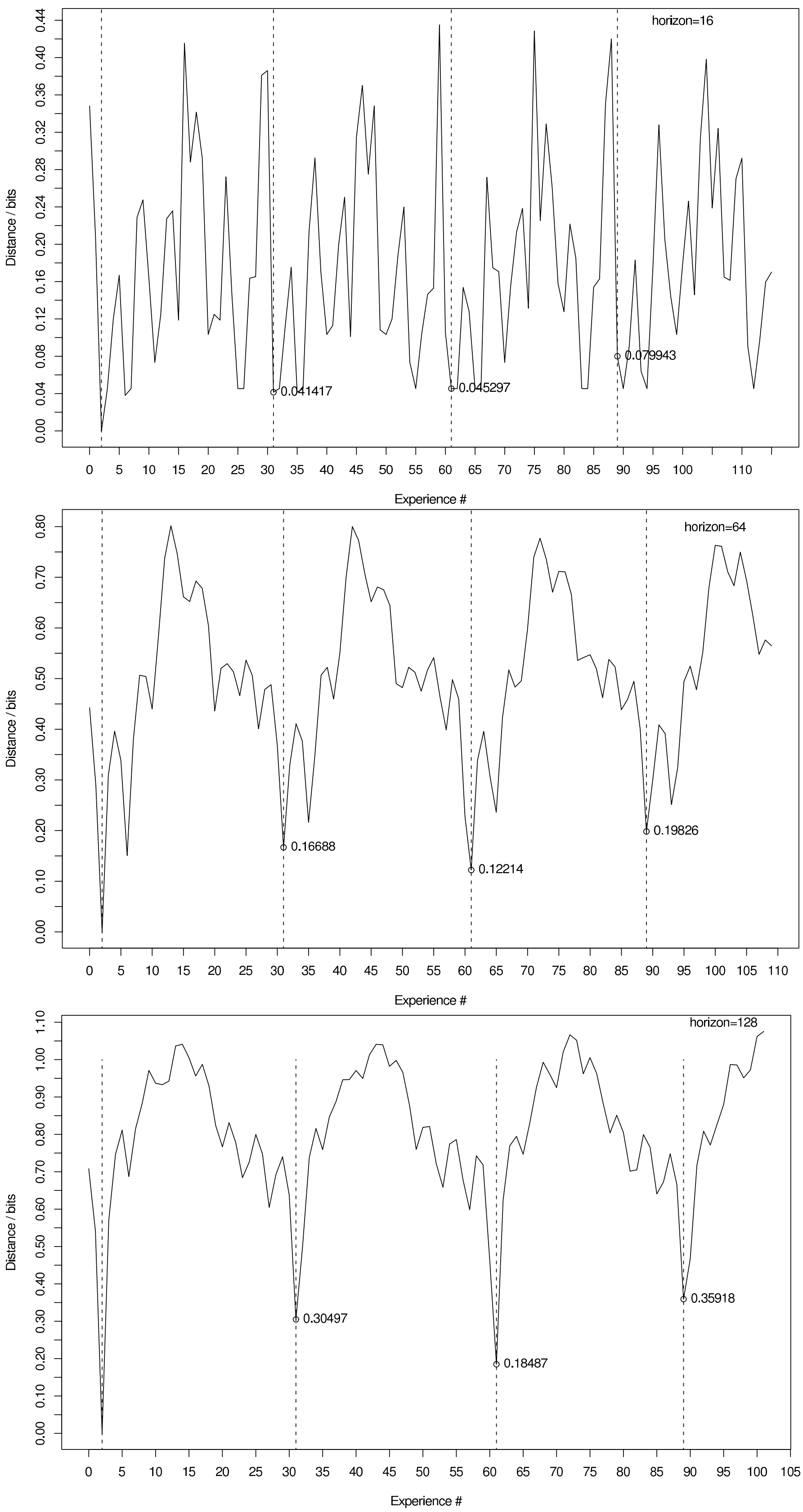


Figure 7.6: Experience distances for 3 different horizons (h=16, 64, 128). Horizon 16 (top) is not long enough to include the sensing of the light in the history at the point of action selection.

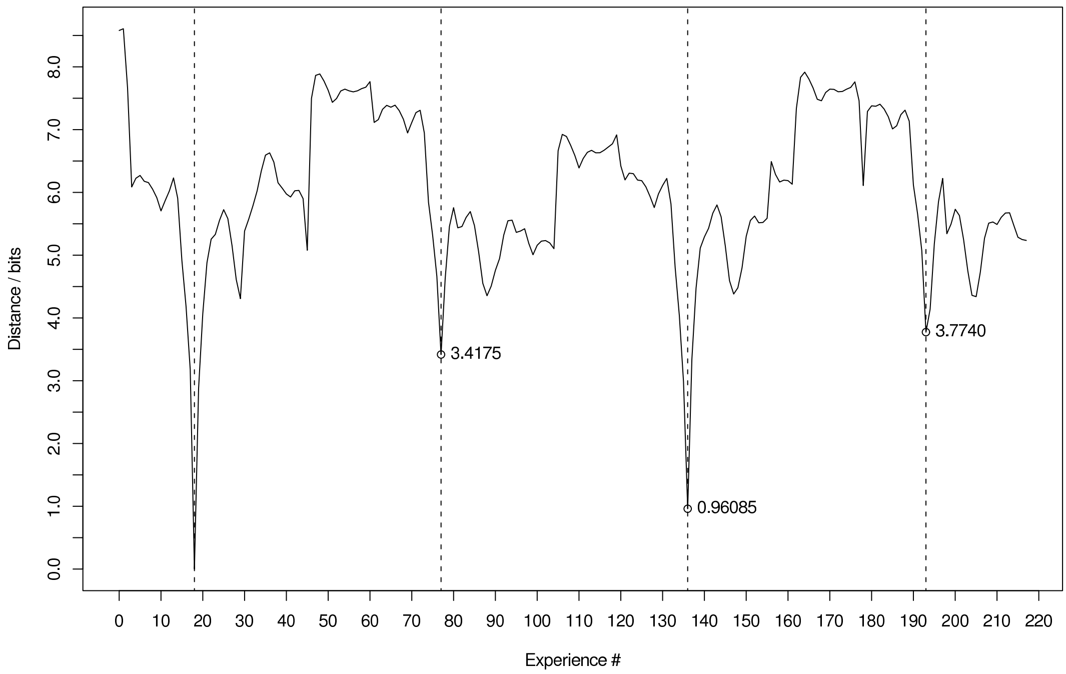


Figure 7.7: Pearson correlation distances (see Figure 7.3 for comparison with information distance.

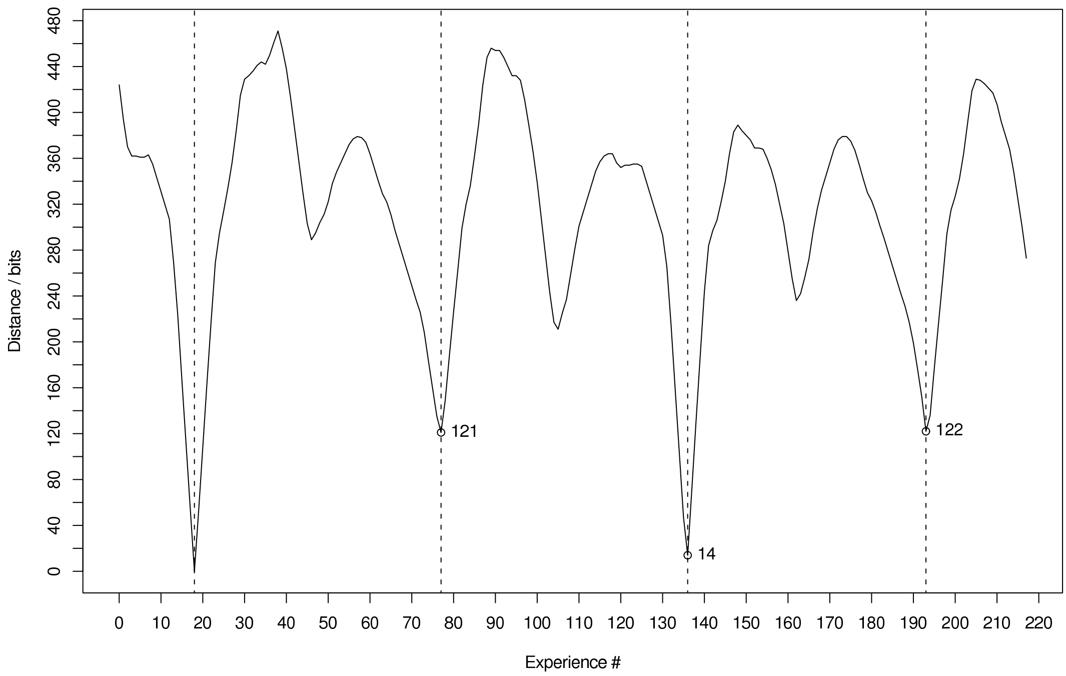


Figure 7.8: Hamming distances (see Figure 7.3 for comparison with information distance.

7.4 Related Work

7.4.1 Comparison with CCBR

Table 7.3: Continuous Case-Based Reasoning: Comparison to the Interaction History Architecture.

Feature	CCBR	IHA
Episodic Memory Representation	"cases" are associations (groupings) of sensors readings and control parameters over a period of time	"experiences" are complete sets of sensorimotor readings over a given window of time
Retrieval	Best matching "case" to current input	K nearest neighbour experiences from history, probability-based selection of experience and action
Distance Measure	Euclidean distance	Experience Metric
Action Control	Schema parameters associated with best matching case are used to modify current behaviour	Action following selected experience is executed
Modification of Memory	Modify chance of retrieval	Merging and deletion of experience, as well as modification of chance of retrieval
Exploration	none	"body-babbling" emulated by selecting random actions with some probability. "Temperature", and nearest neighbours modify this chance.
Emergent Representations	None - cases are modified.	Classes of experiences associated with action emerge as a result of merging (see Section 9.3.4)

Continuous Case-Based Reasoning (CCBR) (Ram and Santamaria, 1997) has many similarities to the Interaction History Architecture (IHA) presented in this thesis, as discussed in Section 2.6.1. See table 7.3 for a comparison of the main features of the approaches. I believe that IHA has certain advantages over the CCBR approach. In particular, the metric used in IHA allows for more robust comparison of sensorimotor details concentrating on the statistics of the particular time-series, and so better able to recognize regularities in time-series than a simple Euclidean metric. Also, the metric nature of the space is also able to recommend a number of increasingly distant matches (neighbours) and is able to weight their similarity along with a qualitative value from the environmental feedback to provide, potentially, more appropriate actions. At the same time, innovations from the CCBR approach could be incorporated into IHA, such as the focus on behaviour rather than action, and how the associations between sensor and action can be tuned.

7.4.2 Road-Sign Problem - Related Work

Much of the recent literature on solutions to the road-sign problem for autonomous agents are either neural-network based or evolutionary algorithm based. Rylatt and Czarnecki (2000) describe an Elman-style recurrent neural network solution using a type of learning called CRBP, *Complementary Reinforcement Backpropagation Learning*. Although in that original paper they do not tackle the whole problem. Bakker (2002) presents a solution to the problem that outperforms the Elman-style network, using a recurrent neural network (LSTM) as a feed into a reinforcement learning system. Thieme and Ziemke (2002) go further, testing four different neural network architectures, with the highest reliability achieved by *Extended Sequential Cascaded Networks* - a high-order recurrent neural network architecture. They showed that a short-term memory can be realized for delayed response tasks through synaptic plasticity and dynamic modulation of sensorimotor mapping.

Interestingly, Thieme and Ziemke (2002) also found that a simple feed-forward neural network could also reactively solve the road-sign problem. This is achieved by moving towards the light and then simply following the wall till the goal is reached. The memory of state is in the agent-environment interaction. In a similar vein, Bovet and Pfeifer (2005) explore the possibility that a "memory-less" agent could solve the road-sign problem. Although the agent does not have a conventional memory, it is not strictly reactive. In their case memory is achieved through a combination of some unchanging aspect of the environment (a coloured wall) and plasticity of synaptic weights between reward and the visual modality. In effect, the visual system has been altered by the interaction with the light and the subsequent presentation of the environmental stimulus induces the appropriate motor response.

Kim (2004) takes an alternative approach of evolving a controller based on Finate State Machines to analyse the role of internal memory. They looked at the size of the internal memory and states required to learn various forms of the problem involving one, two or more lights. They also studied the effect of noise on their model. They found that purely reactive controllers cannot solve the problem and multi-states were required. The simplest arrangement of a single light requiring two-states, with more states required as the number of lights increased.

Linåker and Jacobsson (2001) work at a high level, extracting significant events and clustering them to reduce the number of states down to a handful. They use a vector quantization network to extract model vectors representing event classes which in turn forms the input to a simple recurrent neural network which learns the associations between events and behaviours.

These works show that if the system is "hand-crafted" a memory with only a few states is all that is needed to solve the problem, and even that is not necessary if features of the environment can be used effectively. However, the approach taken in this thesis is not to find the best solution, but to test the interaction history architecture, as a general system for developmental learning, on one category of problems.

The seemingly trivial problem (*i.e.* The Road-Sign Problem) considered here can be thought of in a wider context. Clearly the "road-sign" itself can be anything that distinguishes one experience from another and potentially informs an agent what it should do next, and may be far more complex than a light. I believe however that an embodied interaction history constructed from the agents perspective can still be used to successfully direct future actions of agents in these extended problems.

7.5 Chapter Summary

This chapter introduces the Interaction History Architecture, an architecture whereby an embodied robotic agent can progressively use a developing history of interaction with the environment to direct action towards a high expected reward. The central structure in the architecture is the *interaction history space* which consists of a metric space of experience enhanced with environmental feedback and next action information. The architecture was then used to demonstrate that a robot was able to successfully develop the capability to complete a simple learning task that requires memory. This was achieved after the required behaviour was experienced a very few times, and contrasts with neural networking approaches (*e.g.* Linåker and Jacobsson, 2001; Rylatt and Czarnecki, 2000) which required many thousands of epochs of learning. The road-sign problem, while seeming trivial, is important as it clearly demonstrates developing action directly using the history of sensorimotor experience. The robot used all of its available sensors in the construction of experience, had no designation of which sensor carried the road-sign signal, and variation in the path of the robot provided noise and thus variation in every cycle of the maze. Thus, with appropriate extensions and modifications it may be possible to use an interaction history in other, more complex situations.

As discussed in Section 2.6.1, this approach to ontogeny and developmental learning in embodied agents is closely related to Case-Based Reasoning (CBR) in the continuous domain (Ram and Santamaria, 1997) using matching of experiences ("cases") from the history combined with environmental reinforcement to find appropriate action. Section 2.6.2 discusses extensions to the reinforcement learning paradigm that use historical information to overcome the hidden-state problem, and the "instance-based" state identification approach of (McCallum, 1996) is similar in many ways to the approach presented in this chapter. I emphasize that the use of time-extended episodes of sensorimotor experience (not state) and the experience metric are important distinguishing factors of the approach in this thesis. The experience metric allows for more robust comparison of sensorimotor details concentrating on the statistics of the particular time-series, and so is better able to recognize regularities in time-series than a simple Euclidean metric. The metric nature of the space is also able to recommend a number of increasingly distant matches (neighbours) and is able to weight their similarity along with a qualitative value from the environmental feedback to provide, potentially, more appropriate actions. This approach then does not require a Markovian environment and the agent with extended temporal horizon learns rapidly. Furthermore, it does not require a static state space to be circumscribed at the outset, but instead uses a growing and changing space of experiences, where potentially in the course of ontogeny the set and character of sensors, actuators, and embodiment may change.

Chapter 8

Peekaboo

8.1 Introduction

For robots to develop cognitive abilities appropriate for interaction with human partners and beyond those oriented around objects and navigation, we argue (see Section 2.4.4) that the complex requirements of the social environment in general, and communicative interaction in particular, are necessary in the robot's ontogeny. Motivated by this position, this chapter and the next use a simple non-verbal interaction game as a scenario where a robot can develop communicative skills foundational in the development of a "social" intelligence.

This chapter describes two experiments that use the experience metric space in a robot that develops the capability to play a simple interaction game. In the first, a human partner engages in a "peekaboo" game with a robot, and in the second the effect of the experience horizon length on the ability of a robot to develop the capability to play the game is investigated.

The chapter first motivates the choice of the peekaboo game as an interaction scenario for this study, followed by a description of the experiments and assessment of results as they relate to the research hypotheses.

8.1.1 Peekaboo as a Research Tool

The development of gestural communicative interaction skills is grounded in the early interaction games that infants play. In the study of the ontogeny of social interaction, gestural communication and turn-taking in artificial agents, it is instructive to look at the kinds of interactions that children are capable of in early development and how they learn to interact appropriately with adults and other children. A well known interaction game is "peekaboo", where classically the caregiver, having established mutual engagement through eye-contact, hides their face momentarily. On revealing their face again the care-giver cries "peek-a-boo!", "peep-bo!", or something similar. This usually results in pleasure for the infant which, in early development, may be a result of the relief[1] in the return of something considered lost (*i.e.* the emotionally satisfying mutual contact), but later in development also may be a result of the meeting of an expectation (*i.e.* the contact returning as expected along with the pleasurable and familiar sound), and the recognition of the pleasurable game ensuing (Montague and Walker-Andrews, 2001; Veatch, 1998).

Bruner and Sherwood (1975) studied peekaboo from the viewpoint of play and learning of the rules and structures of games. They also recognize that the game relies on (and is often contingent with) developing a mastery of object permanence as well as being able to predict the future location of the reappearing face. The individual parts of the game can be viewed as gestures in a non-verbal communicative interaction. The hiding of the face is one such gesture, and the vocalization, and the showing of pleasure (laughing) are others. In order for the interaction game to proceed successfully, the gestures must be made by either party at the times expected by the players, and that absence or mis-timing can result in the game cycle being broken. Learning of the game is supported by further gestures such as a rising expectant intonation of the voice during hiding, as a reassurance or cue of the returning contact. Later in development the roles

[1] In the context of humour, peekaboo in its early stages is an example of relief laughter. That is relief that the "caregiver", who is thought to have disappeared, actually has not (Veatch, 1998).

of the game can become reversed with the child initiating the hiding, while still obeying the established rules by, for instance, uttering the vocalization on renewed contact.

In all this, the rhythm and timing of the interaction are crucial and Bruner and Sherwood suggest that the peekaboo game and other early interaction games act as scaffolding on which later forms of interaction, particularly language and the required intricate timing details, can be built (Pea, 2004, pp 424-5). Discussing scaffolding, Roy Pea notes that *"...there are regularly structured situations in which the range of meanings is actually quite limited and that these simple formats provide a highly constrained situation in which the child can bootstrap some of the conventions of turn taking and meaning making with words that are required of a language user."* (Pea, 2004, pp434-425), emphasizing, therefore, the importance of early communication games such as peekaboo in the development of language.

In relation to the development of social cognition in infants, "peekaboo" and other social interaction games, that are characterized by a building and then releasing of tension in cyclic phases, are important as they are considered to contribute developmentally to infant understanding and practise of social interaction. Peekaboo provides the caregiver with the scaffolding upon which infants can co-regulate their emotional expressions with others, build social expectations and establish primary intersubjectivity (Rochat, Querido and Striano, 1999).

8.1.2 Hypotheses

The robotic experiments of this chapter attempt to address Hypothesis 4 (see Section 1.1):

Hypothesis 4: A robot can use its own ongoing interaction history to develop the capability to engage in simple, social, communicative interaction with a human partner.

The communicative interaction chosen is the peekaboo game. Our assumption, informed by the argument above, is: given that the requirement for successful peekaboo is the ability to follow a spatio-temporally structured set of "rules", then peekaboo is an appropriate example of social, communicative interaction.

As well as testing the hypothesis by observation of the robot's behaviour, we will also test the following two sub-hypotheses that will provide a quantitative verification of this hypothesis:

Sub-hypothesis 4a: that using the Interaction History Architecture to engage in the peekaboo interaction, the robot performs better than when randomly selecting action

Sub-hypothesis 4b: that the horizon length of experience needs to be of a similar scale to that of the interaction.

8.2 Motivational Dynamics

The approach taken in the interaction history architecture is to combine a metric space of experiences with environmental reinforcement (*quality*) in order that appropriate past experiences and consequently actions can be selected. The environmental feedback can be general, or task-specific.

8.2.1 Biased Sensor

In these experiments a "biased-sensor" is used (Weng et al., 1999), designed to provide feedback for the peekaboo game. This approach combines motivational reward feedback with sensing. The game has an inherent temporal structure in its cycles, and so use is made of a dynamic system of coupled equations based on a signal originating in the environmental interaction (*e.g.* perception of a face). It should be noted that since motivational systems are important, but not a direct focus of this research, a simple system for feedback was selected. However, other implementations, both more general or more specific, would work well when combined with this architecture. Indeed a modified system including an audio modality is used in Chapter 9. Future work could explore the relation between motivational systems and interaction history architectures more deeply.

A biased sensor m is required such that following a period of peekaboo-like interaction, the sensor will have a high value, providing appropriate feedback to the experience history selection process. Intuitively then, the sensor should both react to seeing a face and react strongly when a face is returns to view after having been lost for some time. Also, if a face is seen continuously without being lost then the resulting signal should not be as high as for the intermittent case nor as low as for a situation where no face is seen at all. This would encourage peekaboo while also preferring a situation where there is a face seen.

Design of Biased Sensor

The biased sensor is based on a a physical sensor that can detect a generic human face in an image. A suitable face detection algorithm is provided by the Intel OpenCV HAAR Cascades (OpenCV, 2000). The second part of the process implements a “desire” d to see a face when one is lost.

The interaction of these two variables m and d then forms a dynamical system that are coupled by equations governing how they change. m is required to reduce steadily in the absence of a face (“falling motivation”) and increase when one is seen (“excitement”). The rate of increase should be modulated by the current value of the “desire” variable d. The desire to see a face conversely decreases while one is seen (“boredom”) but increases (at a rate dependant on m) when one is not seen (“increasing desire”).

Implementation of Biased Sensor

The following equations 8.1, 8.2, 8.3 and 8.4 describe how m and d are computed at every timestep. The equations operate in two distinct situations determined by a binary meta-sensor f, determined by the face detection algorithm, representing detection or not of a face in the image (ignoring small gaps of $< 50ms$).

$$\Delta m = -\delta_3 m \qquad \text{if } f = 0 \qquad \text{“falling motivation”} \tag{8.1}$$

$$\Delta m = \alpha_2 d + \beta(C_{max} - m) \qquad \text{if } f = 1 \qquad \text{“excitement”} \tag{8.2}$$

$$\Delta d = \alpha_1 m - \delta_1 (1 - m) d \qquad \text{if } f = 0 \qquad \text{“boredom”} \tag{8.3}$$

$$\Delta d = -\delta_2 d \qquad \text{if } f = 1 \qquad \text{“increasing desire”} \tag{8.4}$$

$$d, m \text{ constrained such that } d, m \in [0, 1]$$

The labels after the equations indicate which part of the motivation system they govern.

The equations operate such that in the absence of desire d, when a face is seen m tends to a constant value set by C_{max}. When no face is seen, m decays at rate δ_3. See equation 8.1.

In the experiments described in this chapter m is used as the *quality* value for the experiences.

Choice of Parameters

Table 8.1: Parameters of dynamic equations for motivational system.

Parameter	Description	Value[a]
α_1	rate of increase of d based on m	0.12
α_2	rate of increase of m based on d	0.12
C_{max}	value that m tends to after long periods of $f = 1$	0.25
β	rate that m tends to C_{max}	0.02
δ_1	rate of decay of d when no face is seen	0.05
δ_2	rate of decay of d when a face is seen	0.05
δ_3	rate of decay of m when no face is seen	0.05

[a] See text for how parameter values were chosen.

The parameters of the dynamics equations are shown in Table 8.1 along with the values used in the experiments of this chapter. These values were chosen as reasonable settings that would be expected to give reasonable feedback given the speed of motion of the robots motors. Note that these are not particularly tuned values or should they be considered in any way specially selected. Many other combinations of values would also work well in this and other situations. However, it should be noted that with these values, the system is receptive to cyclic peekaboo episodes having a wide range of period lengths.[2] Thus, these parameters do not specify an exact length of time the face should be hidden in order to produce maximum feedback.

8.3 Interaction Experiments

8.3.1 Experiment 1: Sensorimotor contingencies in the interaction game "Peekaboo"

The purpose of this experiment was to investigate whether an embodied interaction history in a robot could be used for the robot to act appropriately in an interaction that requires following a spatio-temporally structured set of "rules", that when followed result in high value according to an internal motivational system.

Experiment 1: Experimental Setup

The robot stays in a "sitting" position (see Figure 8.1) throughout the experiment with the forelegs free to move, facing the human interaction partner at a distance

[2] Actually, this is governed largely by the length of the face-absent part of a repeating peekaboo cycle. High values of motivation can be reached with very short periods of face disappearance of just over $50ms$ up to absent periods of around 9.5 seconds - which is how long it takes for both variables to reach zero from a maximum value with the parameter setting used.

Figure 8.1: *Aibo playing "peekaboo" game.* Left: Sony Aibo with human partner Right: Using a static image. (Top: hiding head with front-leg, Bottom: Aibo's view, showing face detection.)

of 30-50cm. The actions which the robot can execute are listed in Table 8.2. Each action takes two seconds or less.

Table 8.2: Possible Aibo Actions in Peekaboo Early Interaction Game

Action	**Description**
0	Do Nothing
1,2	Look right/left
3	Track ball with head
4,5	Re-centre head
6,7	Hide head with left/right foreleg
8,9	Wave with left/right foreleg
10	Wag tail

The human partner takes a passive role with the usual interaction feedback from the partner provided by an internally generated motivational value in the robot. The action to "hide head with foreleg" means that the robot covers its forward facing camera with one or other of its forelegs, before uncovering it again a short time later.

In this experiment and the next, I define a *peekaboo sequence* to have occurred when the robot having detected a face, through action looses detection and returns to detect the face again, with this cycle repeating at least once. This is marked, due to the nature of the motivational dynamics (see Section 8.2), with a high value for the motivational variable m. The *duration* of the sequence is measured from the point of the first loss of face detection through to the last point at which high motivation can be sustained without a break in the sequence. The *average cycle period* is the average duration of a single face loss/re-detection cycle within a peekaboo sequence.

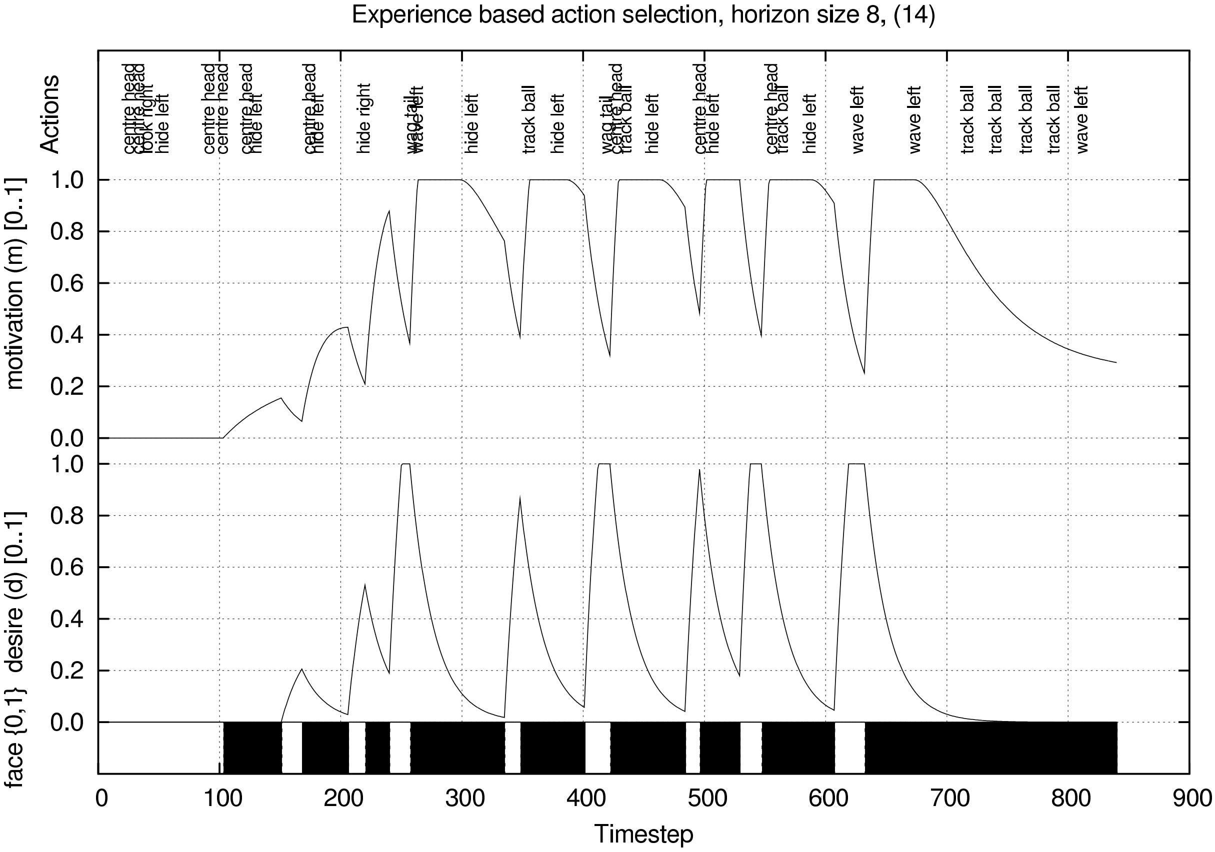


Figure 8.2: *Experiment 1: Time series of motor and sensor values showing engagement of robot in peekaboo game.* The bottom part of the graph shows when the face is seen (black bars), and the two internal variables ("desire" and "motivation") are shown varying in response to this. The actions executed are shown at the top of the trace.

Experiment 1: Results and Analysis

Fifteen exploratory trials were conducted, each lasting between 3 and 5 minutes. The results tend to show that the robot, after a period of random movement does start to engage in repeated cycles of behaviour. In 10 of the trials the robot engages in peekaboo as defined above. If the robot were not to take action to block its own camera view, it would have long periods of detecting a face which does not result in a high value for the motivational variable. Instead the robot generates intermittency in detecting a face by executing actions that turn the head away (actions 1,2,6 or 7 in Table 8.2). The trace of the internal variables as well as the actions executed from one short trial where peekaboo behaviour was observed is shown in Figure 8.2. The sequence consists of 8 repeated cycles of hiding interspersed with other actions, which importantly include actions to re-centre the head.

The trials also showed that it is easy for the robot to “get stuck” in areas of the experience space, especially if all other factors in the environment remain unchanged. This occurs 4 times in these trials, usually with the robot repeating an action such as waving.

Results also show that relatively few experiences are selected and thus modified (with regard to their stored *quality* value) over time. In some of the trials, particular experiences were selected multiple times, but this is not always the case. In the trial of Figure 8.2, 34 choices of action were made, the first 11 were random actions, and 13 of the remaining 23 actions were selected from a total of 12 previous experiences (the other 10 being randomly selected).

8.3.2 Experiment 2: Investigation of the Effect of Horizon Length

The purpose of this investigation was initially to evaluate whether the model for development based on interaction history performed better than random for the task of playing the game of peekaboo. Secondly, the hypothesis that the horizon length of experience would affect the ability to acquire peekaboo behaviour was tested by trying a number of different horizon lengths in a controlled experiment. The hypothesis was that the horizon length of experience needs to be of a similar scale to that of the interaction in question. If it is too short, the experience does not carry enough information to make useful comparisons to the history. If it is too long, then the interesting part of the interaction becomes lost in the larger experience.

Experiment 2: Experimental Setup

Again the robot stays in a “sitting” position throughout the experiments but facing instead a picture of a face (see Figure 8.1) at a fixed distance of 40cm. A picture was used rather than an interaction partner in these particular experiments to allow analysis of the robot’s interactions in isolation when comparing horizon lengths, and for experimental repeatability.

Six trials of two minute duration each, for horizon lengths of 8, 16, 32, 64 and 128 timesteps (0.96, 1.92, 3.84, 7.68 and 15.36 seconds respectively) were run. For comparison, a further six trials were run where the choice of action was random and not based on history. In each of the trials the metric space started unpopulated (empty).

Experiment 2: Results

Table 8.3 summarizes the results of 36 trial runs, while Figure 8.3 shows, for selected trials, time-series graphs of the motivational variables coupled with the actions taken. Peekaboo behaviour, as defined in Section 8.3.1 above, was seen in 18 of the 36 runs. All but one of the horizon size 8 trials, and four of horizon size 16, also showed peekaboo behaviour. The sequences were mostly generated by repetitive actions for long durations.

Table 8.3: *Peekaboo Experiment 2 Results Summary.* Duration and average cycle period in timesteps (ts) of peekaboo sequences for each trial. Where peekaboo is achieved using a waving instead of hiding action this is indicated as "waving".

Run	**Random** length/period	**Horizon 8** length/period	**Horizon 16** length/period	**Horizon 32** length/period	**Horizon 64** length/period	**Horizon 128** length/period
1	120ts / 40ts	180ts / 45ts	260ts / 40ts	none	400ts / 57ts waving	none none
2	220ts / 55ts	150ts / 40ts	none	none	none	none
3	220ts / 45ts	*Fig 6A* 640ts / 42ts	140ts / 45ts 200ts / 50ts	*Fig 6F* none	none	100ts / 40ts
4	200ts / 60ts	130ts / 45ts 150ts / 70ts	*Fig 6E* 260,240ts / 40ts	none	none	none
5	160ts / 50ts	none	140ts / 35ts waving	*Fig 6C* 540ts / 47ts waving	*Fig 6D* 220,100 / 37ts 100 / 40ts	120ts / 40ts
6	*Fig 6B* 80,140ts / 40ts	250ts / 42ts	120ts / 40ts	840ts / 47ts waving	none	none

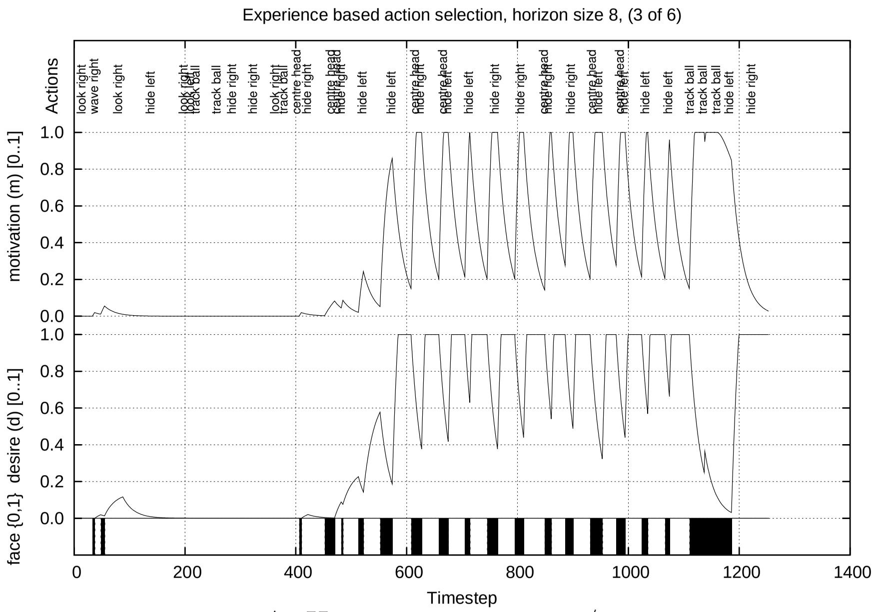
Figure 8.3A (horizon size 8) shows the best example of peekaboo behaviour; the average cycle period is approximately 42 timesteps or 5 seconds, and the sequence duration is around 640 timesteps (76 seconds). During this sequence the head is hidden to the left and right and this is interspersed with head-centring actions. Through all of these episodes, periods of no action serve to alter the timing of the cyclic periods. Although all of the trials using random action selection showed some peekaboo behaviour, they were irregular both in terms of cycle period length and in terms of the actions used to generate the sequence (see Figure 8.3B for example).

Of the longer horizon length (32, 64 and 128) trials, three showed peekaboo behaviour using repeated actions (for example Figure 8.3D) . Three also showed peekaboo using an action (waving) which would not normally cause a break in face detection. In this particular circumstance, "rocking" of the robot caused a break in face detection of more than $50ms$[3] and led to a peekaboo sequence (see Figure 8.3C for an example.)

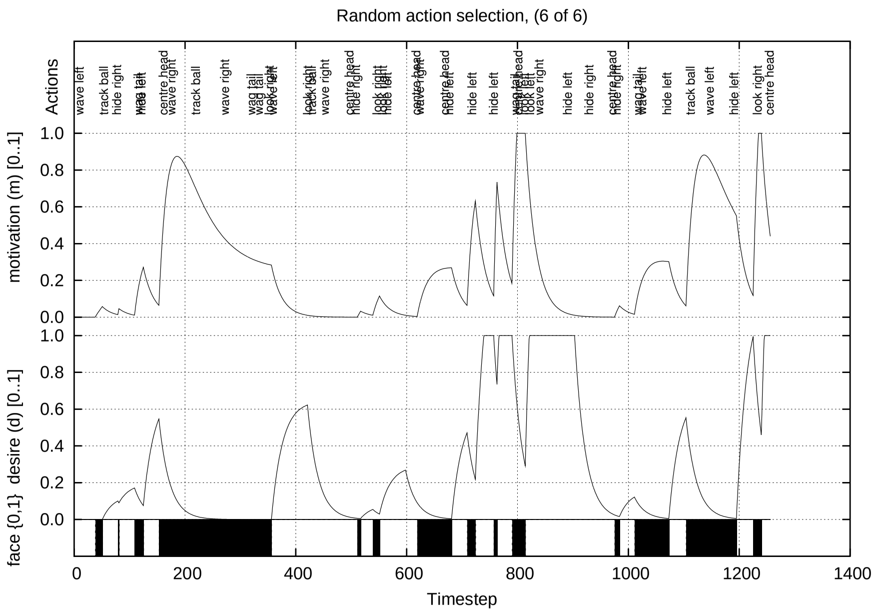
Experiment 2: Analysis

All of the trial runs of random action selection resulted in some peekaboo sequences, although with mixed, irregular actions. There are probably two main

[3] Breaks of less than $50ms$ were ignored by the motivation system. See Section 8.2

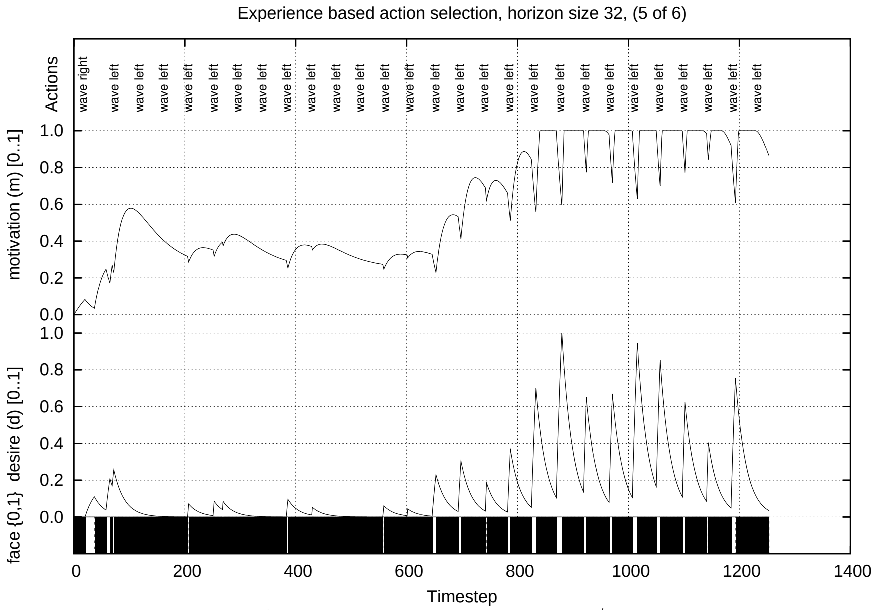


A: Horizon 8, run no. 3/6

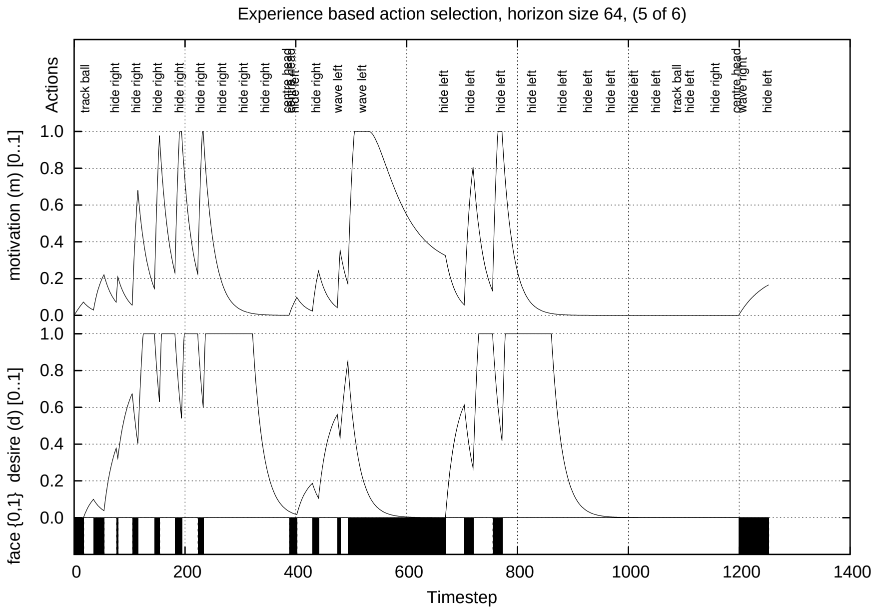


B: Random, run no. 6/6

Figure 8.3: *Motivational Dynamics and Actions for Peekaboo Experiments.* Six selected 2 minute interaction sequences of different horizon lengths. Graphs show when a face is detected (black bars at bottom), the values of the key internal variables, m and d, and the action taken at the top (Note: action 0 - "do nothing", is not shown for clarity). **A:** *Peekaboo.* Horizon size 8. Dynamics during an extended peekaboo sequence. **B:** *Random action selection resulting in high m and d.* Periods of high value are possible, but are not sustained. *continued over …*

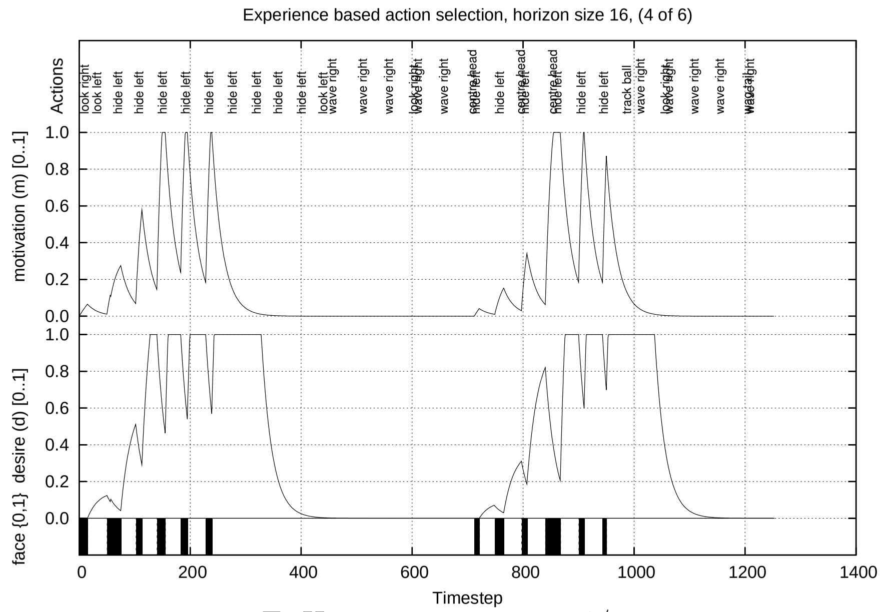


C: Horizon 32, run no. 5/6

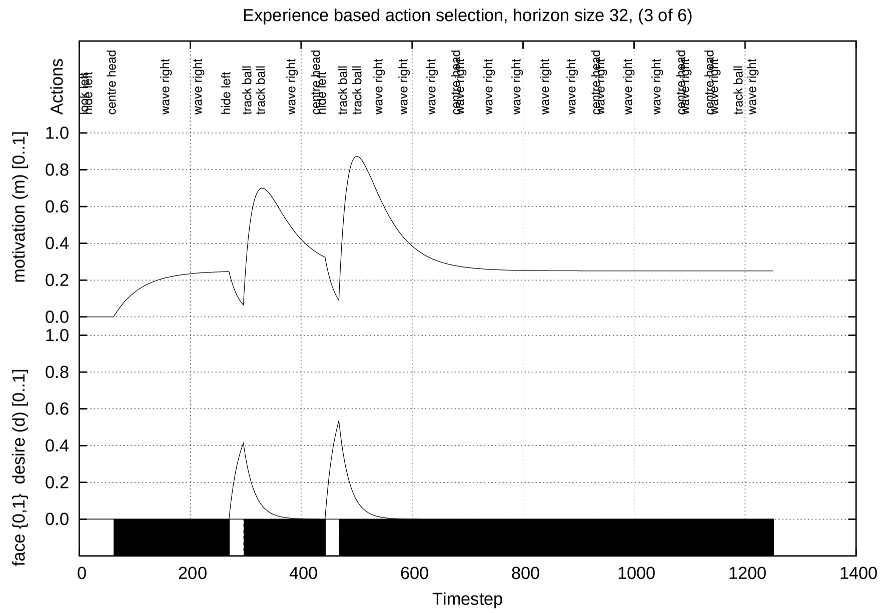


D: Horizon 64, run no. 5/6

Figure 8.3: *… continued …* **C:** *Emergent behaviour resulting in high m and d.* Horizon size 32. Dynamics generate high values when the face is intermittently lost when the waving paw returns to hit the hind knee and jogs the robot. **D:** *Irregular response to regular actions.* Horizon size 64. The regular hiding of the head does not always result in high value, this maybe because the face is not detected during the period that the head points forward. *continued over …*



E: Horizon 16, run no. 4/6



F: Horizon 32, run no. 3/6

Figure 8.3: ... *continued* ... **E:** *Repeated sequence.* Horizon size 16. Sequence of peekaboo repeated after the head is recentred. **F:** *Peekaboo not inevitable.* Horizon size 32. Here although the head is hidden twice, the peekaboo dynamics are not inevitable and coordinated action is necessary for continued high motivation.

contributing factors. Firstly, the dynamics of the motivational system responds to a wide range of periods of the peekaboo cycle. With the parameters given in

Table 8.1 the system would result in high values of the variable m after a few cycles where the face signal was lost for anywhere between $50ms$ to 9.5 seconds. Thus it was very likely that a high motivational value should be reached at some point with even random actions. Secondly, four out of ten of the actions would result in some loss of face detection, and even the wave actions caused jogging of the camera which sometimes caused loss of face detection.

However, to see longer peekaboo sequences with regular actions, some controlled behaviour must be selected and this is *only seen in the experience-driven trials*. As a contrary example see Figure 8.3F where no peekaboo-like dynamics are seen.

In some of the experience-driven trials repeated behaviour was seen that could have resulted in high motivation *if* the head had been pointed forward. Experience alone was not able to re-centre the head. On one occasion however, when the head was re-centred (randomly) then the experience space allowed a resumption of the peekaboo sequence (see Figure 8.3E). Thereafter, a recentering action is selected along with hiding actions.

The best of the cyclic behaviour was seen in the experience-driven trials of horizon size 8 and 16 timesteps (approx. 1 and 2 seconds respectively). This result indicates that it may be necessary to have an appropriately sized time-horizon, and this may be related to the length of single actions (about 2 seconds), and thus the natural period[4] of the cyclic behaviour. A reason why this may be the case is that, to bootstrap the initial repetitive behaviour, it is necessary to focus on an experience of one cycle length when there is only a single (possibly randomly generated) example of the cycle in the agent's experience.

8.4 Chapter Summary

The interaction history architecture was implemented in an Aibo robot that was able to execute a fixed set of simple actions. The simple interaction game "peeka-

[4]Note that the motivational system itself does not dictate this period as any cyclic behaviour of period up to 19 seconds can result in high values of m.

boo" was used to evaluate the Interaction History Architecture in a human-robot interaction scenario. The first exploratory experiment showed that the robot was able to develop the capability to play the game based on its own experience and an internal motivational system that was designed to reinforce a correctly executed peekaboo sequence. Further results indicate that the *horizon length* of experience plays an important role in the types of interaction that can be engaged in. The experimental results support the hypothesis that horizon length needs to be of a similar scale to that of the interaction in question, and thus should be determined, at least in part, by the types of interaction that will take place. Random action-selection regularly resulted in short sequences of peekaboo behaviour, however, only with the interaction history deriving action based on experience was the robot able to engage in sustained peekaboo behaviour, albeit only some of the time. This result supports Sub-hypothesis 4a, and combined with the support for Sub-hypothesis 4b, offers support for Hypothesis 4.

Chapter 9

Peekaboo with a Humanoid Robot

9.1 Introduction

This chapter continues the investigation of the Interaction History Architecture as the basis for developing appropriate actions in response to the ongoing history of the robot-environment interactions. The peekaboo game is used again, but this time with the addition of an audio modality, the use of an upper-body humanoid robot, and environmental reward resulting directly from the human-robot interaction. The architecture is fully implemented and includes both merging of and deletion of experiences as the mechanism for modifying the metric space of experiences.

9.2 Experimental Setup

This section details any additions or variations to the general architecture described in Chapter 7, as well as the specific setup parameters of the metric space creation and control architecture used in these experiments. Reasonable values were chosen for the various parameters, such as horizon length and merging threshold, based on the results of previous experiments and the nature of the present experiment. Furthermore, this section describes the setup for conducting the experiments and retrieving results.

9.2.1 Motivational Dynamics

In this experiment, motivation feedback (reward) is provided through two mechanisms: observation of a face, and audio feedback.

Face

As before, a face can be detected in the robot's camera image and this provides direct positive reward. Habituation causes this reward to drop-off over time. The reward for face detection, R_f, constrained to be in the range $[0, 1]$, is a function of the number of consecutive timesteps a face is seen. First the reward rises linearly, then holds at 1 for a period before decaying towards 0. R_f is calculated incrementally as follows:

$$R_f^{t+1} = R_f^t + \begin{cases} 1/\mathcal{T}_{rise} & t < \mathcal{T}_{rise} \\ 0 & \mathcal{T}_{rise} \leq t < (\mathcal{T}_{rise} + \mathcal{T}_{hold}) \\ -R_f^t/\mathcal{T}_{fall} & (\mathcal{T}_{rise} + \mathcal{T}_{hold}) \leq t \end{cases} \tag{9.1}$$

where $\mathcal{T}_{rise}$, $\mathcal{T}_{hold}$ and $\mathcal{T}_{fall}$ are paremeters that control the length of the attack, hold and decay phases. At any time a face is not detected, $R_f^{t+1} = 0$.

In this experiment the parameters were set as follows: $\mathcal{T}_{rise} = 4$, $\mathcal{T}_{hold} = 2$ and $\mathcal{T}_{fall} = 20$. These parameters were chosen as reasonable values that would give a quick response to seeing a face (reaching the maximum value in 4 timesteps, or around 1.2 seconds given a timestep length of $300ms$) but would also only slowly yield habituation (after around 6 seconds).

Sound

New to this experiment, sound is captured from a microphone, and used both as an additional sensory signal as well as providing further environmental reward.

The "energy" of the sound over the period of a timestep, ε_{sound}, provides a new sensory input to the robot. It is calculated as the sum of the amplitude of the sound signal for every sound sample in a period of a timestep, and is normalized to take values in the range [0,1]. In converting ε_{sound} to a reward signal R_s, low level background noise is attenuated by taking the square of the sound sensor variable for all values below a threshold T_{sound}, above which the reward value is set to 1. Taking the square of the sound signal results in a greater attenuation of smaller values of the variable than larger ones thus effectively reducing background noise and emphasizing the reward when the sound is above the threshold.

$$R_s = \begin{cases} \varepsilon_{sound}^2 & \varepsilon_{sound} < T_{sound} \\ 1 & \varepsilon_{sound} \geq T_{sound} \end{cases} \tag{9.2}$$

Resulting Reward Signal

The final reward signal generated by the robot in response to it's environmental interaction is a combination of the sound and face reward signals, as follows:

$$R = \max(1, \alpha(R_f + R_s)) \tag{9.3}$$

where α, in the range [0,1] attenuates the reward signal. With $\alpha = 0.5$, R is the average of the reward signals, and with $\alpha = 1$, either of the reward signals can result in a maximum resulting reward. For these experiments, α is set between these two values at $\alpha = 0.75$, meaning that neither reward signal on its own can result in a maximum R, but requires support from the other reward signal.

9.2.2 Interaction History Architecture Components and Settings

Metric Space of Experiences

The sensor rate during these experiments resulted in an average timestep length of approximately $300ms$. Experiences were created every $G = 2$ timesteps, quantizing the sensor data into $Q = 5$ bins. The horizon h for experiences was either 16 or 20 depending on the run. Quality was assigned to experiences as the maximum environmental reward received in the subsequent $h_{future} = 32$ or $h_{future} = 40$ timesteps (again, depending on the run).

Experiences older than h_{future} timesteps were deleted (forgotten) where they were associated with a quality value of less than or equal to $T_{purge} = 0.9$. Experiences were merged where both their distance in the metric space of experiences was less than $T_{merge} = 0.6bits$ and they were associated with the same next action. A combination of the merging and forgetting processes resulted in a manageable sized metric space for real-time operation.

Action Selection

The closest $K = 4$ neighbours of the current experience within a radius of $r_{max} = 2.0bits$ of $E_{current}$ were considered in the action-selection process (see Section 7.2.2).

9.2.3 Experimental Materials and Methods

Robot

The robot used was the upper-body humanoid Kaspar2 robot created at the University of Hertfordshire, see Figure 9.1. The robot has 17 individually controlled motors: three in the neck controlling head orientation, two controlling the eyes (the eyeballs are connected and move in unison - there is no vergence control), two controlling the mouth for facial expression, and five controlling each arm. The

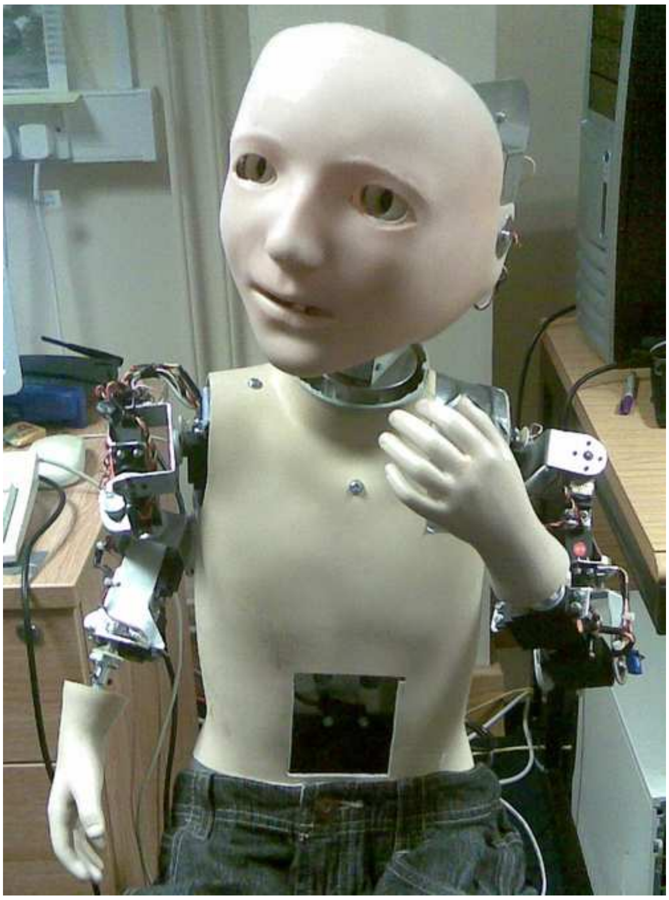


Figure 9.1: The Kaspar2 robot (University of Hertfordshire) used in the experiments.

motor control boards provide a serial link and the control software was written in C++. The interaction history architecture was written in C++ as multiple interacting modules, with the communication layer and abstraction of hardware control provided by the YARP framework (Metta, Fitzpatrick and Natale, 2006).

Actions

A total of 17 actions were available to the robot, and these can be considered in 3 groups: movement actions, facial expressions and resetting actions. These are listed in Table 9.1 and selected actions and expressions are shown in Figures 9.2 and 9.3. The types of action that the robot can execute at any time depends on which action was last executed. This is so that the robot does not attempt to execute actions that could possibly damage it. The configuration therefore defines

Table 9.1: Kaspar2 Peekaboo: Actions

Group	Number	Action	Description
Movement Actions	3	HL	Head Left
	4	HR	Head Right
	6	HID	Hide Head with Hands
	8	RAU	Right Arm Up
	9	LAU	Left Arm Up
	12	RAW	Wave Right Arm
	13	LAW	Wave Left Arm
	14	TR	"Think" Right - raise right arm to chin and look right
	15	TL	"Think" Left - raise left arm to chin
Facial Expressions	1	Smi	Smile
	2	Neu	Neutral
	16	Frn	Frown
Resetting Actions	0	Rst	All motors to resting position
	7	NA	No Action
	5	HF	Head to forward position
	10	RAD	Right Arm Down
	11	LAD	Left Arm Down

the set of next actions possible after any given action and the action selection process is responsible for ensuring that these conditions are met. For reference, these action state dependencies are illustrated in Figure E.1 and in Appendix E.

Defining a Peekaboo Sequence

A "peekaboo" sequence is defined to be a sequence of actions beginning with the robot hiding its face (action 6 - HID), followed by any number of "no-action" actions (action 7 - NA) and ending with the robot back in the resting position (action 0 - Rst). Furthermore, for the purposes of evaluating the results of this experiment the actions should be selected from previous experience rather than executed randomly.

To measure the relative amounts of peekaboo in any given period of behaviour, $p_{sel}(A^{HID})$, the percentage of times the hiding action was *selected* as compared to other "movement" actions, was used as a measure and is calculated as fol-

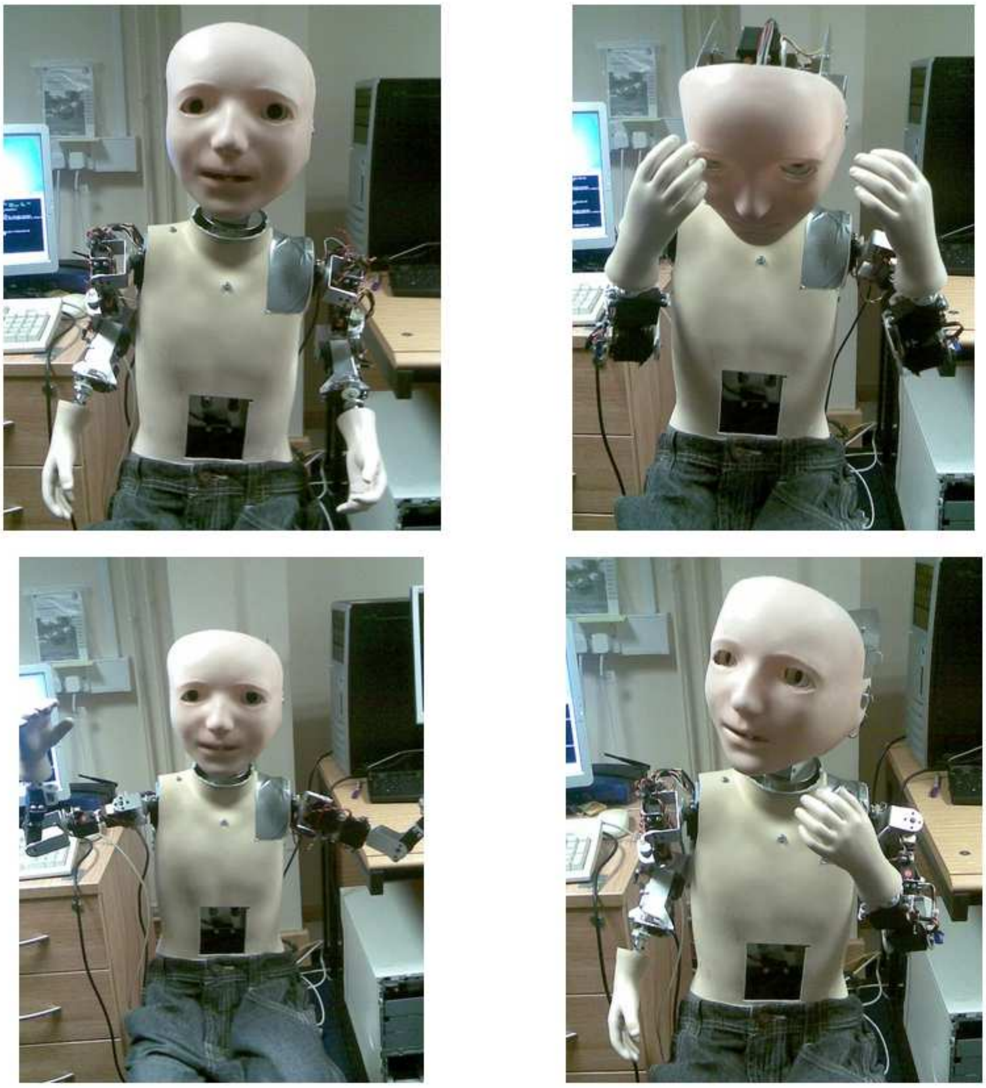


Figure 9.2: *Kaspar2 Sample Actions.* (top-left) Normal resting position, (top-right) Hiding action, (bottom-left) both arms are raised (a combination of two actions required), (bottom-right) The "think right" (TR) action.

lows. Given N possible actions $\{A^1, A^2, \ldots A^N\}$ and a period of behaviour consisting of K actions executed (selected or random), action A^n will be executed $F(A^n) = F_{rand}(A^n) + F_{sel}(A^n)$ times, where F_{rand} indicates the frequency of random executions and F_{sel} the frequency of the action being deliberately selected. Then the percentage of times the Hiding action A^{HID} was selected is given by

$$P_{sel}(A^{HID}) = 100\frac{F_{sel}(A^{HID})}{K} \tag{9.4}$$

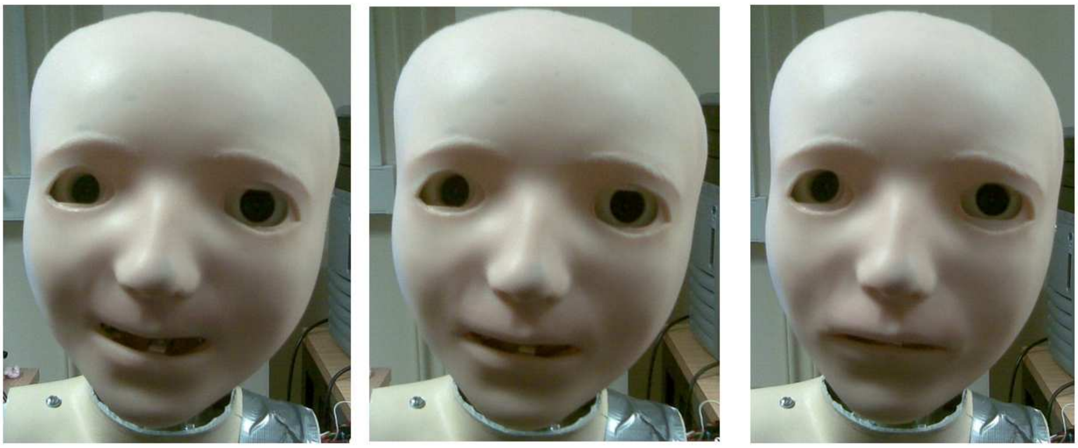


Figure 9.3: *Kaspar2 Expressions.* (left) Smile, (middle) Neutral, (right) Frown.

Note that for the purpose of evaluating "peekaboo", only actions in the "movement actions" group were considered (see Table 9.1).

Method

The robot and human partner were positioned facing each other at a distance of a few feet, with their eye-level at approximately the same height. The robot control software was started with the interaction history containing no previous experiences. Interaction then commenced with the robot executing various actions and the human offering vocal encouragement when it was thought appropriate. The interaction then continued for approximately two to three minutes.

Three different conditions were tried. Firstly, any hiding action was encouraged with a call of "peekaboo" when the robot revealed its face again. The second condition encouraged an alternative action which also turned the robot's head away from the interaction partner. Both "head left" and "think right" were used for this purpose. The final condition was to offer no vocal encouragement at all during the interaction.

The experimental hypothesis was that encouraging the hiding action would result in a higher rate of peekaboo sequences than would be expected from random action selection. Furthermore, this should also be the case when other actions are encouraged instead. Finally, this hypothesis was also tested by the no-encouragement condition with the expectation that no action would be selected in preference to any other. This experimental hypothesis is in support of Hypothesis 4, Section 1.1.

Note that for all these experiments I personally took the role of the human partner and so was fully aware of the capabilities of the robot and of the software. Further experiments should also utilize interaction partners that did not have such prior knowledge.

Success Criteria

To consider a run successful the encouraged behaviour should be executed repeatedly for some extended period of the run. Remembering that the system starts by executing random actions and building-up experience before potentially using its history to execute the appropriate action repeatedly, then we might reasonably consider the run to be successful if the behaviour made up at least a third to half of overall behaviours executed. Furthermore, a full peekaboo cycle would be comprised of more than one (usually 2 or 3) selected actions that together make up the selected behaviour. So from an action perspective if the encouraged action was selected more than around $10 - 15\%$ of the time, then the run could be considered successful. However, the percentage of selection alone was not the sole criteria for judging success. Instead, each trace was examined to see when, if, and how often repeated behaviour was executed (all traces are reproduce in Appendix C for reference). Ultimately however, some runs were still considered borderline - that is they may have failed to satisfy some aspect of the criteria. The comments in Table 9.2 offer explanations for the decisions in these and other cases.

9.3 Results

A total of 22 runs were completed. 16 of these for the first condition (encouraging the Hiding action), 3 for the second condition and 3 for the no-encouragement

Table 9.2: IHA on KasparII: Experimental Runs Summary

Run	Encouragement Type	Horizon	Comment	HID Chosen %	Result[a]
d0032	Peekaboo	16	HID action executed early and repeated many times	55.17%	Success
d0033	Peekaboo	16	HID action executed early and repeated many times	41.18%	Success
d0034	None	16	HID action only twice randomly	0.00%	Success
d0035	Encourage HL	16	HL action chosen often. HID also chosen. HL=36.59%	14.63%	Success
d0036	Peekaboo	16	HID chosen often.	42.11%	Success
d0037	Peekaboo	16	3 HID actions selected, but RAW selected more often	13.64%	Fail
d0038	Peekaboo	16	No random HID to encourage.	0.0%	Fail
d0039	Peekaboo	16	Hid was only action chosen (once) but run too short	12.50%	Borderline
d0041	Peekaboo	16	Mixed actions - some peekaboo	5.49%	Fail
d0042	Peekaboo	16	Mixed actions	9.68%	Fail
d0043	Peekaboo	16	HID only twice	1.09%	Fail
d0044	Peekaboo	16	Peekaboo throughout	18.87%	Success
d0045	None	16	Few random HID actions	0.00%	Success
d0046	Encourage HL	16	HL action chosen many times, HID a few times. HL=11.84%	2.63%	Success
d0049	Peekaboo	20	Only a few random HID actions	3.26%	Fail
d0050	Peekaboo	20	HID chosen often	26.32%	Success
d0051	Peekaboo	20	HID chosen often	19.32%	Success
d0052	Peekaboo	20	HID not chosen enough for success over run. However, regular peekaboo was begining to occur at the end.	4.96%	Borderline
d0053	Peekaboo	20	HID chosen often	17.46%	Success
d0054	Peekaboo	20	HID chosen very much. HID was 1st action	61.76%	Success
d0055	Encourage TR	20	TR (Think-Right) encouraged. TR=26.00%	0.00%	Success
d0056	None	20	Some HID actions chosen	2.53%	Success

[a] See text Section 9.2.3 for explanation of Success/Fail criteria.

condition. The results are summarized in Table 9.2 and more details of the results from the individual experiments are given in Appendix C. In most of the experimental runs it was fairly straightforward to estimate whether the experiment successfully supported, or clearly failed, the hypothesis that the interaction history would result in increases in frequency of the encouraged action. However, in 2 of the runs, this was not possible ("borderline" in Table 9.2). In run d0039, the hiding action was the only one to be selected (rather than chosen randomly) however the run was too short[1] for successful evaluation. In run d0052, the figures for the whole run do not indicate success, however, the results are borderline as the peekaboo behaviour was clearly beginning to occur towards the end of the run.

Where a result could be determined, 14 out of 20 runs (70%) were successful. In the following sections representative results from each condition are discussed.

9.3.1 Peekaboo Encouragement Condition

Figure 9.4 shows for the first run (d0032), how the motivational variables (face, sound and resultant reward) vary with time, along with the actions being executed. The interaction partner encourages the first "peekaboo" sequence ("hide-face" on the diagram). Note that a "peekaboo" action is actually a combination of the action to hide the face (action 6), any number of "no-action" actions (action 7) and an action to return to the forward resting position (action 0) (for clarity only the primary action is shown on the trace). This results in a maximal reward shortly after the hide-face action, and as the interaction partner continues to reinforce the peekaboo behaviour with vocal reward, this pattern can be seen repeated throughout the trace.

As the chance of choosing a random action rather than selecting one using the history gradually declines the early part of the run will be more exploratory (have more randomly selected actions) whereas towards the end of the run, actions will be more likely to be deliberately selected using past experience. It can be seen

[1]In this case the program terminated with a fault before execution was complete.

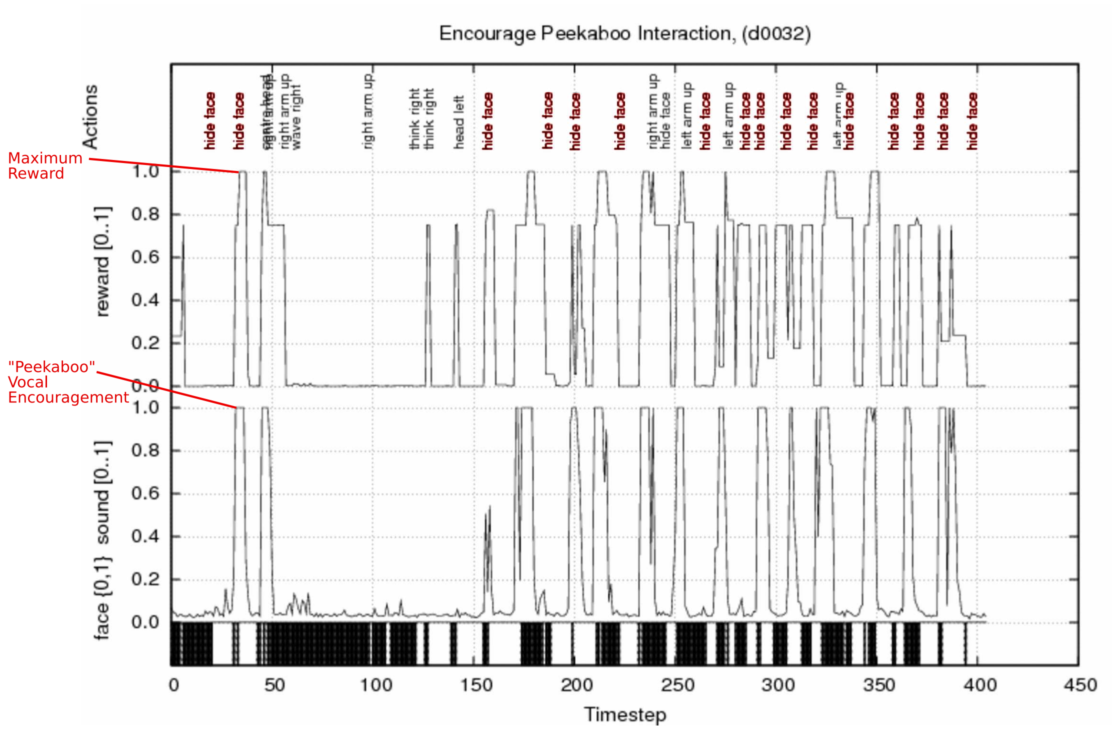


Figure 9.4: *Kaspar2 Results d0032. Example of Peekaboo Encouragement Condition.* The trace shows, against time, the detection of the face and audio encouragement as well as the resulting reward. Along the top are shown the actions executed.

that during the first half of the run various different actions are tried, but during the second half of the run, the "hide-face" action is chosen regularly.

The timing of the motivational feedback given by the interaction partner to the robot is important in determining what actions are executed. In Figure 9.5 from run d0050, the encouragement for the hiding action (and subsequent actions to return the robot to the resting position) is only received *after* the robot additionally turns its head to the side. The result is that when the robot decides to repeat the hiding action, it generates experiences which are likely to generate the actions that were executed following the original hiding action, *i.e.* the robot hides its face, returns to face the front and immediately turns its head to the side.

This behaviour (of the architecture) is an important part of how not just single actions are repeated, but instead how sequences of actions and robot behaviour are replayed, and it is this that encourages a fuller development of capabilities of the robot. It is important to note also that a specific sequence of actions are not

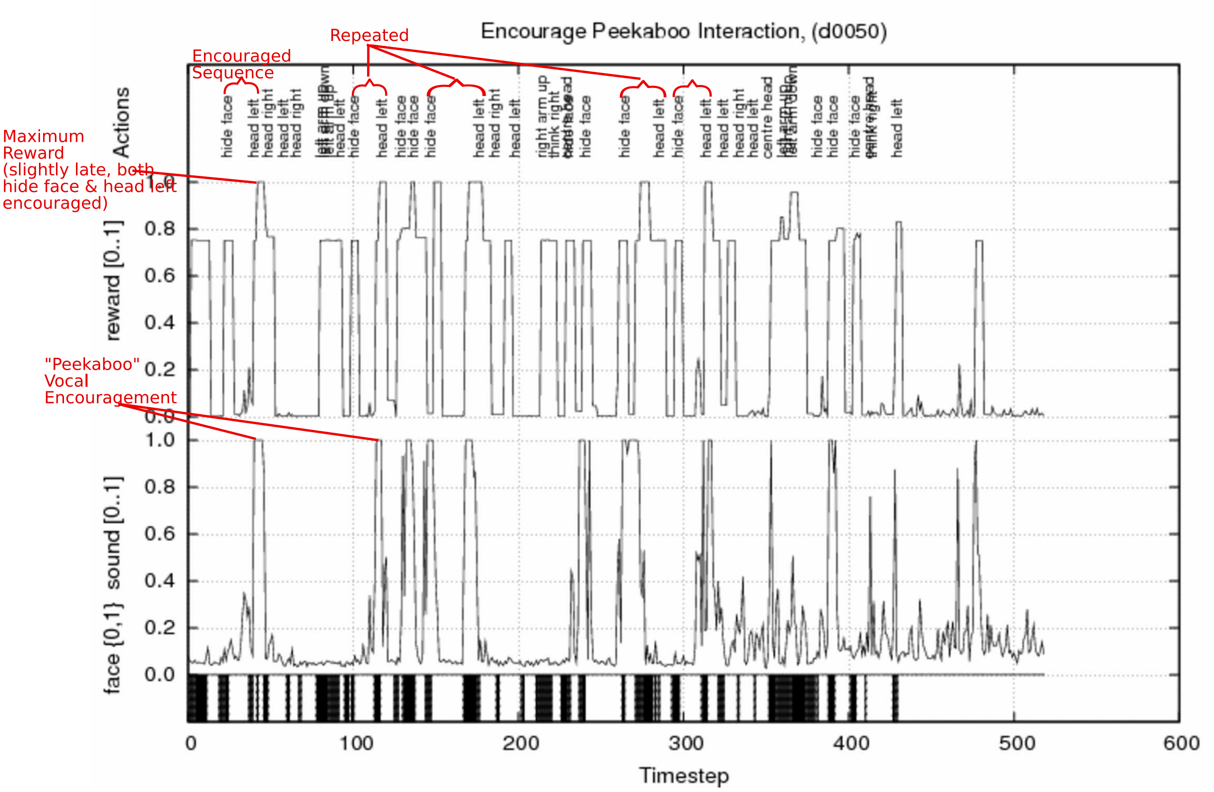


Figure 9.5: *Kaspar2 Results d0050. Showing a repeated action sequence.* A multiple action sequence is encouraged and repeated here.

learnt, instead it is the continuing generation of experience through the structural coupling of the embodied agent and its environment that drives this observed repeated behaviour. This can be clearly seen from Figure 9.5 in that the timing of the subsequent head-turn following a hiding action is not always the same, and indeed does not always occur.

9.3.2 Alternative Action Encouragement Condition

To illustrate that the operation of the interaction history is not limited to the peekaboo behaviour, the interaction partner also encouraged certain alternative actions rather than hiding. In two cases the "head left" (HL) action was encouraged (once also with a different call of "hello!" instead of "peekaboo!") and in one case the "think right" (TR) action was encouraged instead. In each of these cases the predominant action after some time was the encouraged one. Figure 9.6 from run d0035 shows a situation where the head-left action was encouraged, and

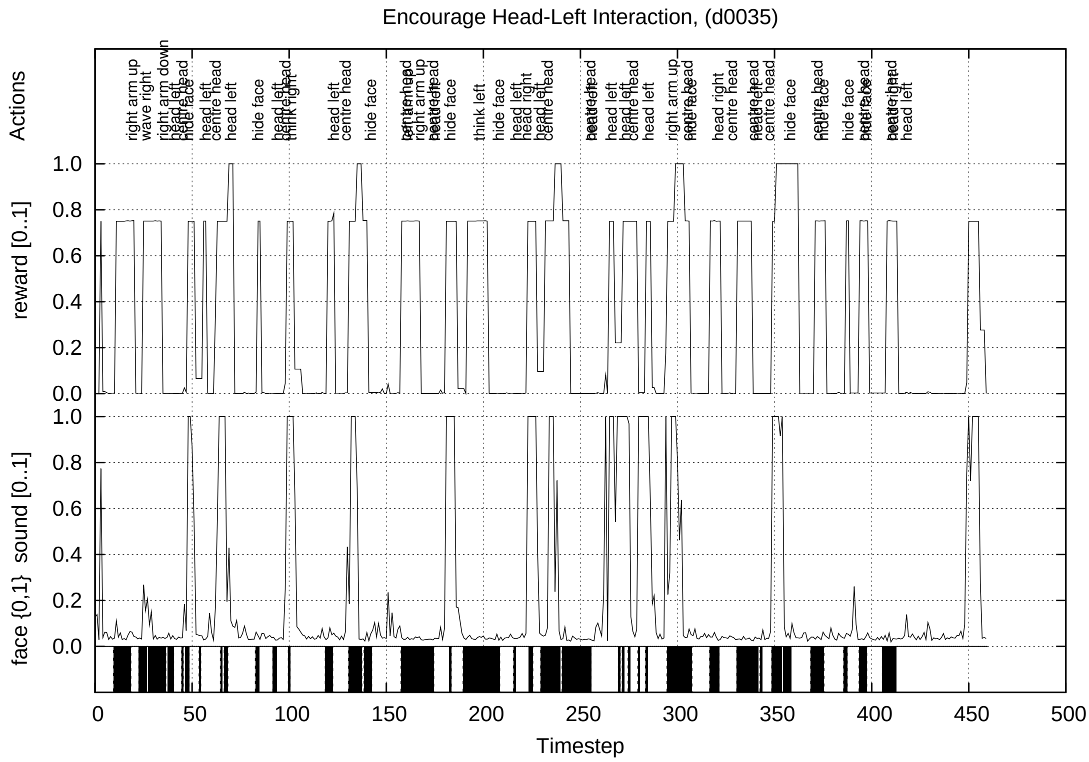


Figure 9.6: *Kaspar2 Results d0035. Encouraging and alternative action.* The “head left” (HL) action is encouraged and repeated.

it can be seen that the HL action was chosen in 36.9% of the “movement” actions whereas the Hiding action, for reference was chosen in 14.63%.

9.3.3 No Encouragement Condition

The final condition where the interaction partner offered no or very little encouragement resulted in various kinds of behaviour, none of which reinforced any particular action over any other, other than “doing nothing”. An example is shown in Figure 9.7, where no encouragement at all is offered. In this case, some random actions are chosen but as time goes on, movement actions are not chosen and the robot executed actions that keep it stationary (the resetting actions in Table 9.1). In this case 152 actions are executed with only 32 actions of the “movement” type, evenly spread among these actions. The remaining 120 being mainly “Rst” and “NA”.

In the other cases where no encouragement was offered (runs d0034 and d0056

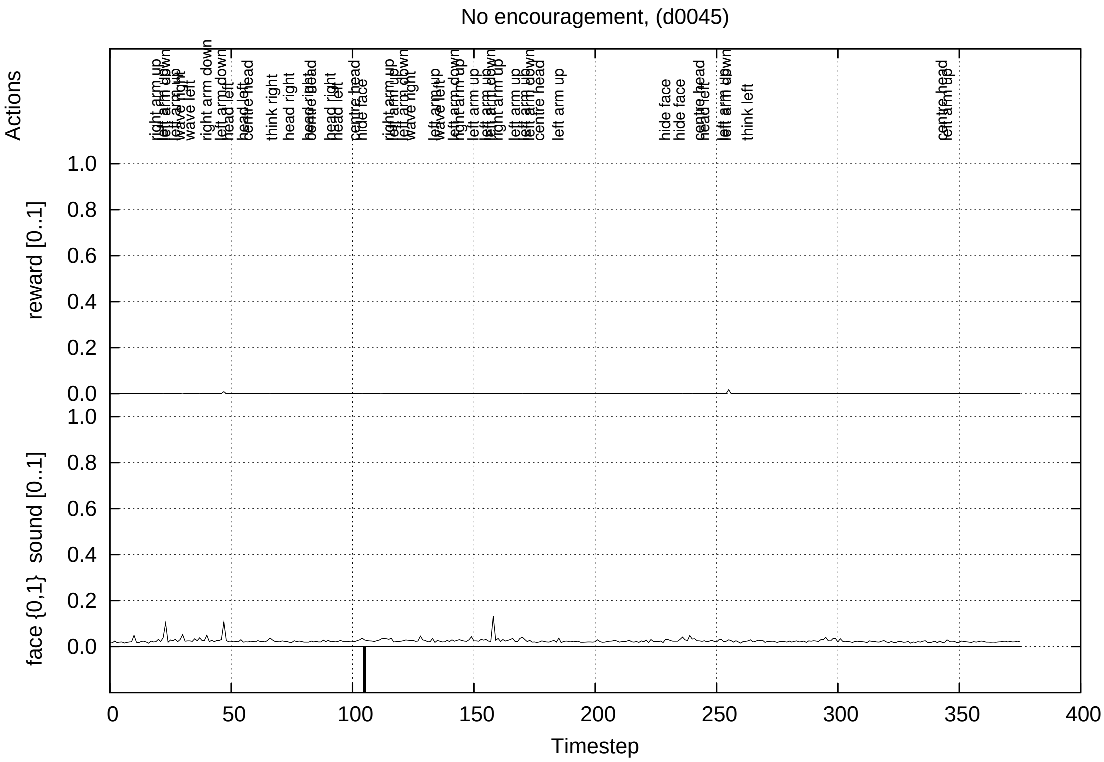


Figure 9.7: *Kaspar2 Results d0045. No Encouragement condition.* No encouragement is offered and the robot develops no action pattern.

- see Appendix C) the robot did receive some reward albeit not a maximal reward. In these cases the robot did have actions from recent behaviour to choose from, however, the behaviour did not become repeated over the long term as continual merging and purging of experiences that do not result in near maximal reward resulted in only transitory behaviour. Thus the modification of the space through merging and deletion plays an important role.

9.3.4 Emergent Classes of Experience

Analysis of the results shows that there was an extensive reduction in the number of experiences in the metric space through forgetting and merging, often reducing the number of experiences by over 50%, and sometimes by much more. The merged experiences were however fairly small in number (as experiences were often deleted rather than merged).

Examining a single example, run d0033, a successful peekaboo run merged

Table 9.3: Merged and Forgotten Experiences

Run	Total	Deleted	Merged	% Deleted	% Merged
d0032	193	73	11	37.82	5.70
d0033	181	63	14	34.81	7.73
d0034	145	119	17	82.07	11.72
d0035	203	97	5	47.78	2.46
d0036	114	35	5	30.70	4.39
d0037	191	114	20	59.69	10.47
d0038	199	183	0	91.96	0.00
d0039	57	6	8	10.53	14.04
d0041	446	315	23	70.63	5.16
d0042	330	170	18	51.52	5.45
d0043	409	356	1	87.04	0.24
d0044	283	101	7	35.69	2.47
d0045	179	163	0	91.06	0.00
d0046	371	243	17	65.50	4.58
d0049	531	346	65	65.16	12.24
d0050	205	58	0	28.29	0.00
d0051	422	133	76	31.52	18.01
d0052	554	389	68	70.22	12.27
d0053	367	92	55	25.07	14.99
d0054	448	268	12	59.82	2.68
d0055	329	145	15	44.07	4.56
d0056	305	264	21	86.56	6.89

experiences 14 times. One experience that was merged with many later ones was experience number 1 (the second experience). That experience was merged with 8 other experiences and was associated with action 6 (HID - the “hiding” action). Often when the HID action was chosen, it was experience number 1 which was found to be similar to the current experience. Thus it is possible to say that a class of experiences was emerging during this run that “represented” to the robot that it should next execute the peekaboo “hiding” action.

9.4 Chapter Summary

The Interaction History Architecture was implemented for the upper-body humanoid robot Kaspar2. The peekaboo interaction game was used to evaluate the architecture in terms of how the robot could use its own personal interaction history to develop the capability to engage in the game. Results show that giving appropriate encouragement to the robot as it executed certain series and groups of behaviours can result in those behaviours being selected in preference to others in equivalent conditions and this result supports the hypothesis that encouraging the hiding action would result in a higher rate of peekaboo sequences than would be expected from random selection. Furthermore, encouraging alternative action sequences resulted in those actions being repeated and inviting the conclusion that this behaviour is fairly general and is not limited to peekaboo. Additional support for the hypothesis was found in the conditions that offered no encouragement and in these cases no single action or sequence was selected in preference to any other.

It was found that classes of experiences emerged through the process of merging of experiences as the interaction progressed. These classes of experience and their associated next-action can be said to be emergent, grounded "representations" that have "meaning" from the robot's own perspective in the actions they generate.

Chapter 10

Conclusions

10.1 Summary

This thesis has presented a framework for the ontogeny of behaviour in artificial autonomous agents (and particularly in robots) that is centred on the agent's own grounded sensorimotor history of interactions. The thesis started by defining what an interaction history is for an autonomous agent and motivated that definition with reference to literature, taken from psychology and cognitive science as well as artificial intelligence, that supported an embodied grounded perspective on memory, development and cognition.

Next, exploratory research into how such a definition for an interaction history could be realized was presented. A Sony AIBO robotic dog was used to investigate how a robot might characterize and recognize its own behaviour from its sensorimotor history alone, with the view that such a capability would be an essential feature of a useful interaction history. In these experiments changing information theoretic relationships between sensors provided a useful characterization of behaviour.

Having defined what it means to have a grounded interaction history, and conducted some preliminary research, it was possible to arrive at a natural operational definition of a sensorimotor "experience" and how such experiences can be aggregated to form a space of experiences. Various information theoretic measures of the distance between experiences were developed and presented to support this aggregation. One such measure, the *experience distance*, was taken forward in the subsequent research and its properties investigated. The experience distance is a metric measure and the space of experiences a metric space. The dimensionality of an autonomously constructed metric space of experiences is potentially very large, however it was established in experiments that the actual dimension was in fact considerably smaller and reflected the ordered nature of the space when populated with natural experiences of a robot interacting within its environment. In robotic experiments where a ball was moved in front of a robot, the metric space was successfully used to anticipate the future motion of the ball by finding historical experiences with short experience distance to the current experience (*i.e.* were "similar"). Additionally, methods of constructing (and maintaining) an experience space were investigated with the goal that they should be computationally manageable.

The research mentioned thus far addressed the first two research goals providing formalism to concepts such as experience and history as well as establishing quantitative methods for comparison of robot self-experience (See "Research Questions and Challenges" Section1.1). The remaining research (Chapters 7,8 and 9) addressed the third goal, *i.e.* "To find, implement and test mechanisms whereby an agent may autonomously and open-endedly shape its control structures for action and behaviour, based on its ongoing history of past experiences." Thus, an architecture for controlling an autonomous embodied agent based around the metric space of experience, the *Interaction History Architecture* (IHA), was designed and implemented with controllers for various robots, simulated and physical. The architecture was tested on a simple test scenario - the "Road-Sign Problem" and it was established that a robot could use its sensorimotor history as embodied in the IHA to make the correct turning decisions.

The IHA was then given a considerably harder task, that is, to develop in a robot the capability to play the simple communicative interaction game "peekaboo" with a human partner. The architecture was implemented for both an AIBO robotic dog as well as an upper-body humanoid robot with expressive capabilities. Two different schemes were used for generating environmental reward to support the developmental learning, the first being based on the detection of a face only and the second receiving reward for both seeing a face and hearing a sound. The results of the experiments showed how it was possible for the robot to play peekaboo in various ways with the human partner, in some cases using unexpected combinations of actions to achieve high reward from the interaction. Furthermore, in the second experiment with the humanoid robot, it was established that sequences of interactions and behaviours other than the peekaboo hiding interaction could be specifically encouraged and then repeated. It was also established that the history length of experiences in the metric space needed to be of an appropriate length in relation to the interaction such that the best performing horizon lengths were approximately the same length as the time it took the robot to hide and reveal its “face”.

Clearly, there is much research to conduct yet before the capability of developing wide ranges of behaviours in wide ranges of scenarios through ontogeny over long periods of time is possible for a robot using its interaction history. However this research takes some important and significant steps towards such a goal. Needless to say, the ontogeny of prospective ability of children and other mammals is an extended process lasting years and we cannot yet hope to mirror its complexity and success in artificial systems, although the work presented here suggests that we have made a small start in this direction.

This summary concludes by reviewing the research questions and contributions to knowledge. Then, in the following sections, various issues arising from and implications due to the research presented are discussed. Finally, we look forward at possible directions for further research.

10.1.1 Review of Research Questions

Hypothesis 1: *The changing gross informational relationships between groups of sensors of an embodied agent, situated and acting in an environment, can be used to characterize the behaviour of that agent (agent-environment interaction).*

The experimental results of Chapter 4 established that this was the case; a robot executed different behaviours (walking, turning *etc.*) and it was possible for the experimenter to characterize the various behaviours based on the informational relationships between sensors. Furthermore, later experiments (Section 5.4) showed that motion of a ball, both with and without concomitant motor information from sensorimotor coordinated action, could be distinguished and thus characterized as similar to recent movement.

Hypothesis 2: *It is possible for an agent to recognize its own behaviour in terms of these informational relationships between groups of sensors.*

This was studied and established in Sections 4.5 and 4.6 as regards the Aibo wandering in the arena, as well as in the experiments mentioned regarding ball-motion in Section 5.4.

Hypothesis 3: *By using a temporally extended history as the basis for action, links between experiences and actions may be built that allow the agent to act such that it exhibits the appearance of prospection of repeated and familiar events in its environment.*

The experiment regarding prediction of ball-motion (Section 5.4) lays the foundation for this capability establishing that self-recognition of experience can be used for prediction. Later experiments with the interaction history, Section 7.3 in particular, show that such a history can be combined with actions that appear to predict where future reward might be highest.

Hypothesis 4: *A robot can use its own ongoing interaction history to develop the capability to engage in simple, social, communicative interaction with a human partner.*

This kind of capability was demonstrated in Chapters 8 and 9. The task required the robot to aquire the capability to engage in a communicative interaction ("peekaboo") with a human partner. The robot could execute a limited number of actions and had a reward system that encouraged peekaboo-like interaction, but otherwise had no direct knowlege of the game. It was found that, by exploring the possibilities by executing actions at random, a history of interactions would be developed and exploited in an on-line manner that enabled the robot to successfully engage in a peekaboo interaction.

Hypothesis 5: *A dynamically constructed history of interactions that is used to generate and select actions in an embodied agent can serve to scaffold the ontogenetic development of the agent.*

The experiments of this thesis neither confirm nor refute this hypothesis. Certainly, it has been established that a robot may use its own interaction history to generate appropriate action, and that this history is dynamic (through *merging*, *forgetting*, and - depending on implementation - update of the value of previous experience) and so should change as circumstances change. However, development in terms of scaffolding new behaviour on top of old and learning increasingly complex behaviour patterns has not been addressed in these experiments, and so such a property of the interaction history cannot be confirmed until further experiments are conducted (see "Future Directions", Section 10.3).

10.1.2 Summary of Contributions

The main contributions of this thesis are that it:

1. Defines “Interaction History” from the perspective of autonomous embodied artificial agents;

2. Shows that the information theoretic relationships between a robot’s sensors (exterioceptive, interioceptive and proprioceptive) can be used to autonomously characterize behaviour (*i.e.* distinguish classes of behaviours) and identify behaviours (as being similar to one or another previously experienced behaviour or behaviour class);

3. Defines the Average Information Distance as a measure of sensory relations;

4. Operationalizes the meaning of “experience” from the perspective of embodied artificial agents and robots;

5. Introduces, validates and applies the *experience metric* (an information theoretic measure) to comparison of experiences in robots, and shows that distances between experiences with low values of the metric correspond to experiences that are similar as judged by an external observer;

6. Develops techniques for self-construction and modification of a metric space of experiences as a model of a temporally extended remembering/memory for robotic control systems;

7. Demonstrates the operation of an architecture, that chooses actions based on proximity of experiences in a growing metric space, on different robotic and simulated platforms and on different tasks;

8. Introduces “Peekaboo” as a tool for research in early communicative interaction of robots with humans and as a scenario in which ontogenetic development can be studied in robots.

10.2 Issues, Reflections and Implications

This section discusses some of the issues raised during this research and some of the lessons learnt.

10.2.1 Assumptions

During this thesis certain claims for properties of experiences and their comparisons, as well as the approach to basing action on past experience, were made. However these rely on some assumptions. These assumptions and their implications are briefly discussed here.

An important assumption that is made in the underlying interaction history approach is that the environment is predictable at some (temporal) level. That is, it is essentially causal and that it is not random. That this assumption holds is essential for past experience to be a guide to future experience. However, it is not required to be completely deterministic as the stochasticity in the approach allows for varied responses. Nor is it necessary for the environment to exhibit the same unchanging dynamics as the system is adaptive and can adapt to changing environmental conditions.

It is usually assumed in the calculation of entropy of a random variable that it is *stationary* (*i.e.* its mean, and statistical variance are constant over the length of the series). It is likely however, that random variables estimated from the changing sensorimotor sensor time-series of a robot will not be stationary over the whole time-series but will exhibit local stationarity. Thus calculation of entropy for shorter horizon length experiences will be valid. Nevertheless, should this not be the case (for instance, where the horizon is much longer than windows of stationarity of the interaction environment), it is not an issue for the metric space of experiences as, in this case, the entropy calculation will give an "average" entropy and this is sufficient for the practical comparison of experiences.

10.2.2 Characteristics of Experience Space and Metrics

The first sets of experiments in Chapter 4 involving the Aibo robot executing simple behaviours show how the changing informational relationships between sensors can be used to characterize those behaviours from a grounded sensorimotor perspective. However, the question remained as to what level of distinction could be drawn from a trajectory in a 2-dimensional space as the difference between behaviours becomes less well defined. The extension to the experience metric in Chapter 5 addresses this issue and shows how behaviours can be distinguished to the extent that a motion of a ball can be predicted from previous continuous experiences. These experiments demonstrate that an anticipatory mechanism that operates from continuous experience is a possibility.

The results of these experiments suggest the following properties of the metric space of experiences:

- the distance between sensorimotor experiences in the metric space reflects their subjective similarity;
- proximity in the metric space is not dependant on exact matching between sensorimotor timeseries, but instead depends on statistical informational similarity of a sensory stream with the same stream in the past, summed over all sensors;
- the metric space can be continually and incrementally constructed directly from sensorimotor experience.

10.2.3 Interaction History Architecture (IHA) Experiments

The interaction scenarios reported in Chapters 7,8 and 9 have a number of limitations which were relaxed as the experiments increased in complexity. Having established some of the capabilities of metric space of experiences in the ball-prediction experiments (Section 5.4) , the IHA was demonstrated successfully on a benchmark delayed-response task in Section 7.3 which offered environmental interaction but was limited in that there was no interaction partner. The second scenario, “peekaboo”, introduced in Chapter 8, is considerably more complex, requiring that the robot developed behaviour sensitive to environment and timing to achieve success. However, for the purpose of testing the dependence on horizon length, the environment was simplified, using a static image of a face in place of the interaction partner. In Chapter 9 however a full interactive peekaboo scenario was employed that had a human interaction partner that could provide feedback both by showing and hiding their own face as well as through audio responses and calls. The robot also had the ability to feedback its current reward state to the interaction partner through facial expressions.

10.2.4 Development or Just Learning?

A reasonable question to ask, given that something akin to *physical* development in an animal does not occur in these experimental demonstrations, is: Is this development, or is it just learning? To answer this, let us examine the important facets of development as identified by Lungarella et al. (2003): (a) Incremental (b) Importance of Constraints (c) Self-Organizing (d) Self-Exploration (e) Spontaneous Activity (f) Prospective Control (g) Categorization, Sensorimotor Coordination (h) Value Systems (i) Social Interaction. It is clear that what is being described here is something that is able to not only learn, but also structure its learning over time, increasing its capabilities over time and building (scaffolding) new learning on previously mastered tasks. Development involves a general form of self-organized, unsupervised, open-ended learning, where goals and motivations drive the agent towards better and better coupling with its environment. The interaction history architecture put forward by this thesis meets most if not all of these important facets of a developmental architecture, at least at a rudimentary level. However, the implemented and demonstrated solutions are only a step along the way, and still, for instance, do not have the constraint unfolding capability and progressive drives and value (motivation) systems required for true development. Therefore, even though the mastery of individual tasks may be called “learning”, these are just snapshots of a more wide-ranging developmental activity that the system is capable of due to the dynamic capacity for remembering of the history of interactions that lies at the heart of the system.

10.2.5 Action Correspondence

An important issue in any interaction is how actions and behaviour of other agents can be both recognized and elicited. Humans can understand (give meaning to and ground with reference to own actions) the physical actions and behaviour of others as well as socially motivated action. In this thesis, the way that the robot can elicit social communicative behaviour through their own gestures in the peekaboo game scenario is an example of this. The interaction history architecture at the moment relies on generic visual sensors and the special "face-detection" sensor, combined with the temporally extended statistical model built up in the interaction history, to provide information about the other party in the interaction.

However, there are ways of improving this and one such is to consider the mirror neuron system (Rizzolatti et al., 1996). Mirror neurons in the primate cortex have been shown to fire both when an action is executed and when the same action is seen in others. There is an argument that this shows that there is a neural basis for social interaction. For the purposes of an interaction history, if internally constructed "meta-sensors"/"meta-actuators" could be conceived that would offer the same functionality of a mirror neuron system, then their inclusion in the metric space of experience would greatly enhance the ability of the system to develop action capabilities that took into account the actions of other people and robots in the environment.

One way that such a system could be developed through interaction as part of the interaction history, rather than through an explicit design and creation of a mirror-neuron system, is to give the agent an early developmental drive for imitation. Such a drive would quickly start building up correspondences between certain sequences of image sensor signals and the agent's own motor systems. However, to make full use of this, enhancements and modifications would be needed to the architecture, particularly in the area of separation of motor and sensor experience systems and how action is generated.

10.2.6 Applications in and Outside Robotics

The immediate application for the interaction history architecture is as a developmental architecture for autonomous robots operating in complex, incomplete-information environments that require an adaptive flexibility to cope with different scenarios as well as a level of plasticity so that learning and development can be scaffolded on previous capabilities and experience.

Other applications are any in which action policies need to be adapted and developed based on time-series data. An example might be activating alarm systems in the prediction of severe weather conditions. Another might be in advising trading policy dependent on market and economic data. However, in these systems, accuracy is important and so a refined version of the system would be needed.

Another application may be in assistive technologies that learn appropriate actions depending on a wide-range of sensor input and observed patterns of behaviour. Sensor input might be room temperature, the switching on and off of electrical items, *etc.*, and actions might be to switch on the lights in anticipation of the arrival home of the resident.

10.2.7 Emergent Classes of Experience

The possibility of creation and use of grounded emergent categories of experience was discussed in Section 6.3.3 and demonstrated in the humanoid peekaboo experiment of Chapter 9. In the peekaboo experiment with a humanoid robot, it was observed that a class of experiences emerged during the interaction that was associated with a next action of "hide". It can be said then, that a class of experiences had emerged, grounded in the sensorimotor interaction of the agent, that "represented" that it should next execute the "hide" action.

Thus, merged experiences combined with action present the possibility of truly grounded representations. This is an important consequence of having an incrementally constructed interaction history, and is in-fact essential if that history is to be practical. The applications of such grounded representations are wide-reaching. One speculative example would be a capability of forming categories of objects that depended on appearance as well as how they were interacted with (affordances). This learning of affordances could also be combined with verbal (audio) representations, leading to an "understanding" of meaning of "words" and proto-language.

10.3 Future Directions

As mentioned in the introduction, developmental AI is a new field, and as such appropriate tools for study are still to be developed. The peekaboo scenario is a contribution to this, and has also since been used as an experimental scenario in other cognitive robotics research (Ogino, Ooide, Watanabe and Asada, 2007), however, further scenarios where development (ontogeny) can be both demonstrated and measured are required. The scenarios currently demonstrated are complex, involving a human partner, but limited in terms of the potential to demonstrate wide-reaching ontogenetic development.

The next important area is the requirement for an action system that can change and grow, incorporating new actions and abilities, and refining old ones is essential to allow more open-ended development. Similarly, the structures required to model other agent's actions in comparison to one's own is an area where important progress can be made starting with this architecture as a basis.

Another area for future research is to understand the properties of the metric space of experience. A better grasp of the mathematical and topological structure of the space may lead to more efficient implementations, especially regarding experience recognition and the emergence of 'prototypical' experiences and activities and dimension reduction in the space.

In terms of technical challenges, a demonstrated long-term capability to create and maintain a metric space for the life-time of an agent is required. The interaction history architecture has been shown to run in real-time for over an hour in the delayed response task, but needs to be able to be self-maintaining for much longer periods. To cope with the vast number of experiences, mechanisms such as forgetting and merging of experiences are required. This may be done on-line as was demonstrated in the experiments in Chapter 9, and/or it may be possible to further consolidate experiences in a "sleeping state".

While the current implementation of the metric space and the associated action architecture is currently practical in terms of demonstrating underlying principles, an alternative approach is worth considering. A more biologically plausible configuration is certainly conceivable. A possible future direction would be to implement an artificial neural network that uses the basic ideas of the interaction history architecture. A system can be envisaged that had many overlapping associative networks that were able to store patterns of sensory activity (see for example Vogel, 2005; Shanahan, 2006), and these be connected hierarchically using systems of "information distance" neurons. Each of these special neurons would continually output the information distance between the stream of data at its inputs, thus recognizing when current sensory input was similar to a memory of that sensory input triggered by the re-experiencing mediated by the associative networks. These neurons then feedback into action control areas to close the perception action loop. However, it is not completely clear how temporally extended experience could be captured and employed in such a system.

Further research work which is already underway will see the Interaction History Architecture incorporated as a higher-level behaviour advice module in the open-source robot iCub built by the European FP6 RobotCub project (http://www.robotcub.org/). The iCub robot has an order of magnitude more sensors and has complex control structures, behaviour modules and reactive processes that allow dynamic actions such as reaching, grasping, crawling and drumming, and is therefore a challenge in terms of computing requirements as well as generating complex behaviour and action advice. This complexity however offers interesting possibilities for ontogenetic development to be studied further in the context of the interaction history.

10.4 Conclusion

This thesis puts forward a computational framework that can be used by embodied artificial agents (and in particular autonomous robots) for ontogenetic development. The temporal horizon of an agent is extended so that past experience can be self-organized into a developing structure that can be used to anticipate the future and act appropriately in environments where state information is incomplete, such as the social environment. The Crutchfield-Rényi information metric is used as the basis for the experience metric to compare sensor time-series modelled as random variables, and was demonstrated to be able to characterize and identify time-extended behaviour and help in selecting actions for robots and agents operating over a broad temporal horizon, *i.e.* requiring episodic memory. A metric space consisting of sensorimotor "experiences" was presented and was also demonstrated to reflect subjective ideas of similarity of behaviour in the proximity of experiences in the space. Capabilities afforded by using the metric-space were combined with a reinforcement-learning paradigm using temporally extended experience, not state, to create an architecture that could develop complex time-dependent action capabilities. Demonstrations on various robotic platforms in various scenarios indicate that this may be a promising approach to ontogenetic development in robots.

Appendix A

Robot Sensors

Table A.1: AIBO ERS-7 Sensors

Sensor	Min	Max	Description
legRF1	-134	120	Right fore leg
legRF2	-9	91	Right fore leg
legRF3	-29	119	Right fore leg
legRH1	-134	120	Right hind leg
legRH2	-9	91	Right hind leg
legRH3	-29	119	Right hind leg
legLF1	-120	134	Left fore leg
legLF2	-9	91	Left fore leg
legLF3	-29	119	Left fore leg
legLH1	-120	134	Left hind leg
legLH2	-9	91	Left hind leg
legLH3	-29	119	Left hind leg
neck	-79	2	Neck tilt1
headTilt	-16	44	Neck tilt2
headPan	-91	91	Head pan
tailPan	-59	59	Tail pan
tailTilt	2	63	Tail tilt
mouth	-58	-3	Mouth
pawLF	0	1	Left fore leg,paw sensor
pawLH	0	1	Left hind leg,paw sensor
pawRF	0	1	Right fore leg,paw sensor
pawRH	0	1	Right hind leg,paw sensor
accelX	-19.6	19.6	Acceleration sensor(front-back)
accelY	-19.6	19.6	Acceleration sensor(right-left)
accelZ	-19.6	19.6	Acceleration sensor(up-down)
chinSensor	0	1	Chin sensor
backSensorF	0	60	Back sensor(front)
backSensorM	0	60	Back sensor(middle)
backSensorR	0	60	Back sensor(rear)
headSensor	0	35	Head sensor
distanceChest	19	90	Chest distance sensor
distanceNear	5.7	50	Head distance sensor(near)
distanceFar	20	150	Head distance sensor(far)

Table A.2: Simulated Pioneer Sensors

Sensor	Min	Max	Description
posX	-4.00	4.00	Horizontal Position
posY	0.00	8.00	Vertical Position
Yaw	-2.00	2.00	Direction
minL	0.00	8.00	Min of left sonar ranger values
minR	0.00	8.00	Min of right sonar ranger values
rangeL	0.00	8.00	Average of Left sonar group
rangeF	0.00	8.00	Average of Front sonar group
rangeR	0.00	8.00	Average of Right sonar group
blobL	0	4661	Left blob detector
blobR	0	4661	Right blob detector
reward	0	4	Resulting Reward sensor
action	0	3	Action

Table A.3: Kaspar2 Sensors

Sensor	Min	Max	Description
HEAD_LR	740	2100	Head Pan Left-Right
HEAD_UD_L	640	2200	Left Neck Elevation Motor
HEAD_UD_R	820	2200	Right Neck Elevation Motor
EYES_LR	930	2060	Eyes Pan Left-Right
EYES_UD	980	1920	Eyes Up-Down
EYELIDS	1150	1700	Eyelids Open-Close
MOUTH_OPEN	600	1730	Mouth Open
MOUTH_SMILE	600	2200	Mouth Corner Elevation
ARM_R_1	650	2200	Right Shoulder Rotate
ARM_R_2	1090	2200	Right Shoulder Elevate
ARM_R_3	910	2200	Right Arm Rotate
ARM_R_4	600	2200	Right Elbow Bend
ARM_R_5	780	2200	Right Forearm Rotate
ARM_L_1	600	2200	Left Shoulder Rotate
ARM_L_2	780	2000	Left Shoulder Elevate
ARM_L_3	600	2140	Left Arm Rotate
ARM_L_4	600	2200	Left Elbow Bend
ARM_L_5	600	2200	Left Forearm Rotate
FACE	0	1	Face detection signal
SOUNDS	0	1	Sum of Sound Amplitudes for Timestep
reward	0	1	Resulting Reward sensor
action	0	20	Action

Appendix B

Publications

The research work of this thesis has contributed to 10 publications, 9 in peer-reviewed publications including one journal, one book-chapter and 7 publications in conference proceedings. I am first author in the majority of the work with my co-authors being my research supervisors: Professor Chrystopher L. Nehaniv, Professor Kerstin Dautenhahn and Dr. René te Boekhorst. The book chapter, still in press, showcases this work along with the work of Prof. Nehaniv and Dr. Lars Olsson on the related theme of the use of information theory to learn sensory relations and mappings using uninterpreted sensory data.

B.1 Chronological List of Publications

(1) 2005 : Type: CONFERENCE

Authors: N. A. Mirza, C. L. Nehaniv, K. Dautenhahn, and R. te Boekhorst.

Title: Using sensory-motor phase-plots to characterise robot-environment interactions.

Details: 'Proc. of 6th IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA2005)', pp. 581–586.

Notes: Contains description of "AID Phase-plots", referred to in this thesis as AID *vs.* AID plots, and experiments with characterization and identification of behaviour of an Aibo (Chapter 4).

(2) 2005 : Type: TECHNICAL REPORT

Authors: N. A. Mirza, C. L. Nehaniv, R. te Boekhorst and K. Dautenhahn.

Title: Robot self-characterisation of experience using trajectories in sensory-motor phase space.

Details: Technical Report 424, University of Hertfordshire, Computer Science, 2005.

Notes: Introduces the box-counting fractal dimension estimation as an additional morphometric for characterizing AID *vs.* AID plots (Section 4.4.2). Introduces the segmentation method described in Section 4.6.

(3) 2005 : Type: CONFERENCE

Authors: N. A. Mirza, C. L. Nehaniv, K. Dautenhahn, and R. te Boekhorst.

Title: Using temporal information distance to locate sensorimotor experience in a metric space.

Details: 'Proc. of 2005 IEEE Congress on Evolutionary Computation (CEC2005)', Vol. 1, IEEE Press, Edinburgh, Scotland, UK, pp. 150–157.

Notes: First description of Sensorimotor Experience as described in this thesis, and the experience metric, but referred to as the "total temporal information distance". Discusses Local and Global views, and shows results of the metric applied to an Aibo wandering in an arena. See Sections 5.2 and 5.3.2.

(4) 2005 : Type: CONFERENCE

Authors: C. L. Nehaniv.

Title: Sensorimotor Experience and Its Metrics: Informational Geometry and the Temporal Horizon.

Details: 'Proc. of 2005 IEEE Congress on Evolutionary Computation (CEC2005)', Vol. 1, IEEE Press, Edinburgh, Scotland, UK, pp. 142-149.

Notes: Though this does not contain my name on the author list, the work presented includes results from experiments designed and conducted by myself. See Section 5.2.

(5) 2005 : Type: CONFERENCE (Poster and Short Procedings Article)

Authors: N. A. Mirza, C. L. Nehaniv, R. te Boekhorst and K. Dautenhahn.

Title: Robot self-characterisation of experience using trajectories in sensory-motor phase space.

Details: 'Proc. of Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems (EpiRob2005)', Lund University Cognitive Studies, pp. 143–144.

Notes: Box-counting fractal dimension estimation as an additional morphometric for characterizing AID *vs.* AID plots (Section 4.4.2).

(6) 2006 : Type: CONFERENCE

Authors: C. L. Nehaniv, N. A. Mirza, K. Dautenhahn, and R. te Boekhorst.

Title: Extending the temporal horizon of autonomous robots.

Details: Proc. of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE2005), pages 389-395. Springer, 2006.

Notes: Summaries work on experience metrics and local/global picture.

(7) 2006 : Type: CONFERENCE

Authors: N. A. Mirza, C. L. Nehaniv, K. Dautenhahn, and R. te Boekhorst.

Title: Interaction histories: From experience to action and back again.

Details: In Proceedings of the 5th IEEE International Conference on Development and Learning (ICDL 2006), Bloomington, IN, USA, 2006.

Notes: Introduces Interaction History Architecture and Ball Prediction experiments (Chapter 7) and early Peekaboo experiments (Chapter 8, and

Section 8.3.1).

(8) 2006 : Type: CONFERENCE

Authors: N. A. Mirza, C. L. Nehaniv, K. Dautenhahn, and R. te Boekhorst.

Title: Peekaboo: Effect of experience length on the interaction history driven ontogeny of a robot.

Details: Proceedings the of 6th International Conference on Epigenetic Robotics, pages 71-78, Paris, France, 20-22 September 2006. Lund University Cognitive Studies.

Notes: Experiments with Peekaboo investigating Horizon length and number of bins - see Section 8.3.2.

(9) 2007 : Type: JOURNAL

Authors: N. A. Mirza, C. L. Nehaniv, K. Dautenhahn, and R. te Boekhorst.

Title: Grounded sensorimotor interaction histories in an information theoretic metric space for robot ontogeny.

Details: Journal of Adaptive Behaviour: Special Issue, 2007. Volume 15, Number 2, pages 167-187. SAGE Publications.

Notes: Brings together work on experience, experience metrics and peekaboo experiments.

(10) 2007 : Type: BOOK CHAPTER

Authors: C. L. Nehaniv, N. A. Mirza, L. Olsson.

Title: Development via Information Self-structuring of Sensorimotor Experience and Interaction

Details: 50 Years of Artificial Intelligence: Essays Dedicated to the 50th Anniversary of Artificial Intelligence. Volume 4850/2007, pages 87-98. Springer.

Notes: Experience metrics and brief description of Aibo experiments.

Appendix C

Kaspar Peekaboo Results

This appendix contains the results from 15 experimental runs of the history architecture running on the KasparII robot. The human interaction partner either encourages Peekaboo, another action or gives no encouragement at all. These results are summarised in Chapter 9.

Table C.1: Actions executed (consolidated): Run d0032

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	16	2	0	1	1	16	4	1	0	0	1	0	1	0	43
chosen	33	0	0	0	16	21	0	2	0	0	0	0	1	0	73
both	49	2	0	1	17	37	4	3	0	0	1	0	2	0	116

Table C.2: Actions executed (primary): Run d0032

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	20.00	0.00	10.00	40.00	10.00	10.00	0.00	10.00	0.00	100.00
chosen :	0.00	0.00	84.21	0.00	10.53	0.00	0.00	5.26	0.00	100.00
both :	6.90	0.00	58.62	13.79	10.34	3.45	0.00	6.90	0.00	100.00
Percentage Random v Chosen Actions										
random	100.00	0.0	5.88	100.00	33.33	100.00	0.0	50.00	0.0	
chosen	0.00	0.0	94.12	0.00	66.67	0.00	0.0	50.00	0.0	
Overall Chosen %:	0.00	0.0	55.17	0.00	6.90	0.00	0.0	3.45	0.0	65.52

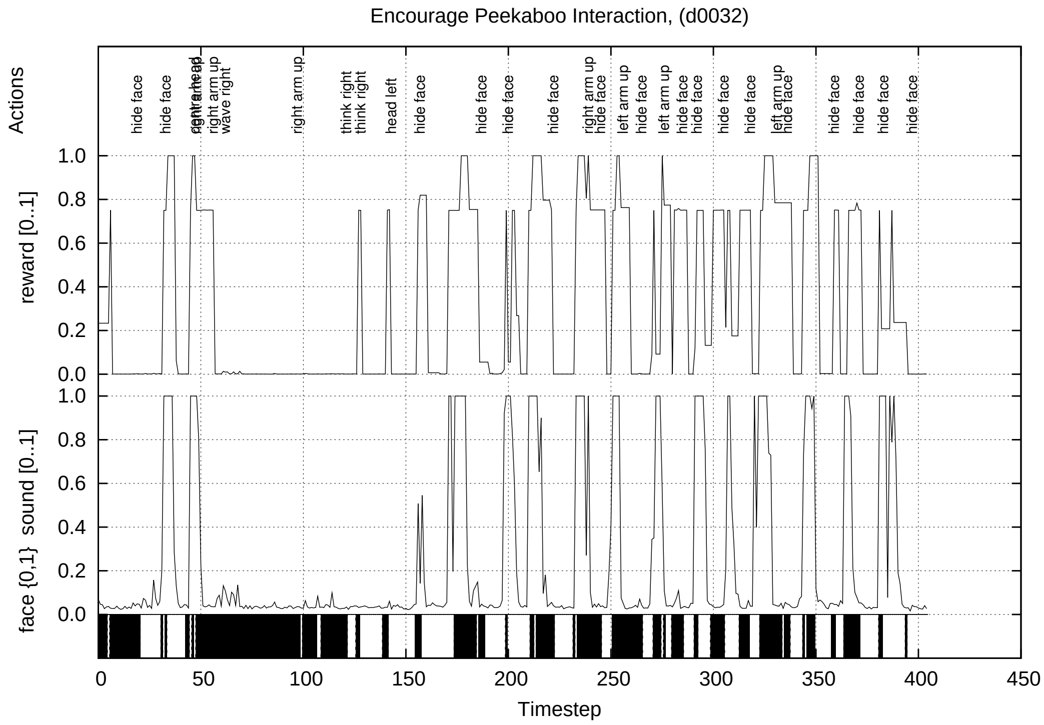


Figure C.1:

Table C.3: Actions executed (consolidated): Run d0033

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	27	0	0	1	3	17	5	3	1	2	1	0	4	2	66
chosen	9	0	0	0	14	3	0	0	0	0	0	0	1	1	28
both	36	0	0	1	17	20	5	3	1	2	1	0	5	3	94

Table C.4: Actions executed (primary): Run d0033

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	0.00	0.00	16.67	27.78	16.67	5.56	0.00	22.22	11.11	100.00
chosen :	0.00	0.00	87.50	0.00	0.00	0.00	0.00	6.25	6.25	100.00
both :	0.00	0.00	50.00	14.71	8.82	2.94	0.00	14.71	8.82	100.00
Percentage Random v Chosen Actions										
random	0.0	0.0	17.65	100.00	100.00	100.00	0.0	80.00	66.67	
chosen	0.0	0.0	82.35	0.00	0.00	0.00	0.0	20.00	33.33	
Overall Chosen %:	0.0	0.0	41.18	0.00	0.00	0.00	0.0	2.94	2.94	47.06

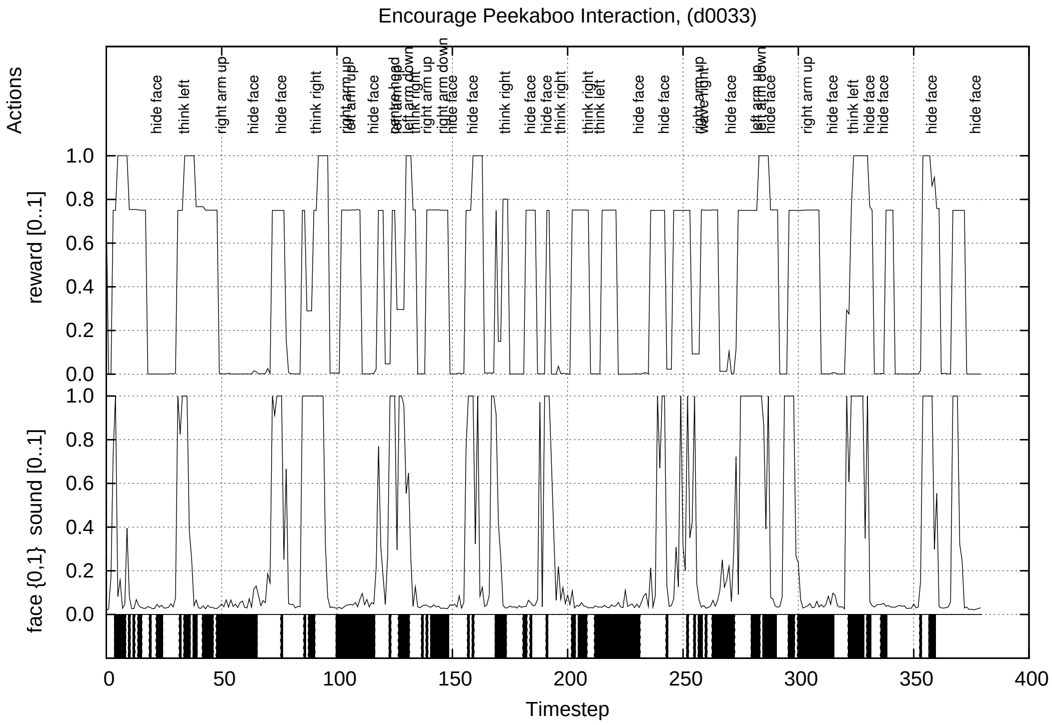


Figure C.2:

Table C.5: Actions executed (consolidated): Run d0034

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	7	1	0	5	2	9	4	2	2	1	2	0	0	2	37
chosen	60	1	0	0	0	0	1	0	1	0	0	0	0	0	63
both	67	2	0	5	2	9	5	2	3	1	2	0	0	2	100

Table C.6: Actions executed (primary): Run d0034

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	7.69	0.00	15.38	30.77	15.38	15.38	0.00	0.00	15.38	100.00
chosen :	50.00	0.00	0.00	50.00	0.00	0.00	0.00	0.00	0.00	100.00
both :	13.33	0.00	13.33	33.33	13.33	13.33	0.00	0.00	13.33	100.00
Percentage Random v Chosen Actions										
random	50.00	0.0	100.00	80.00	100.00	100.00	0.0	0.0	100.00	
chosen	50.00	0.0	0.00	20.00	0.00	0.00	0.0	0.0	0.00	
Overall Chosen %:	6.67	0.0	0.00	6.67	0.00	0.00	0.0	0.0	0.00	13.33

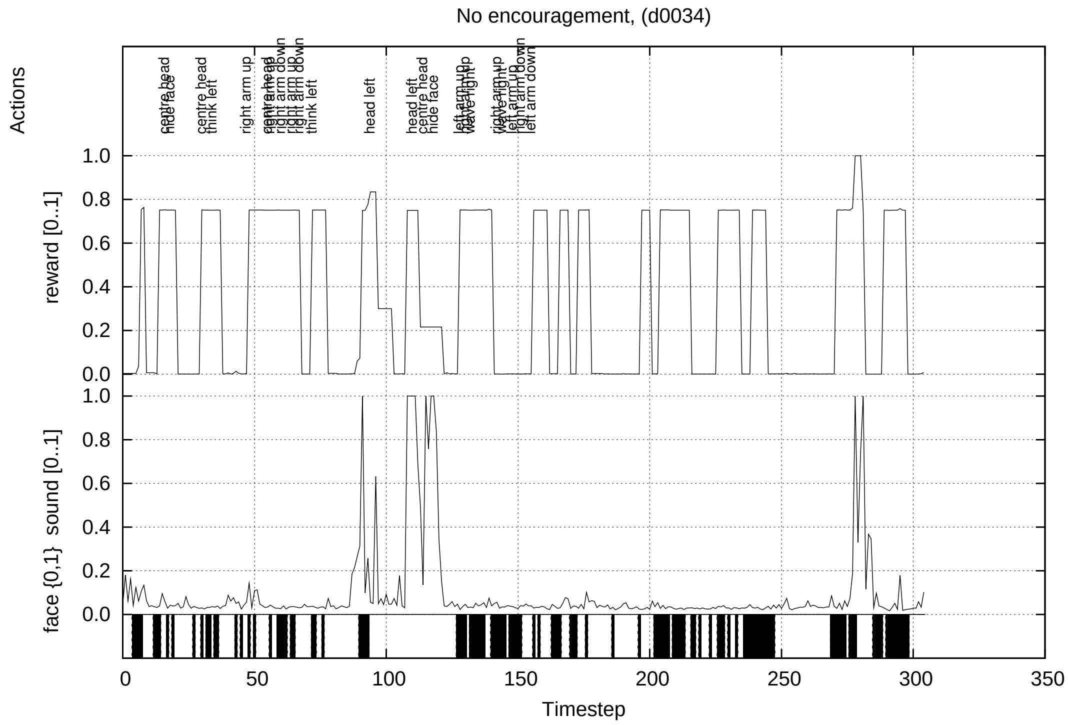


Figure C.3:

Table C.7: Actions executed (consolidated): Run d0035

	0	3	4	5	6	7	8	9	10	11	12	13	14	15	Tot
	Rst	HL	HR	HF	Hid	NA	RAU	LAU	RAD	LAD	RAW	LAW	TR	TL	
random	17	5	2	9	4	15	3	1	1	0	1	0	1	1	60
chosen	13	15	2	11	6	2	0	0	0	0	0	0	0	0	49
both	30	20	4	20	10	17	3	1	1	0	1	0	1	1	109

Table C.8: Actions executed (primary): Run d0035

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	27.78	11.11	22.22	16.67	5.56	5.56	0.00	5.56	5.56	100.00
chosen :	65.22	8.70	26.09	0.00	0.00	0.00	0.00	0.00	0.00	100.00
both :	48.78	9.76	24.39	7.32	2.44	2.44	0.00	2.44	2.44	100.00
Percentage Random v Chosen Actions										
random	25.00	50.00	40.00	100.00	100.00	100.00	0.0	100.00	100.00	
chosen	75.00	50.00	60.00	0.00	0.00	0.00	0.0	0.00	0.00	
Overall Chosen %:	36.59	4.88	14.63	0.00	0.00	0.00	0.0	0.00	0.00	56.10

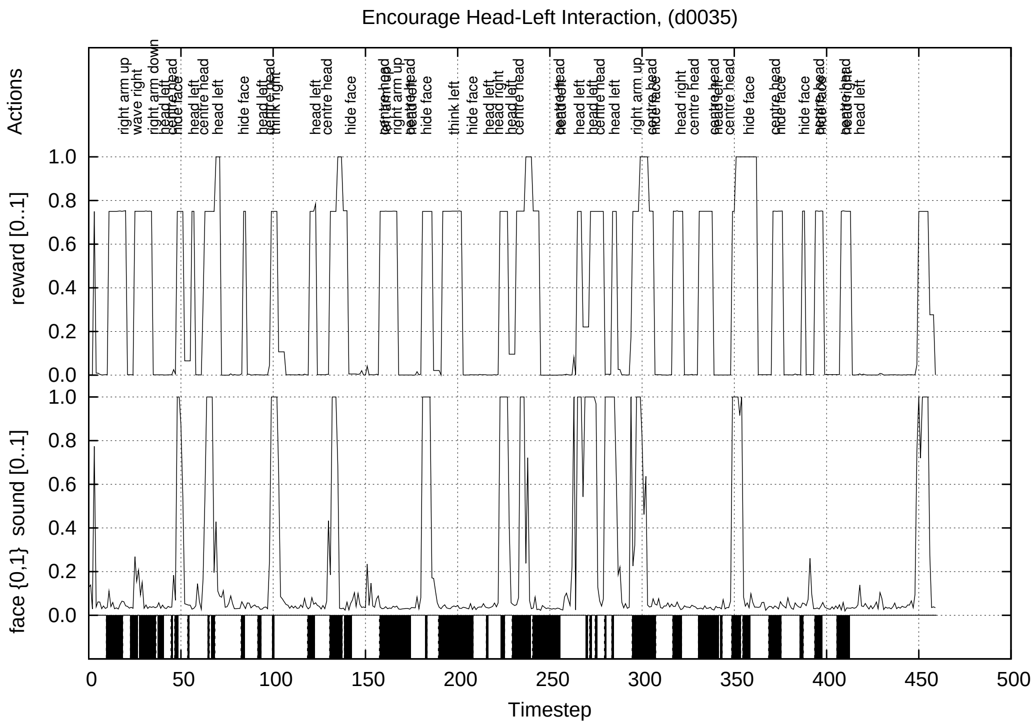


Figure C.4:

Table C.9: Actions executed (consolidated): Run d0036

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	12	3	1	4	3	14	1	2	1	2	0	0	1	0	44
chosen	4	0	0	0	8	7	0	0	0	0	0	0	0	0	19
both	16	3	1	4	11	21	1	2	1	2	0	0	1	0	63

Table C.10: Actions executed (primary): Run d0036

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	27.27	9.09	27.27	9.09	18.18	0.00	0.00	9.09	0.00	100.00
chosen :	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
both :	15.79	5.26	57.89	5.26	10.53	0.00	0.00	5.26	0.00	100.00
Percentage Random v Chosen Actions										
random	100.00	100.00	27.27	100.00	100.00	0.0	0.0	100.00	0.0	
chosen	0.00	0.00	72.73	0.00	0.00	0.0	0.0	0.00	0.0	
Overall Chosen %:	0.00	0.00	42.11	0.00	0.00	0.0	0.0	0.00	0.0	42.11

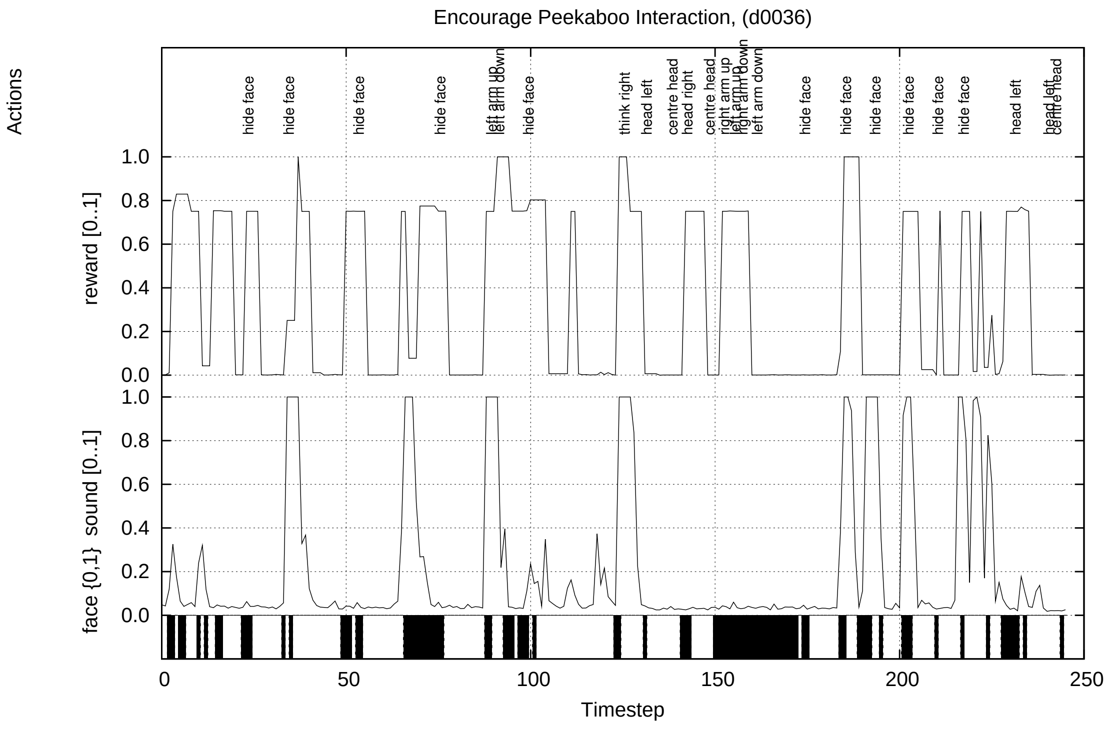


Figure C.5:

Table C.11: Actions executed (consolidated): Run d0037

	0	3	4	5	6	7	8	9	10	11	12	13	14	15	Tot
	Rst	HL	HR	HF	Hid	NA	RAU	LAU	RAD	LAD	RAW	LAW	TR	TL	
random	5	1	2	1	1	13	3	1	1	0	1	0	2	1	32
chosen	57	0	0	6	3	0	0	0	0	0	4	0	3	0	73
both	62	1	2	7	4	13	3	1	1	0	5	0	5	1	105

Table C.12: Actions executed (primary): Run d0037

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	8.33	16.67	8.33	25.00	8.33	8.33	0.00	16.67	8.33	100.00
chosen :	0.00	0.00	30.00	0.00	0.00	40.00	0.00	30.00	0.00	100.00
both :	4.55	9.09	18.18	13.64	4.55	22.73	0.00	22.73	4.55	100.00
Percentage Random v Chosen Actions										
random	100.00	100.00	25.00	100.00	100.00	20.00	0.0	40.00	100.00	
chosen	0.00	0.00	75.00	0.00	0.00	80.00	0.0	60.00	0.00	
Overall Chosen %:	0.00	0.00	13.64	0.00	0.00	18.18	0.0	13.64	0.00	45.45

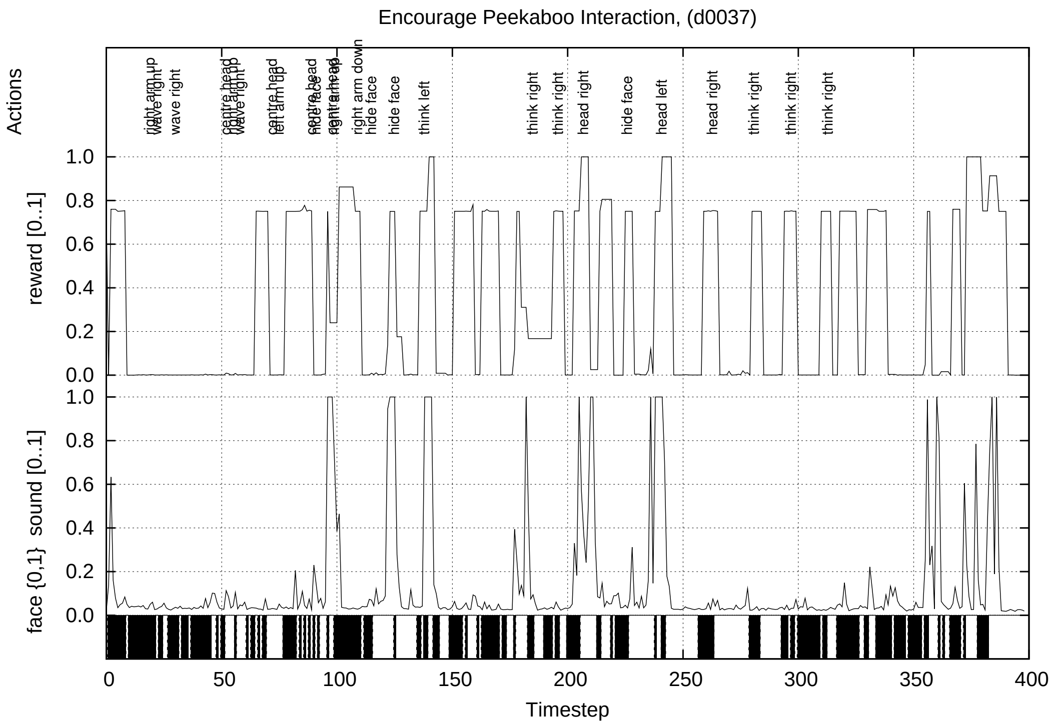


Figure C.6:

Table C.13: Actions executed (consolidated): Run d0038

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	3	0	0	0	0	2	0	1	0	1	0	0	0	2	9
chosen	140	0	0	0	0	22	0	0	0	0	0	0	0	5	167
both	143	0	0	0	0	24	0	1	0	1	0	0	0	7	176

Table C.14: Actions executed (primary): Run d0038

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	0.00	0.00	0.00	0.00	33.33	0.00	0.00	0.00	66.67	100.00
chosen :	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	100.00
both :	0.00	0.00	0.00	0.00	12.50	0.00	0.00	0.00	87.50	100.00
Percentage Random v Chosen Actions										
random	0.0	0.0	0.0	0.0	100.00	0.0	0.0	0.0	28.57	
chosen	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.0	71.43	
Overall Chosen %:	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.0	62.50	62.50

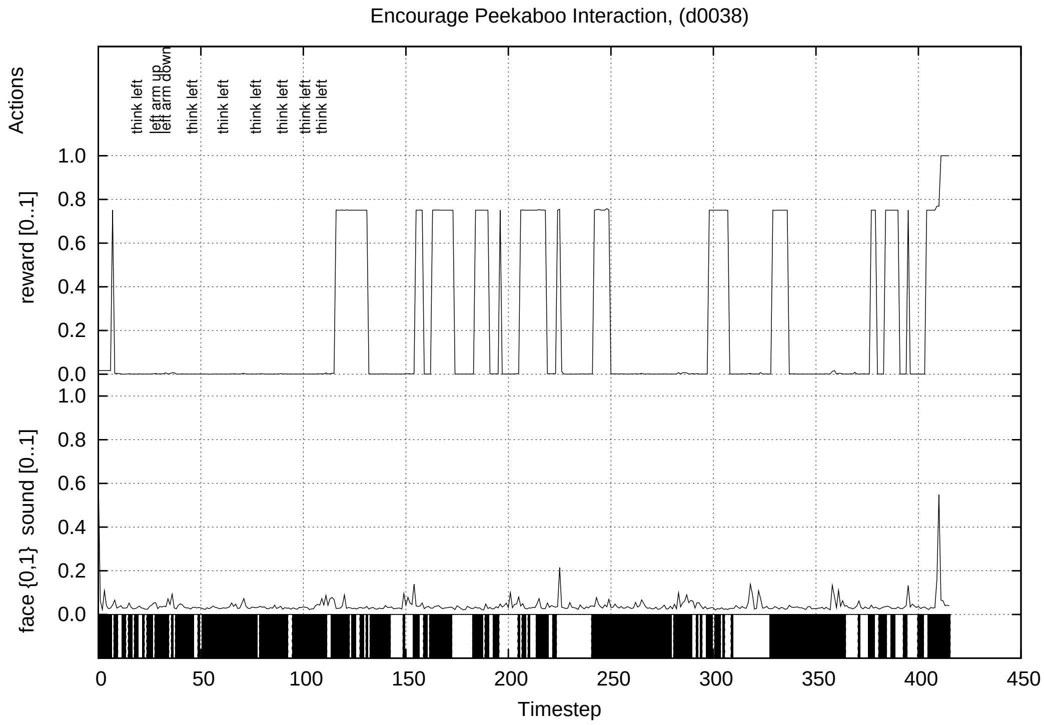


Figure C.7:

Table C.15: Actions executed (consolidated): Run d0039

	0	3	4	5	6	7	8	9	10	11	12	13	14	15	Tot
	Rst	HL	HR	HF	Hid	NA	RAU	LAU	RAD	LAD	RAW	LAW	TR	TL	
random	5	2	0	1	2	3	1	1	0	1	0	0	0	1	17
chosen	4	0	0	0	1	14	0	0	0	0	0	0	0	0	19
both	9	2	0	1	3	17	1	1	0	1	0	0	0	1	36

Table C.16: Actions executed (primary): Run d0039

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	28.57	0.00	28.57	14.29	14.29	0.00	0.00	0.00	14.29	100.00
chosen :	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
both :	25.00	0.00	37.50	12.50	12.50	0.00	0.00	0.00	12.50	100.00
Percentage Random v Chosen Actions										
random	100.00	0.0	66.67	100.00	100.00	0.0	0.0	0.0	100.00	
chosen	0.00	0.0	33.33	0.00	0.00	0.0	0.0	0.0	0.00	
Overall Chosen %:	0.00	0.0	12.50	0.00	0.00	0.0	0.0	0.0	0.00	12.50

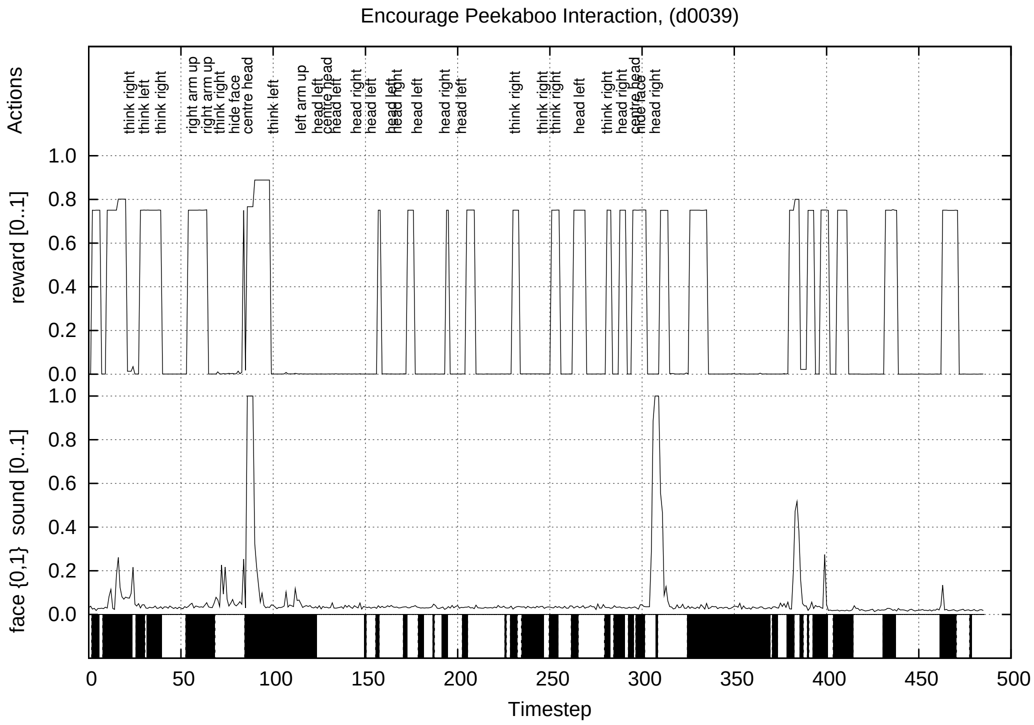


Figure C.8:

Table C.17: Actions executed (consolidated): Run d0041

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	25	14	6	5	4	25	5	6	1	2	0	3	1	4	101
chosen	96	16	6	0	5	10	13	5	0	0	0	0	1	2	154
both	121	30	12	5	9	35	18	11	1	2	0	3	2	6	255

Table C.18: Actions executed (primary): Run d0041

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	32.56	13.95	9.30	11.63	13.95	0.00	6.98	2.33	9.30	100.00
chosen :	33.33	12.50	10.42	27.08	10.42	0.00	0.00	2.08	4.17	100.00
both :	32.97	13.19	9.89	19.78	12.09	0.00	3.30	2.20	6.59	100.00
Percentage Random v Chosen Actions										
random	46.67	50.00	44.44	27.78	54.55	0.0	100.00	50.00	66.67	
chosen	53.33	50.00	55.56	72.22	45.45	0.0	0.00	50.00	33.33	
Overall Chosen %:	17.58	6.59	5.49	14.29	5.49	0.0	0.00	1.10	2.20	52.75

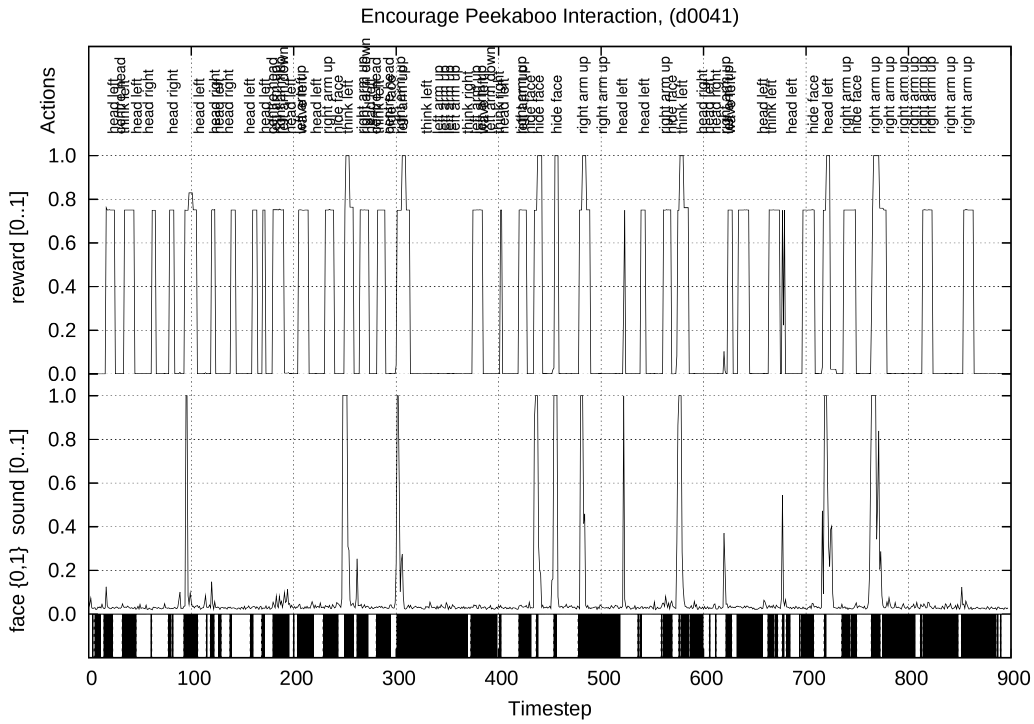


Figure C.9:

Table C.19: Actions executed (consolidated): Run d0042

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	28	6	8	11	4	22	5	3	1	0	0	2	3	1	94
chosen	8	1	5	8	6	13	0	1	0	0	0	16	0	1	59
both	36	7	13	19	10	35	5	4	1	0	0	18	3	2	153

Table C.20: Actions executed (primary): Run d0042

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	18.75	25.00	12.50	15.62	9.38	0.00	6.25	9.38	3.12	100.00
chosen :	3.33	16.67	20.00	0.00	3.33	0.00	53.33	0.00	3.33	100.00
both :	11.29	20.97	16.13	8.06	6.45	0.00	29.03	4.84	3.23	100.00
Percentage Random v Chosen Actions										
random	85.71	61.54	40.00	100.00	75.00	0.0	11.11	100.00	50.00	
chosen	14.29	38.46	60.00	0.00	25.00	0.0	88.89	0.00	50.00	
Overall Chosen %:	1.61	8.06	9.68	0.00	1.61	0.0	25.81	0.00	1.61	48.39

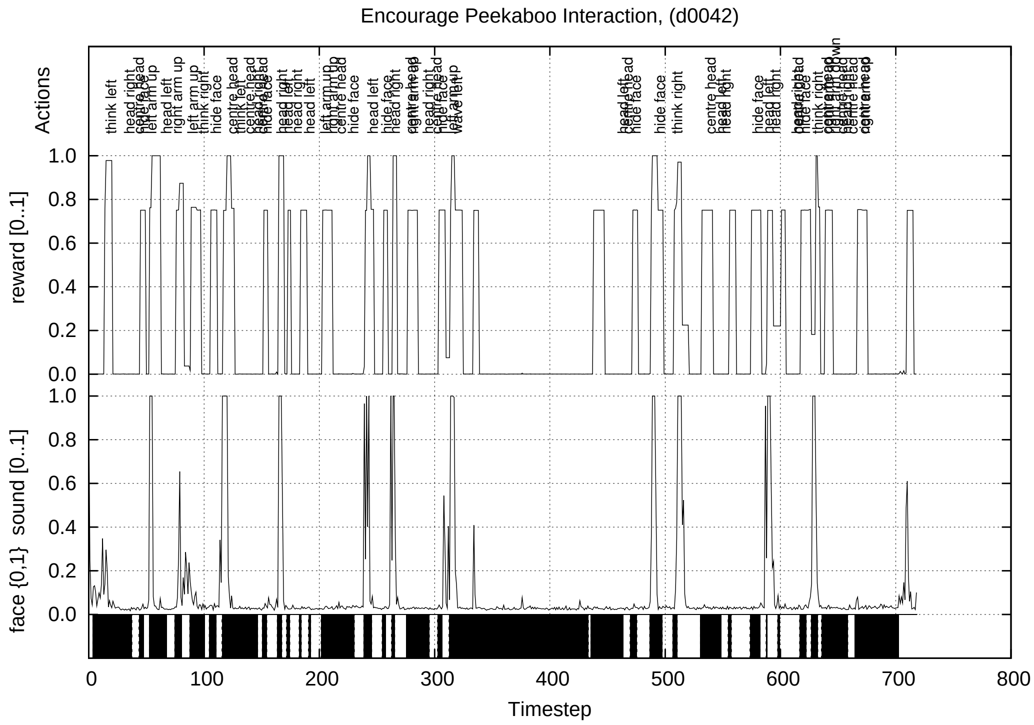


Figure C.10:

Table C.21: Actions executed (consolidated): Run d0043

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	23	11	9	13	1	22	5	4	2	1	3	2	4	5	105
chosen	32	19	26	4	1	11	0	0	0	0	0	0	0	2	95
both	55	30	35	17	2	33	5	4	2	1	3	2	4	7	200

Table C.22: Actions executed (primary): Run d0043

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	25.00	20.45	2.27	11.36	9.09	6.82	4.55	9.09	11.36	100.00
chosen :	39.58	54.17	2.08	0.00	0.00	0.00	0.00	0.00	4.17	100.00
both :	32.61	38.04	2.17	5.43	4.35	3.26	2.17	4.35	7.61	100.00
Percentage Random v Chosen Actions										
random	36.67	25.71	50.00	100.00	100.00	100.00	100.00	100.00	71.43	
chosen	63.33	74.29	50.00	0.00	0.00	0.00	0.00	0.00	28.57	
Overall Chosen %:	20.65	28.26	1.09	0.00	0.00	0.00	0.00	0.00	2.17	52.17

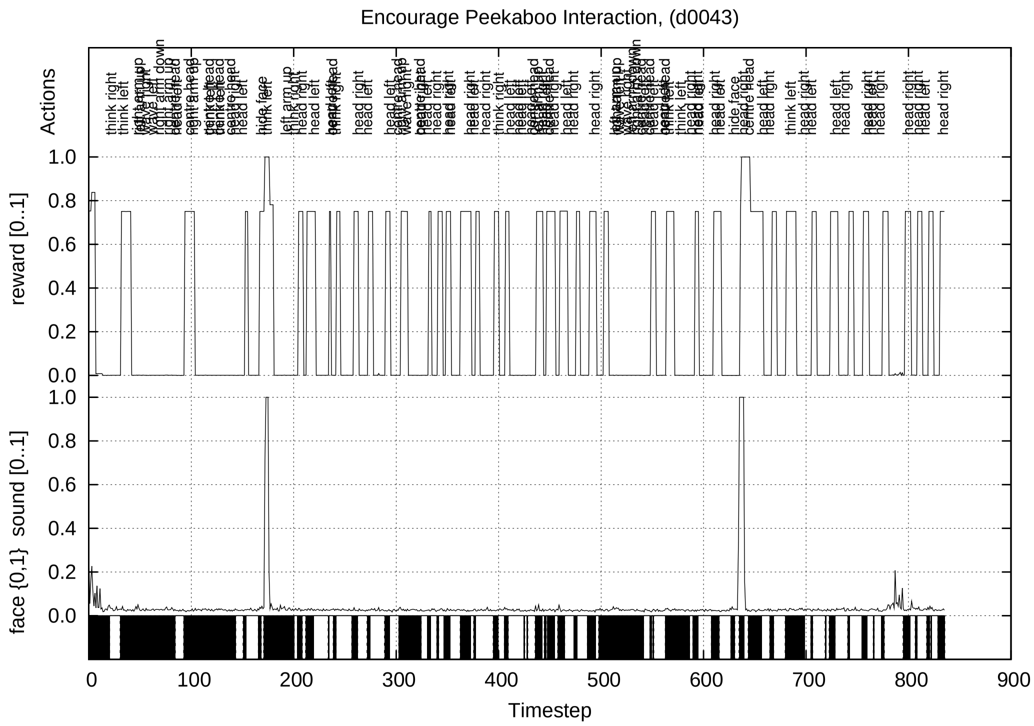


Figure C.11:

Table C.23: Actions executed (consolidated): Run d0044

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	32	9	3	7	3	13	6	4	1	4	0	1	5	2	90
chosen	15	3	1	1	10	4	1	1	0	1	0	0	3	1	41
both	47	12	4	8	13	17	7	5	1	5	0	1	8	3	131

Table C.24: Actions executed (primary): Run d0044

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	27.27	9.09	9.09	18.18	12.12	0.00	3.03	15.15	6.06	100.00
chosen :	15.00	5.00	50.00	5.00	5.00	0.00	0.00	15.00	5.00	100.00
both :	22.64	7.55	24.53	13.21	9.43	0.00	1.89	15.09	5.66	100.00
Percentage Random v Chosen Actions										
random	75.00	75.00	23.08	85.71	80.00	0.0	100.00	62.50	66.67	
chosen	25.00	25.00	76.92	14.29	20.00	0.0	0.00	37.50	33.33	
Overall Chosen %:	5.66	1.89	18.87	1.89	1.89	0.0	0.00	5.66	1.89	37.74

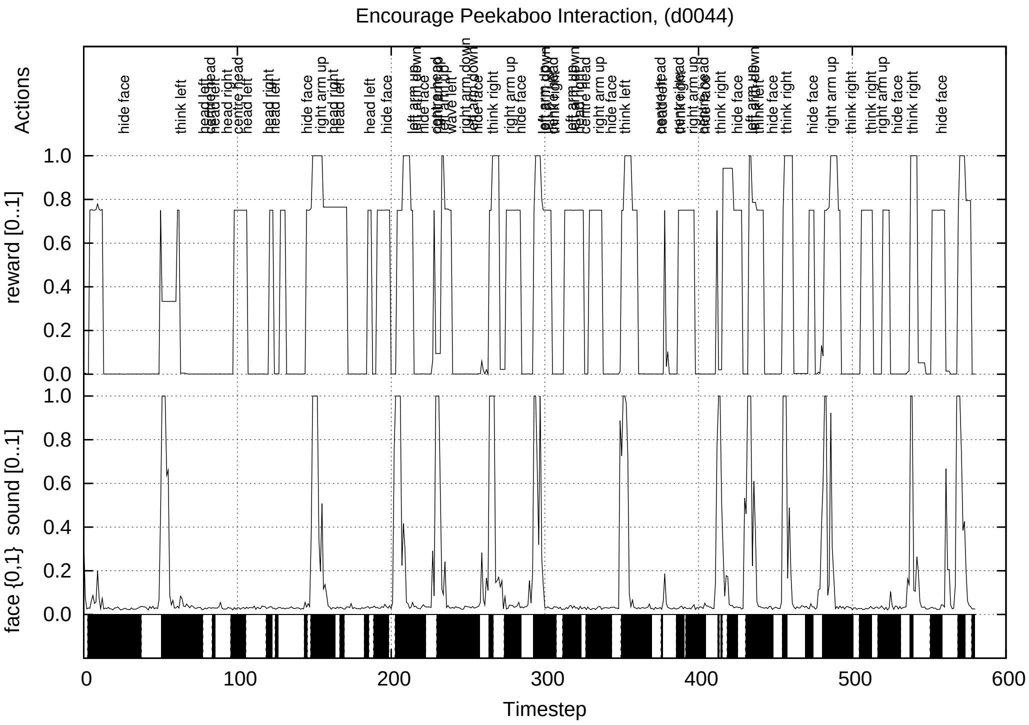


Figure C.12:

Table C.25: Actions executed (consolidated): Run d0045

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	7	4	3	6	3	10	3	8	1	5	2	2	1	1	56
chosen	57	1	0	0	0	32	1	3	0	2	0	0	0	0	96
both	64	5	3	6	3	42	4	11	1	7	2	2	1	1	152

Table C.26: Actions executed (primary): Run d0045

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	14.81	11.11	11.11	11.11	29.63	7.41	7.41	3.70	3.70	100.00
chosen :	20.00	0.00	0.00	20.00	60.00	0.00	0.00	0.00	0.00	100.00
both :	15.62	9.38	9.38	12.50	34.38	6.25	6.25	3.12	3.12	100.00
Percentage Random v Chosen Actions										
random	80.00	100.00	100.00	75.00	72.73	100.00	100.00	100.00	100.00	
chosen	20.00	0.00	0.00	25.00	27.27	0.00	0.00	0.00	0.00	
Overall Chosen %:	3.12	0.00	0.00	3.12	9.38	0.00	0.00	0.00	0.00	15.62

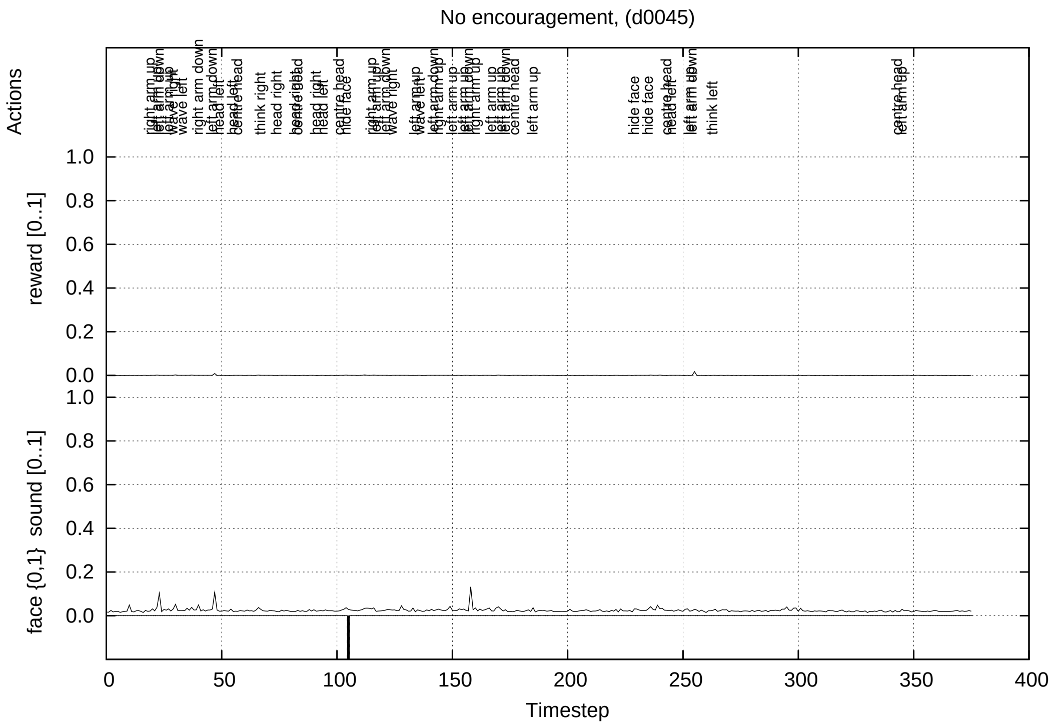


Figure C.13:

Table C.27: Actions executed (consolidated): Run d0046

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	21	6	2	5	3	20	4	6	0	2	1	1	4	3	78
chosen	32	9	0	3	2	19	1	1	0	0	0	30	3	0	100
both	53	15	2	8	5	39	5	7	0	2	1	31	7	3	178

Table C.28: Actions executed (primary): Run d0046

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	20.00	6.67	10.00	13.33	20.00	3.33	3.33	13.33	10.00	100.00
chosen :	19.57	0.00	4.35	2.17	2.17	0.00	65.22	6.52	0.00	100.00
both :	19.74	2.63	6.58	6.58	9.21	1.32	40.79	9.21	3.95	100.00
Percentage Random v Chosen Actions										
random	40.00	100.00	60.00	80.00	85.71	100.00	3.23	57.14	100.00	
chosen	60.00	0.00	40.00	20.00	14.29	0.00	96.77	42.86	0.00	
Overall Chosen %:	11.84	0.00	2.63	1.32	1.32	0.00	39.47	3.95	0.00	60.53

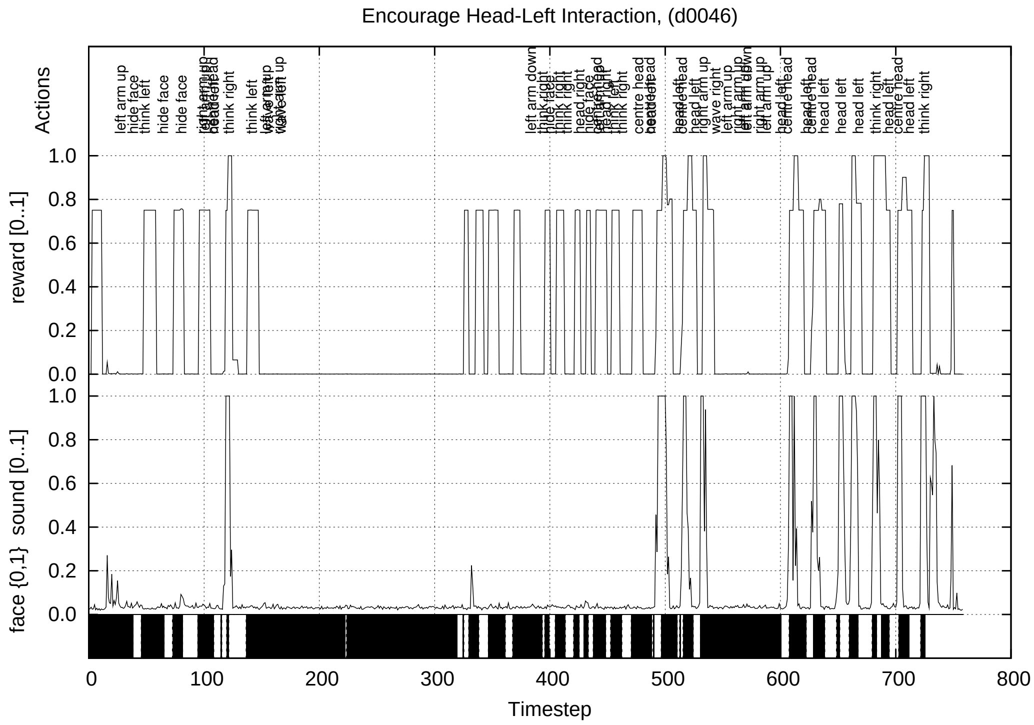


Figure C.14:

Table C.29: Actions executed (consolidated): Run d0049

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	33	14	6	10	4	29	7	6	4	4	1	2	7	6	133
chosen	87	13	9	3	3	25	0	0	0	0	0	0	12	2	154
both	120	27	15	13	7	54	7	6	4	4	1	2	19	8	287

Table C.30: Actions executed (primary): Run d0049

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	26.42	11.32	7.55	13.21	11.32	1.89	3.77	13.21	11.32	100.00
chosen :	33.33	23.08	7.69	0.00	0.00	0.00	0.00	30.77	5.13	100.00
both :	29.35	16.30	7.61	7.61	6.52	1.09	2.17	20.65	8.70	100.00
Percentage Random v Chosen Actions										
random	51.85	40.00	57.14	100.00	100.00	100.00	100.00	36.84	75.00	
chosen	48.15	60.00	42.86	0.00	0.00	0.00	0.00	63.16	25.00	
Overall Chosen %:	14.13	9.78	3.26	0.00	0.00	0.00	0.00	13.04	2.17	42.39

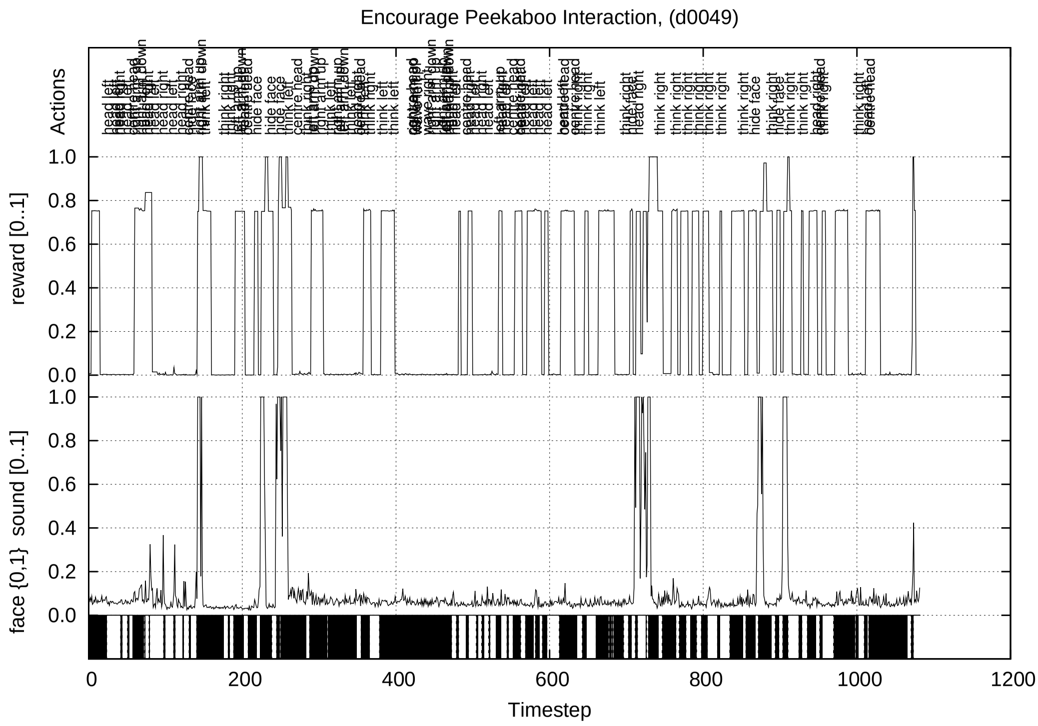


Figure C.15:

Table C.31: Actions executed (consolidated): Run d0050

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	19	7	7	3	2	12	2	3	0	2	0	0	2	0	59
chosen	7	5	0	0	10	2	0	0	0	0	0	0	0	0	24
both	26	12	7	3	12	14	2	3	0	2	0	0	2	0	83

Table C.32: Actions executed (primary): Run d0050

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	30.43	30.43	8.70	8.70	13.04	0.00	0.00	8.70	0.00	100.00
chosen :	33.33	0.00	66.67	0.00	0.00	0.00	0.00	0.00	0.00	100.00
both :	31.58	18.42	31.58	5.26	7.89	0.00	0.00	5.26	0.00	100.00
Percentage Random v Chosen Actions										
random	58.33	100.00	16.67	100.00	100.00	0.0	0.0	100.00	0.0	
chosen	41.67	0.00	83.33	0.00	0.00	0.0	0.0	0.00	0.0	
Overall Chosen %:	13.16	0.00	26.32	0.00	0.00	0.0	0.0	0.00	0.0	39.47

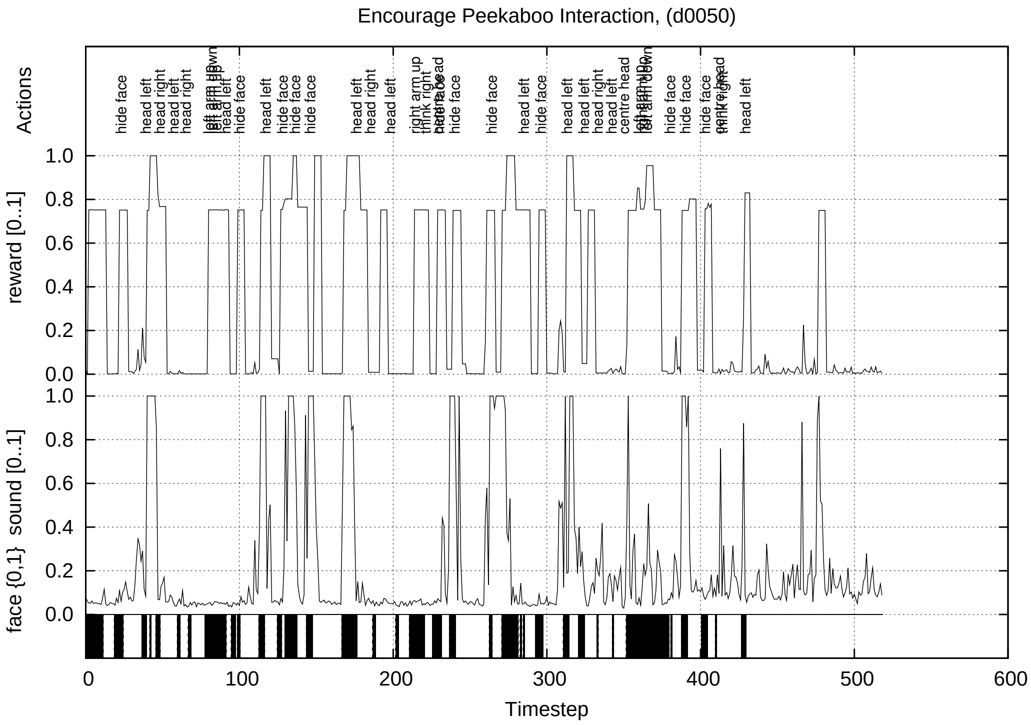


Figure C.16:

Table C.33: Actions executed (consolidated): Run d0051

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	32	6	3	7	4	27	10	12	5	2	2	6	1	6	123
chosen	53	3	0	0	17	5	3	2	0	0	0	0	0	13	96
both	85	9	3	7	21	32	13	14	5	2	2	6	1	19	219

Table C.34: Actions executed (primary): Run d0051

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	12.00	6.00	8.00	20.00	24.00	4.00	12.00	2.00	12.00	100.00
chosen :	7.89	0.00	44.74	7.89	5.26	0.00	0.00	0.00	34.21	100.00
both :	10.23	3.41	23.86	14.77	15.91	2.27	6.82	1.14	21.59	100.00
Percentage Random v Chosen Actions										
random	66.67	100.00	19.05	76.92	85.71	100.00	100.00	100.00	31.58	
chosen	33.33	0.00	80.95	23.08	14.29	0.00	0.00	0.00	68.42	
Overall Chosen %:	3.41	0.00	19.32	3.41	2.27	0.00	0.00	0.00	14.77	43.18

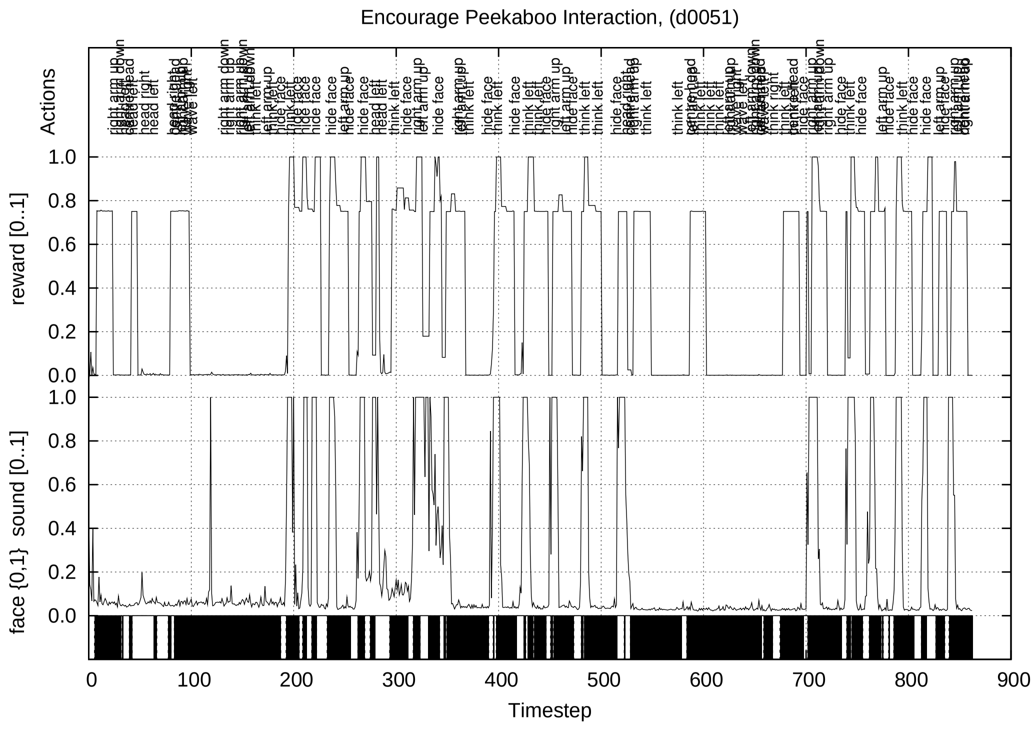


Figure C.17:

Table C.35: Actions executed (consolidated): Run d0052

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	22	23	8	13	2	24	7	3	4	0	6	1	7	5	125
chosen	74	31	9	16	6	11	3	0	0	0	2	0	6	2	160
both	96	54	17	29	8	35	10	3	4	0	8	1	13	7	285

Table C.36: Actions executed (primary): Run d0052

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	37.10	12.90	3.23	11.29	4.84	9.68	1.61	11.29	8.06	100.00
chosen :	52.54	15.25	10.17	5.08	0.00	3.39	0.00	10.17	3.39	100.00
both :	44.63	14.05	6.61	8.26	2.48	6.61	0.83	10.74	5.79	100.00
Percentage Random v Chosen Actions										
random	42.59	47.06	25.00	70.00	100.00	75.00	100.00	53.85	71.43	
chosen	57.41	52.94	75.00	30.00	0.00	25.00	0.00	46.15	28.57	
Overall Chosen %:	25.62	7.44	4.96	2.48	0.00	1.65	0.00	4.96	1.65	48.76

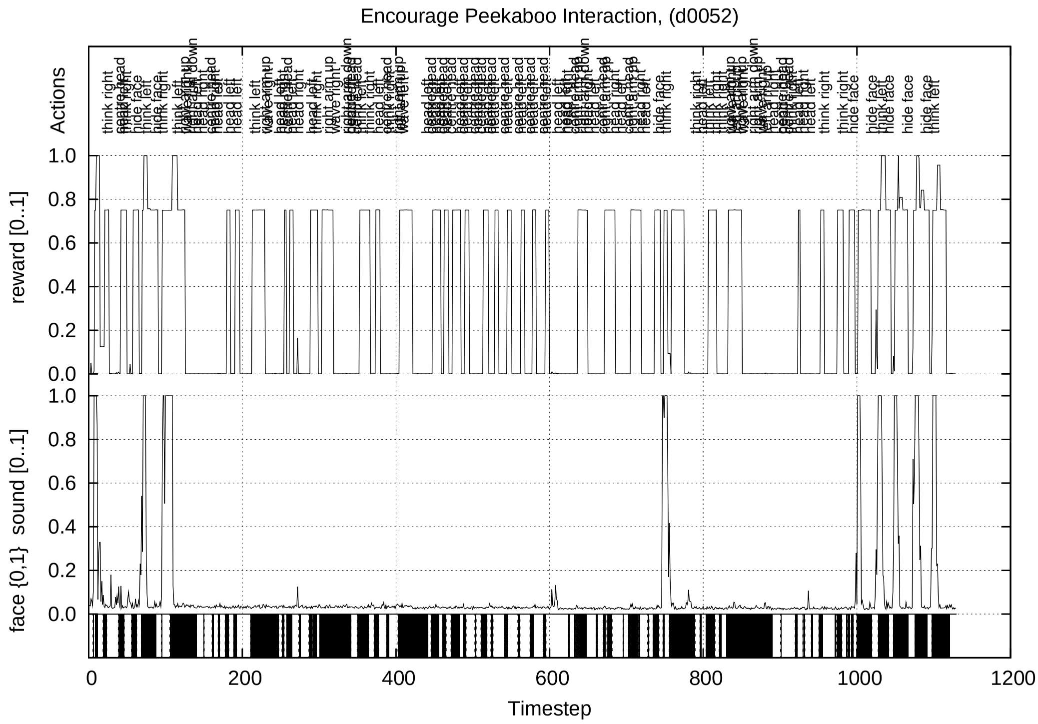


Figure C.18:

Table C.37: Actions executed (consolidated): Run d0053

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	31	3	5	6	6	21	5	7	5	4	5	2	3	3	106
chosen	26	1	1	21	11	4	7	0	1	0	0	0	3	1	76
both	57	4	6	27	17	25	12	7	6	4	5	2	6	4	182

Table C.38: Actions executed (primary): Run d0053

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	7.69	12.82	15.38	12.82	17.95	12.82	5.13	7.69	7.69	100.00
chosen :	4.17	4.17	45.83	29.17	0.00	0.00	0.00	12.50	4.17	100.00
both :	6.35	9.52	26.98	19.05	11.11	7.94	3.17	9.52	6.35	100.00
Percentage Random v Chosen Actions										
random	75.00	83.33	35.29	41.67	100.00	100.00	100.00	50.00	75.00	
chosen	25.00	16.67	64.71	58.33	0.00	0.00	0.00	50.00	25.00	
Overall Chosen %:	1.59	1.59	17.46	11.11	0.00	0.00	0.00	4.76	1.59	38.10

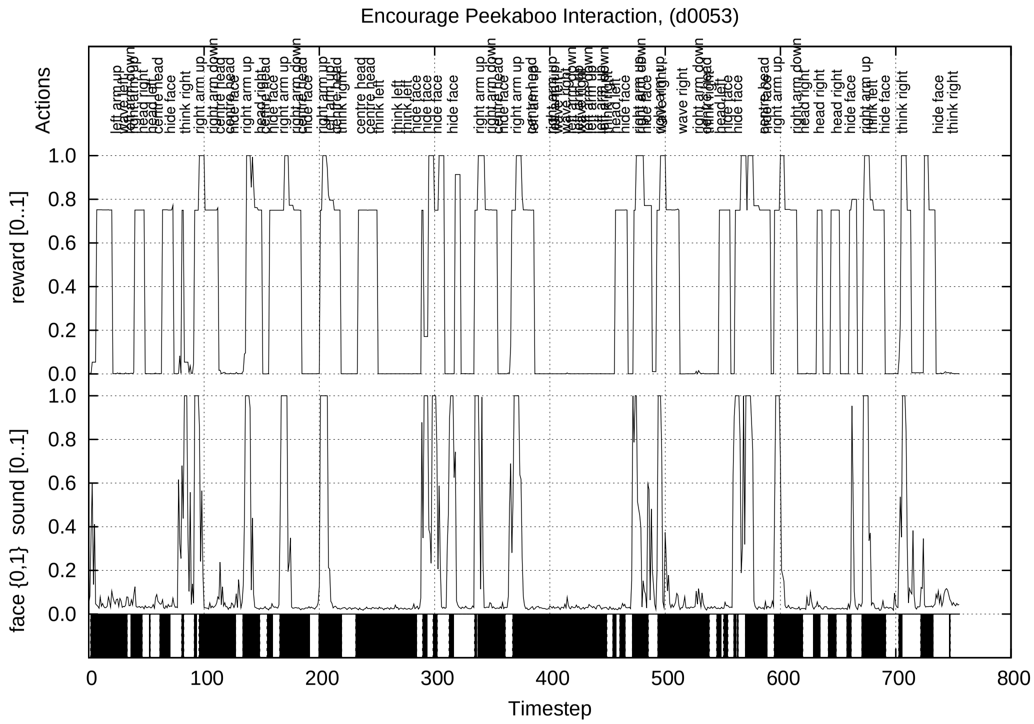


Figure C.19:

Table C.39: Actions executed (consolidated): Run d0054

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	47	3	6	8	3	57	3	2	2	2	1	0	3	1	138
chosen	19	3	1	49	42	8	0	0	0	0	0	0	0	0	122
both	66	6	7	57	45	65	3	2	2	2	1	0	3	1	260

Table C.40: Actions executed (primary): Run d0054

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	13.64	27.27	13.64	13.64	9.09	4.55	0.00	13.64	4.55	100.00
chosen :	6.52	2.17	91.30	0.00	0.00	0.00	0.00	0.00	0.00	100.00
both :	8.82	10.29	66.18	4.41	2.94	1.47	0.00	4.41	1.47	100.00
Percentage Random v Chosen Actions										
random	50.00	85.71	6.67	100.00	100.00	100.00	0.0	100.00	100.00	
chosen	50.00	14.29	93.33	0.00	0.00	0.00	0.0	0.00	0.00	
Overall Chosen %:	4.41	1.47	61.76	0.00	0.00	0.00	0.0	0.00	0.00	67.65

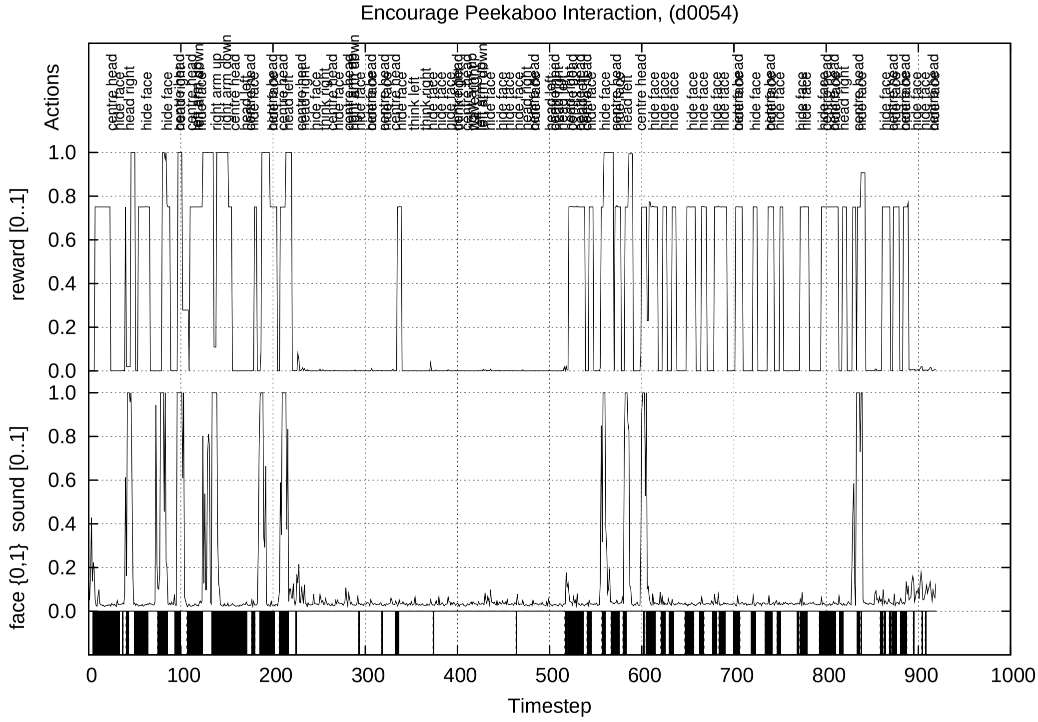


Figure C.20:

Table C.41: Actions executed (consolidated): Run d0055

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	23	7	5	10	4	19	3	7	2	3	0	2	4	0	89
chosen	20	5	0	1	0	39	0	0	0	0	0	0	13	0	78
both	43	12	5	11	4	58	3	7	2	3	0	2	17	0	167

Table C.42: Actions executed (primary): Run d0055

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	21.88	15.62	12.50	9.38	21.88	0.00	6.25	12.50	0.00	100.00
chosen :	27.78	0.00	0.00	0.00	0.00	0.00	0.00	72.22	0.00	100.00
both :	24.00	10.00	8.00	6.00	14.00	0.00	4.00	34.00	0.00	100.00
Percentage Random v Chosen Actions										
random	58.33	100.00	100.00	100.00	100.00	0.0	100.00	23.53	0.0	
chosen	41.67	0.00	0.00	0.00	0.00	0.0	0.00	76.47	0.0	
Overall Chosen %:	10.00	0.00	0.00	0.00	0.00	0.0	0.00	26.00	0.0	36.00

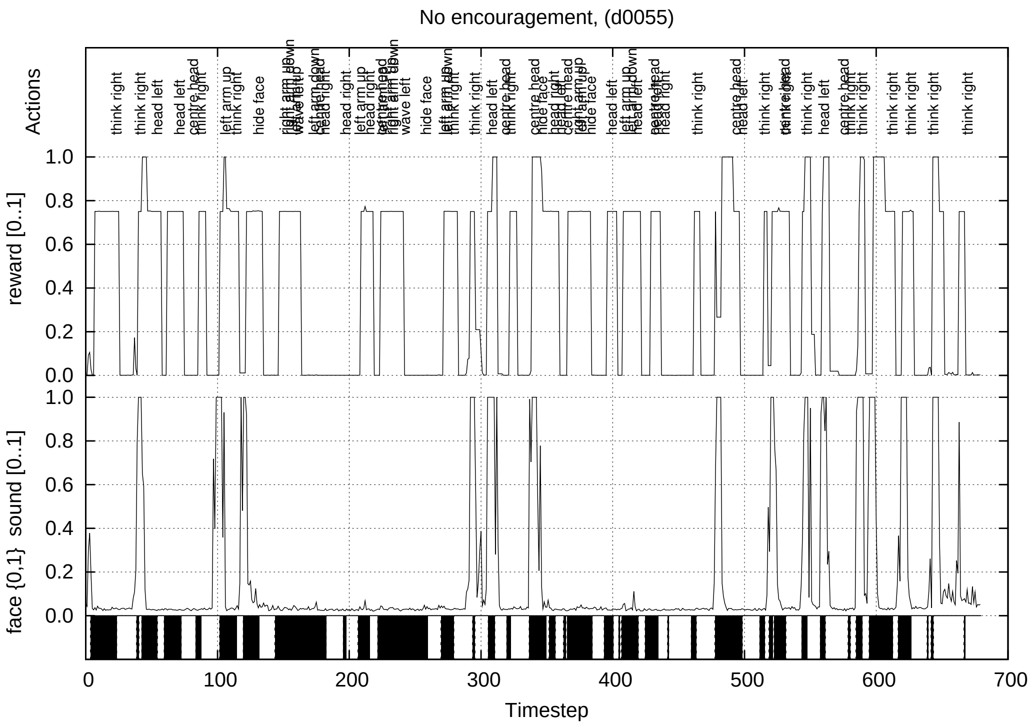


Figure C.21:

Table C.43: Actions executed (consolidated): Run d0056

	0 Rst	3 HL	4 HR	5 HF	6 Hid	7 NA	8 RAU	9 LAU	10 RAD	11 LAD	12 RAW	13 LAW	14 TR	15 TL	Tot
random	21	10	6	9	7	22	11	8	6	2	5	0	0	2	109
chosen	18	10	8	0	2	6	2	0	1	0	8	0	0	0	55
both	39	20	14	9	9	28	13	8	7	2	13	0	0	2	164

Table C.44: Actions executed (primary): Run d0056

	HL	HR	Hid	RAU	LAU	RAW	LAW	TR	TL	total
Frequency As Percentage of Primary Actions										
random :	20.41	12.24	14.29	22.45	16.33	10.20	0.00	0.00	4.08	100.00
chosen :	33.33	26.67	6.67	6.67	0.00	26.67	0.00	0.00	0.00	100.00
both :	25.32	17.72	11.39	16.46	10.13	16.46	0.00	0.00	2.53	100.00
Percentage Random v Chosen Actions										
random	50.00	42.86	77.78	84.62	100.00	38.46	0.0	0.0	100.00	
chosen	50.00	57.14	22.22	15.38	0.00	61.54	0.0	0.0	0.00	
Overall Chosen %:	12.66	10.13	2.53	2.53	0.00	10.13	0.0	0.0	0.00	37.97

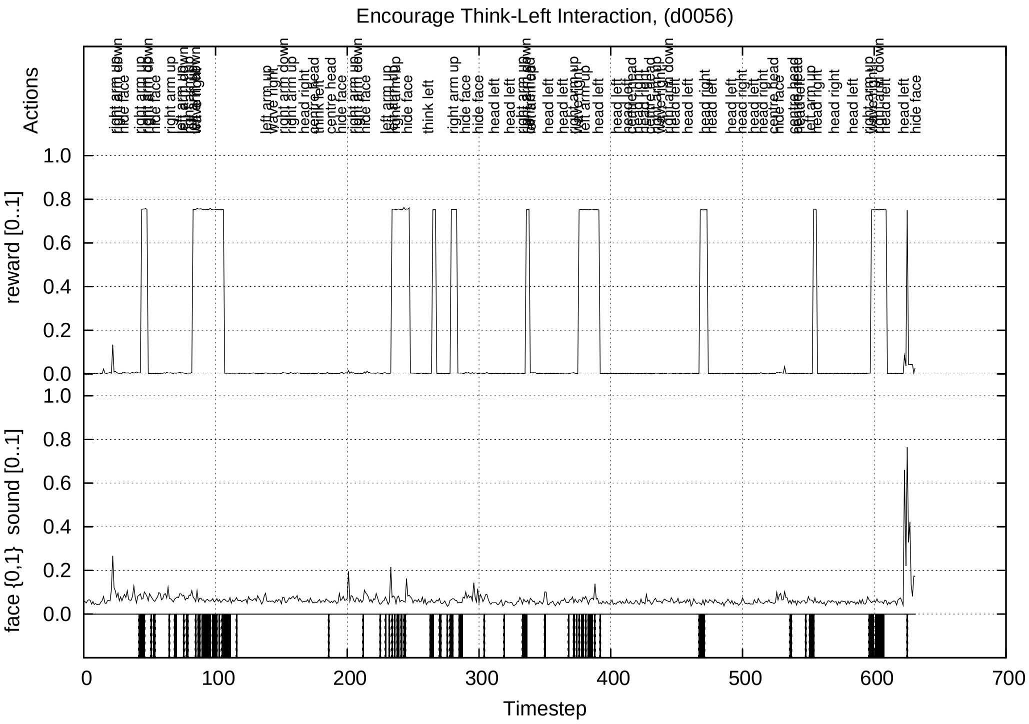


Figure C.22:

Appendix D

Source Code CD

Source Code CD

AID

Java code.

Requires Tekkotsu

Online visualization of the AID vs AID plot.

Robot controlled is an Aibo ERS220.

IHA

C++ code. Using YARP framework.

Full implementation of Interaction History Architecture for robot ontogeny.

Robots that can be controlled are: Aibo (URBI), Simulated Pioneer (Player/Stage 2.0) and KASPAR (SSC32 serial control).

This code is Open Source and is also available as part of the iCub software issued by the RobotCub project (http://www.robotcub.org/). See http://eris.liralab.it/wiki/Main_Page

Appendix E

Kaspar2 Peekaboo Action States

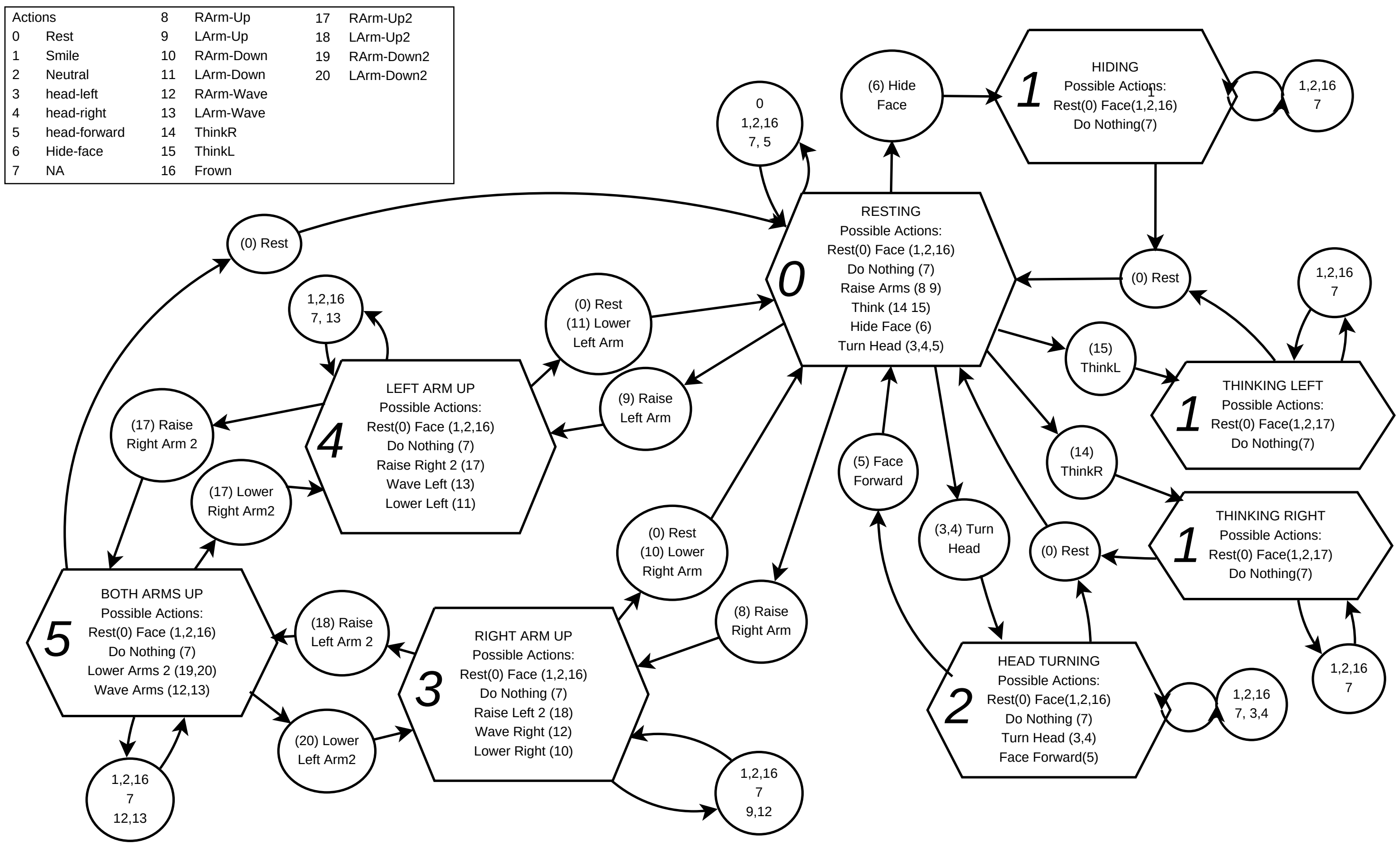


Figure E.1: Kaspar2 Action States

Appendix F

Interaction History Architecture Software Modules

F.1 Modules

F.1.1 Data Store

Description: The Data Store collects sensor data and creates experiences, placing them in a metric space and associating quality values and action values to create the interaction history space.

Executable: data_store.cpp

Files: **data_store.cpp** Main executable

experience/DataFrame.h Class to store a single data frame

experience/Experience.h Class to store a single experience

experience/ExperienceProcessor.h Processing functionality for experiences *e.g.* merge/delete them.

experience/ExperienceProcessor.cpp

experience/DistanceSpaceClass.h Class to hold the Distance Space and processing functionality at the Distance Space level.

experience/DistanceSpaceClass.cpp

experience/BinWindowMaxEntropy.h Adaptive binning using entropy maximization

experience/WindowIDCalc.h Information Distance calculation (Moving Window)

experience/ExperienceProcessorFileRW.cpp Read-Write Experiences

experience/serialization.h Serialization code

Execution Parameters: –file <file> configuration file

–dbg <N> Debug level (0-60)

–save <file> Save Experience Space when finished

–load <file> Load Experience Space from <file> before start

–connect_to_sensors <port> Connect to the specified sensor port on startup

Configuration Options: To be set in configuration file

name *String* Base name for ports (default iha)

dsnumber *Int* Data Store Number for multiple data stores (default 1)

HORIZONS *Int Int ...* List of horizons to keep metric spaces for

num_bins *Int* Number of bins in quantization. (default 5)

granularity *Int* Rate at which experiences are created in timesteps (default 1)

experience_action_gap *Int* For tuning correct association of action with experience (default 1, *i.e.* next action)

regular_experiences *String* Experiences created depending on timestep (default TRUE)

action_experiences *String* Experiences created when action changes (default FALSE)

value_experiences *String* Experiences created when reward value changes (default FALSE)

num_actions *Int* Number of actions configured (default 4)

write_curr_dist_to_port *String* Current Distance list written to a port (default TRUE)

write_max_dsp_neighbours *Int* Number of neighbours to output (default 0)

write_max_dsp_radius *Double* Max radius of experiences in neighbour list (default 0)

neighbour_radius *Double* Neighbourhood size (default 1.0)

merge_adapt_type *String* Merge Adaptation Type NONE, CYCLE_TIME, NUM_COMPARISONS (default NONE)

merge_threshold *Double* Adaptive Merge Threshold (default 0.0)

merge_increment *Double* Adaptive Merge Increment (default 0.01)

only_merge_same_actions *String* (default FALSE)

merge_exp_threshold *Int* For NUM_COMPARISONS Merge Adaptation (default 400)

merge_cycle_time_threshold *Int* For CYCLE_TIME Merge Adaptation (default 400)

purge_experiences *String* Purge Experiences switch (default FALSE)

purge_threshold *Double* To purge only experience with quality less than this(default 0.0)

adaptive_binning *String* Adaptive Binning using Entropy Maximization (default FALSE)

adaptive_binning_window_size *Int* Adaptive Binning - window over which entropy is maximized (default 32)

histogram_resolution *Int* Adaptive Binning (default 256)

future_horizon *Int* Horizon over which quality is updated (default 200)

future_value_update_type *String* Can be MAX, or BIASED (default MAX)

metric_space_heuristic *String* Can be NONE, TREE or NEIGHBOUR (default NONE)

verify_heuristic *String* For testing (default FALSE)

heuristic_start_threshold *Int* For Neighbour Heuristic algorithm (default 40)

heuristic_tree_radius *Double* For Tree heuristic algorithm (default 1.0)

num_image_sensors_x *Int* Number of image sensors to make from image - X direction (default 8)

num_image_sensors_y *Int* Number of image sensors to make from image - Y direction (default 8)

use_reward_action_in_exp *String* Whether experience includes the reward and action as sensors (default TRUE)

Ports Created: **/<name>/ds<dnumber>/data:in** Input port

/<name>/ds/currdist:out:<horizon> Output port for current experience neighbours

F.1.2 Kaspar2 Control

Description: Control for the Kaspar2 Robot and Sensor Collector. As well as providing the sendAction() function for the Kaspar2 robot, this module reads all necessary sensor data including image data, motivation feedback data, sound data and face detection data then consolidates them and writes them to a port.

Executable: kaspar/kaspar_control

Files: **kaspar/kaspar_control.cpp** Main executable

kaspar/KasparActions.cpp Class for holding kaspar action specifications

kaspar/KasparActions.h

kaspar/KasparSequence.h Class for holding kaspar motor control sequences for actions

Execution Parameters: **–file <file>** configuration file

–hwconfig <file> hardware configuration file

–dbg <N> Debug level (0-60)

–connect_to_image <port> Connect to the specified image port on startup

–connect_to_coords <port> Connect to the specified port for detected face coordinates on startup

–connect_to_reward <port> Connect to the specified port for reward data on startup

–connect_to_soundsensor <port> Connect to the specified sound sensor port on startup

Configuration Options: To be set in configuration file

name *String* Base name for ports (default iha)

action_defs_file *String* File in which action definitions are configured (default action_defs.txt)

num_image_sensors_x *Int* Number of image sensors to make from image - X direction (default 8)

num_image_sensors_y *Int* Number of image sensors to make from image - Y direction (default 8)

sensordatarate *Int* Sensor data rate for output in ms (default 100)

reward_display *String* Display reward by using expressive actions (default TRUE)

action_ehi *Int* Action (expression) to execute for High reward (default 1)

action_elo *Int* Action (expression) to execute for Low reward (default 16)

action_emid *Int* Action (expression) to execute for Mid reward (default 2)

th_ehi *Int* High Threshold for expression change (default 0.8)

th_elo *Int* Low Threshold for expression change (default 0.3)

Ports Created: **/<name>/ac/action:out** Action Advice output port

/<name>/sensor:out Sensor output port

/<name>/action:cmd Action Reader input port

/<name>/image:in Image input port

/<name>/coords:in Face Coordinates input port

/<name>/reward:in Reward input port

/<name>/soundsensor:in Sound Sensor input port

F.1.3 Kaspar Action Selection Process

Description: Wrapper for the action selection process

Executable: kaspar_action_selection

Files: **kaspar_action_selection.cpp** Main executable

include/iCub/iha/action_selection_main_loop.h Generic action selection loop. This is the main process that takes in the nearest neighbour list and uses the roulette wheel action selection process to generate action advice.

kaspar/KasparActions.cpp Class for holding kaspar action specifications

kaspar/KasparActions.h

kaspar/KasparSequence.h Class for holding kaspar motor control sequences for actions

Execution Parameters: **–file <file>** configuration file

–dbg <N> Debug level (0-60)

–connect_to_action <port> Connect to the specified action port on startup

–connect_to_dist <port> Connect to the specified nearest neighbour distance port on startup

Configuration Options: To be set in configuration file

name *String* Base name for ports (default iha)

action_defs_file *String* File in which action definitions are configured (default action_defs.txt)

neighbour_radius *Double* Max radius of neighbourhood. (default 1.0)

temperature *Double* Starting temperature (affecting chance of random) (default 4.0)

temp_dec *Double* Decrement of temperature per action step (default 0.002)

Ports Created: /<name>/ac/action:out Action Advice output port

/<name>/ac/currdist:in:<horizon> Input port for current experience neighbours

F.1.4 Send Action Utility

Description: Utility to send an action to an active control process

Executable: send_action

Files: <control>/send_action.cpp Main executable

Execution Parameters: –file <file> configuration file

–dbg <N> Debug level (0-60)

–connect_to_action <port> Connect to the specified action port on startup

Configuration Options: To be set in configuration file

name *String* Base name for ports (default iha)

Ports Created: /<name>/ac/singleaction:out Action Advice output port

F.1.5 Motivation Dynamics

Description: Collects the sound sensor and face detection data and writes a resultant reward to a port

Executable: motivation_dynamics

Files: motivation_dynamics/motivation_dynamics.cpp Main executable

Execution Parameters: –file <file> configuration file

–dbg <N> Debug level (0-60)

–connect_to_coords <port> Connect to the specified port for detected face coordinates on startup

–connect_to_soundsensor <port> Connect to the specified sound sensor port on startup

Configuration Options: To be set in configuration file

name *String* Base name for ports (default iha)

Ports Created: **/<name>/reward:out** Reward output port

/<name>/coords:in Face Coordinates input port

/<name>/soundsensor:in Sound Sensor input port

F.1.6 Sound Sensor

Description: Creates a single valued sensor from a YARP sound stream

Executable: sound_sensor

Files: **sound/sound_sensor.cpp** Main executable

Execution Parameters: **–file <file>** configuration file

–dbg <N> Debug level (0-60)

–connect_to_soundsensor <port> Connect to the specified sound sensor port on startup

Configuration Options: To be set in configuration file

name *String* Base name for ports (default iha)

soundsensorrate *Int* Rate at which the sound sensor data is produced on the output port in ms. (default 100)

soundgain *Double* To compensate for low volume sound source. (default 2.5)

Ports Created: **/<name>/soundsensor:out** Sound Sensor output port

/<name>/sound:in Sound Stream input port

F.1.7 Face Detector - IHA modifications to opencv_facedetect

Description: Detects faces in YARP images on a port using multiple HAAR cascades. Chooses largest face if more than one is detected and outputs the coordinates on a YARP port.

Executable: facedetect

Files: **iha_facedetect/face_detect.cpp** Main executable

Execution Parameters: **–file <file>** configuration file

–dbg <N> Debug level (0-60)

Configuration Options: To be set in configuration file

PORTS *Group* Group level; List of ports. Requires definitions for input, output and coords ports.

CASCADES *Group* Group level; List of cascades.

Ports Created: Specified in config file. Opens an Input port for images, an Output port for images and an output port for Coordinates of detected faces.

F.1.8 iCub Control

Description: Control for the icub robot (ODE simulator currently) and Sensor Collector. As well as providing the sendAction() function for the iCub robot, this module reads all necessary sensor data including image data, motivation feedback data, sound data and face detection data then consolidates them and writes them to a port.

Executable: iCub/icub_control

Files: **iCub/icub_control.cpp** Main executable

iCub/ICubActions.cpp Class for holding iCub action specifications

icub/ICubActions.h

icub/ICubSequence.h Class for holding iCub motor control sequences for actions

Execution Parameters: **–file <file>** configuration file

–**hwconfig <file>** hardware configuration file

–**dbg <N>** Debug level (0-60)

–**connect_to_image <port>** Connect to the specified image port on startup

–**connect_to_coords <port>** Connect to the specified port for detected face coordinates on startup

–**connect_to_reward <port>** Connect to the specified port for reward data on startup

–**connect_to_soundsensor <port>** Connect to the specified sound sensor port on startup

Configuration Options: To be set in configuration file

name *String* Base name for ports (default iha)

action_defs_file *String* File in which action definitions are configured (default action_defs.txt)

num_image_sensors_x *Int* Number of image sensors to make from image - X direction (default 8)

num_image_sensors_y *Int* Number of image sensors to make from image - Y direction (default 8)

sensordatarate *Int* Sensor data rate for output in ms (default 100)

Ports Created: **/<name>/ac/action:out** Action Advice output port

/<name>/sensor:out Sensor output port

/<name>/action:cmd Action Reader input port

/<name>/image:in Image input port

/<name>/coords:in Face Coordinates input port

/<name>/reward:in Reward input port

/<name>/soundsensor:in Sound Sensor input port

F.1.9 iCub Action Selection Process

Description: Wrapper for the action selection process

Executable: icub_action_selection

Files: **icub_action_selection.cpp** Main executable

include/iCub/iha/action_selection_main_loop.h Generic action selection loop. This is the main process that takes in the nearest neighbour list and uses the roulette wheel action selection process to generate action advice.

icub/ICubActions.cpp Class for holding iCub action specifications

icub/ICubActions.h

icub/ICubSequence.h Class for holding iCub motor control sequences for actions

Execution Parameters: **–file <file>** configuration file

–dbg <N> Debug level (0-60)

–connect_to_action <port> Connect to the specified action port on startup

–connect_to_dist <port> Connect to the specified nearest neighbour distance port on startup

Configuration Options: To be set in configuration file

name *String* Base name for ports (default iha)

action_defs_file *String* File in which action definitions are configured (default action_defs.txt)

neighbour_radius *Double* Max radius of neighbourhood. (default 1.0)

temperature *Double* Starting temperature (affecting chance of random) (default 4.0)

temp_dec *Double* Decrement of temperature per action step (default 0.002)

Ports Created: **/<name>/ac/action:out** Action Advice output port

/<name>/ac/currdist:in:<horizon> Input port for current experience neighbours

Appendix G

IHA Process Diagram

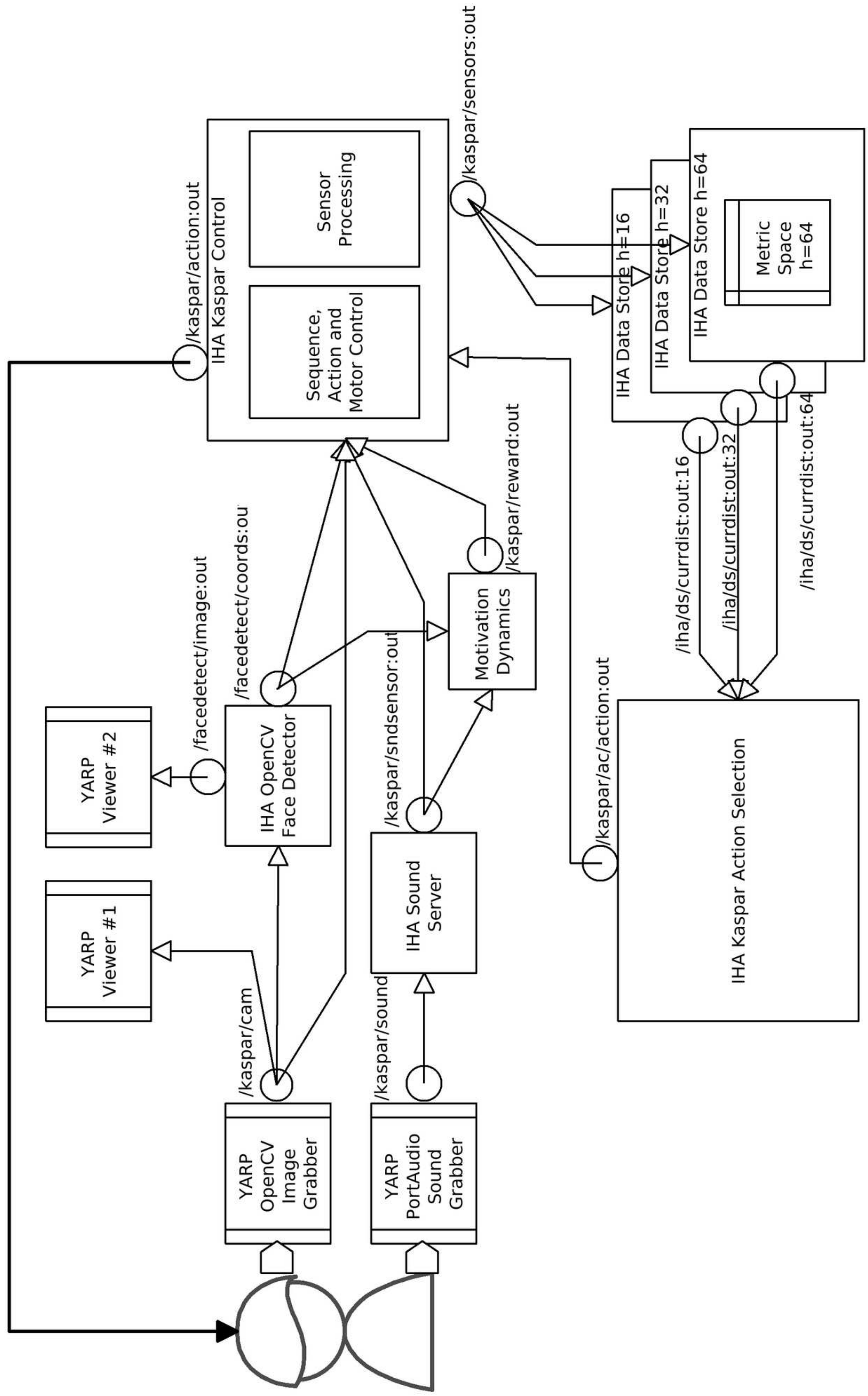


Figure G.1: IHA Process Diagram showing main processes and connections.

Bibliography

Akhras, F. and Self, J. (2000), Computers as cognitive tools: No more walls, *in* S. P. Lajoie, ed., 'Modeling the Process, Not the Product, of Learning', Vol. 2, Lawrence Erlbaum Associates.

Anderson, J. R. (1996), 'Act: A simple theory of complex cognition', *American Psychologist* **51**, 355–365.

Anderson, M. L. (2003), 'Embodied cognition: A field guide', *Artificial Intelligence* **149**, 91–130.

Arai, S., Sycara, K. and Payne, T. R. (2000), Experience-based reinforcement learning to acquire effective behavior in a multi-agent domain, *in* 'Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence', pp. 125–135.

Avila-García, O. and Cañamero, L. (2005), Hormonal modulation of perception in motivation-based action selection architectures, *in* L. Cañamero and D. Evans, eds, 'Agents that Want and Like: Motivational and Emotional Roots of Cognition and Action, Symposium of the AISB'05 Convention', University of Hertfordshire, UK.

Baddeley, A., Conway, M. and Aggleton, J., eds (2002), *Episodic Memory: New Directions in Research*, Oxford University Press.

Baillie, J.-C. (2005), URBI: Towards a universal robotic low-level programming language, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2005'. http://www.urbiforge.com.

Baker, G. L. and Gollub, J. P. (1990), *Chaotic Dynamics and Introduction*, Cambridge University Press.

Bakker, B. (2002), Reinforcement learning with long short-term memory, *in* T. G. Dietterich, S. Becker and Z. Ghahramani, eds, 'Advances in Neural Information Processing Systems 14', MIT Press, Cambridge, MA, pp. 1475–1482.

Balkenius, C. and Morén, J. (1998), Computational models of classical conditioning: a comparative study, *in* 'Proceedings of the fifth international conference on simulation of adaptive behavior on From animals to animats 5', pp. 348–353.

Bartlett, F. C. (1932), *Remembering*, Cambridge University Press.

Barto, A. G. and Şimşek, Ö. (2005), Intrinsic motivation for reinforcement learning systems, *in* 'Proceedings of the Thirteenth Yale Workshop on Adaptive and Learning Systems', pp. 113–118.

Billard, A. and Dautenhahn, K. (1998), 'Grounding communication in autonomous robots: an experimental study', *Robotics and Autonomous Systems, special issue on "Scientific Methods in Mobile Robotics"* **24**(1-2), 71–81.

Billard, A. Dautenhahn, K. and Hayes, G. (1998), Experiments on human-robot communication with robota, an imitative learning and communication doll robot, Technical Report CPM-98-38, Centre for Policy Modelling, Manchester Metropolitan University. Contribution to Workshop "Socially Situated Intelligence" at SAB98 conference, Zuerich.

Billard, A., Robins, B., Dautenhahn, K. and Nadel, J. (2006), 'Building robota, a mini-humanoid robot for the rehabilitation of children with autism', *RESNA Assistive Technology Journal* **19**, 37–49.

Blank, D., Kumar, D., Meeden, L. and Marshall, J. (2005), 'Bringing up robot: Fundamental mechanisms for creating a self-motivated, self-organizing architecture', *Cybernetics and Systems* **36**(2).

Bonarini, A., Lazaric, A., Restelli, M. and Vitali, P. (2006), Self-development framework for reinforcement learning agents, *in* 'Proceedings the of 5th International Conference on Development and Learning (ICDL 2006)'.

Bond, A. (2002), Modeling social relationship: An agent architecture for voluntary mutual control, *in* K. Dautenhahn, A. H. Bond, L. Cañamero and B. Edmonds, eds, 'Socially Intelligent Agents: Creating Relationships with Computers and Robots', Kluwer Academic Publishers, pp. 29–36.

Bovet, S. and Pfeifer, R. (2005), Emergence of delayed reward learning from sensorimotor coordination, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 841–846.

Braitenberg, V. (1984), *Vehicles: Experiments in Synthetic Psychology*, A Bradford Book MIT Press.

Breazeal, C. and Scassellati, B. (2000), 'Infant-like social interactions between a robot and a human caregiver', *Adaptive Behavior* **8**(1), 49–74.

Bruner, J. S. and Sherwood, V. (1975), Peekaboo and the learning of rule structures, *in* J. Bruner, A. Jolly and K. Sylva, eds, 'Play: Its Role in Development and Evolution', New York: Penguin, pp. 277–285.

Byrne, R. W. and Whiten, A. (1988), *Machiavellian Intelligence*, Clarendon Press.

Clancey, W. J. (1997), *Situated Cognition: On human knowledge and computer representations*, Learning in doing: Cognitive and compuational perspectives, Cambridge University Press.

Conway, M. (1995), *Flashbulb Memories*, Essays in Cognitive Psychology, Lawrence Erlbaum Associates.

Coolican, H. (1994), *Research methods and statistics in psychology*, 2 edn, Hodder & Stoughton.

Crutchfield, J. (1990), Information and its metric, *in* L. Lam and H. Morris, eds, 'Nonlinear Structures in Physical Systems - Pattern Formation, Chaos and Waves', Springer-Verlag, New York, pp. 119–130.

Dautenhahn, K. (1994), Trying to imitate- a step towards releasing robots from social isolation, *in* P. Gaussier and J.-D. Nicoud, eds, 'Proceedings From Perception to Action Conference', IEEE Computer Society Press, Lausanne, Switzerland, pp. 290–301.

Dautenhahn, K. (1996), Embodied cognition in animals and artifacts, *in* 'Proc. AAAI FS Embodied Cognition and Action', AAAI Press, pp. 27–32. Technical report FS-96-02.

Dautenhahn, K. (1999), Embodiment and interaction in socially intelligent life-like agents, *in* C. L. Nehaniv, ed., 'Computation for Metaphors, Analogy and Agent', Vol. 1562 of *Springer Lecture Notes in Artificial Intelligence*, Springer, pp. 102–142.

Dautenhahn, K., Bond, A. H., Cañamero, L. and Edmonds, B., eds (2002), *Socially Intelligent Agents: Creating Relationships with Computers and Robots*, Multiagent Systems, Artificial Societies, and Simulated Organizations, Kluwer Academic Publishers.

Dautenhahn, K. and Christaller, T. (1996), Remembering, rehearsal and empathy - towards a social and embodied cognitive psychology for artifacts, *in* Seán Ó Nualláin, Paul McKevitt and Eoghan Mac Aogáin, eds, 'Two Sciences of the Mind: Readings in cognitive science and consciousness', John Benjamins North America Inc., pp. 257–282.

Dautenhahn, K. and Werry, I. (2001), 'The aurora project: Using mobile robots in autism therapy', *Learning Technology, online newsletter, publication of IEEE Computer Society Learning Technology Task Force (LTTF)* **3**(1). http://lttf.ieee.org/learn_tech/issues/january2001/index.html#12.

Dillenbourg, P. (1999), What do yuo mean by collaborative leraning?., *in* P. Dillenbourg, ed., 'Collaborative-learning: Cognitive and Computational Approaches.', Oxford: Elsevier.

du Boulay, B. (2000), Computers as cognitive tools: No more walls, *in* S. P. Lajoie, ed., 'Fallible, Distrctable, Forfetful, Willful, and Irrational Learners', Vol. 2, Lawrence Erlbaum Associates.

Edelman, G. (1992), *Brilliant air, brilliant fire. On the matter of mind.*, Basic Books, New York.

Edmonds, B. (2002), Developing agents who can relate to us: Putting agents in our loop via situated self-creation, *in* K. Dautenhahn, A. H. Bond, L. Cañamero and B. Edmonds, eds, 'Socially Intelligent Agents: Creating Relationships with Computers and Robots', Kluwer Academic Publishers, pp. 37–44.

Fong, T., Nourbakhsh, I. and Dautenhahn, K. (2003), 'A survey of socially interactive robots', *Robotics and Autonomous Systems* **42**(3-4), 143–166.

Glenberg, A. M. (1997), 'What is memory for?', *Behavioral and Brain Sciences* **20**(1), 1–55.

Gower, J. (1966), 'Some distance properties of latent root and vector methods used in multivariate analysis', *Biometrika* **53**(3/4), 325–338.

Gurney, K. (1997), *An Introduction to Neural Networks*, Taylor & Francis, CRC Press.

Harnad, S. (1990), 'The symbol grounding problem', *Physica D* **42**(4), 335–346.

Ho, W. C., Dautenhahn, K. and Nehaniv, C. L. (2008), 'Computational memory architectures for autobiographic agents interacting in a complex virtual environment: a working model', *Connection Science* **20**(1), 21–65.

Hoernagl, E., Poelz, P. and Prem, E. (2004), 'Anchoring symbols by mapping sequences of distance measurements: Experimental results', *Anchoring Symbols* *to Sensor Data, Papers from the 2004 AAAI Workshop, Technical Report WS-04-03* pp. 33–39.

Holland, O. (2003), 'Exploration and high adventure: the legacy of grey walter', *Phil. Trans. R. Soc. Lond.* **361**, 2085–2121.

Jaeger, H. and Haas, H. (2004), 'Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communications', *Science* **304**(5667), 78–80.

Jain, A. K., Duin, R. P. and Mao, J. (2000), 'Statistical pattern recognition: A review', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(1), 4–37.

Jolliffe, I. T. (2002), *Principal Component Analysis*, 2 edn, Springer, Berlin.

Jolly, A. (1966), 'Lemur social behaviour and primate intelligence', *Science* **153**, 501–506.

Kaplan, F. and Hafner, V. (2005), Mapping the space of skills: An approach for comparing embodied sensorimotor organizations, *in* 'Proc. 4th IEEE International Conference on Development and Learning (ICDL-05)', IEEE, pp. 129–134.

Kelso, J. S. (1995), *Dynamic patterns, the self-organization of brain and behavior*, Cambridge, MA: MIT Press.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983), 'Optimization by simulated annealing', *Science* **220**(4598), 671–680.

Kohonen, T. (1984), *Self-organization and associative memory*, Springer.

Kolodner, J. (1993), *Case-based Reasoning*, Morgan Kaufman.

Kose-Bagci, H., Dautenhahn, K., Syrdal, D. S. and Nehaniv, C. L. (2007), Drum-mate: A human-humanoid drumming experience, *in* 'Proc. IEEE-RAS International Conference on Humanoid Robots (Humanoids2007)'.

Kozima, H. (2002), Infanoid: A babybot that explores the social environment, *in* K. Dautenhahn, A. H. Bond, L. Cañamero and B. Edmonds, eds, 'Socially Intelligent Agents: Creating Relationships with Computers and Robots', Kluwer Academic Publishers, pp. 157–164.

Kozima, H., Nakagawa, C. and Yano, H. (2005), Using robots for the study of human social development, *in* 'AAAI Spring Symposium on Developmental Robotics', Palo Alto, CA, USA, pp. 111–114.

Lakoff, G. and Johnson, M. (1999), *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*, Basic Books, New York.

Lin, L.-J. and Mitchell, T. (1992), Reinforcement learning with hidden states, *in* J.-A. Meyer, H. L. Roitblat and S. W. Wilson, eds, 'From animals to animats 2: Proceedings of the second international conference on simulation of adaptive behavior', MIT Press, Cambridge, MA, pp. 271–280.

Linåker, F. and Jacobsson, H. (2001), 'Learning delayed response tasks through unsupervised event extraction', *International Journal of Computational Intelligence and Applications* **1**(4), 413–426.

Lindblom, J. and Ziemke, T. (2003), 'Social situatedness of natural and artificial intelligence: Vygotsky and beyond', *Adaptive Behavior* **11**(2), 79–96.

Lungarella, M., Metta, G., Pfeifer, R. and Sandini, G. (2003), 'Developmental robotics: A survey', *Connection Science* **15**(4), 151–190.

Lungarella, M., Pegors, T., Bulwinkle, D. and Sporns, O. (2005), 'Methods for quantifying the informational structure of sensory and motor data', *Neuroinformatics* **3**(3), 243–262.

Lungarella, M. and Pfeifer, R. (2001), Robots as cognitive tools information theoretic analysis of sensory-motor data, *in* 'Proc. of the 1st IEEE-RAS Int. Conf. on Humanoid Robots', pp. 245–252.

Lungarella, M. and Sporns, O. (2005), Information self-structuring: Key principles for learning and development, *in* 'Proc. of 4th International Conference on Development and Learning', pp. 25–30.

MacKay, D. J. C. (2003), *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press.

Marshall, J. Blank, D. and Meeden, L. (2004), An emergent framework for self-motivation in developmental robotics, *in* '3rd International Conference on Development and Learning'.

Matarić, M. J. (1992), 'Integration of representation into goal-driven behaviour-based robots', *IEEE Transactions on Robotics and Automation* **8**(3), 304–312.

Maturana, H. R. and Varela, F. J. (1987), *The Tree of Knowledge: The Biological Roots of Human Understanding*, New Science Library (Shambhala), Boston.

McCallum, R. (1996), 'Hidden state and reinforcement learning with instance-based state identification', *Systems, Man, and Cybernetics, Part B, IEEE Transactions on* **26**(3), 464–473.

Meltzoff, A. and Moore, M. (1997), 'Explaining facial imitation: a theoretical model', *Early Development and Parenting* **6**, 179–192.

Metta, G., Fitzpatrick, P. and Natale, L. (2006), 'YARP: Yet Another Robot Platform', *International Journal of Advanced Robotic Systems* **3**(1), 43–48.

Michaud, F. and Matarić, M. J. (1998), 'Learning from history for behavior-based mobile robots in non-stationary conditions', *Machine Learning* **31**(1-3), 141–167.

Miikkulainen, R. (1992), 'Trace feature map: A model of episodic associative memory', *Biological Cybernetics* **66**(3), 273–282.

Miller, L. K. (1999), 'The savant syndrome: Intellectual impairment and exceptional skill', *Psychological Bulletin* **125**(1), 31–46.

Mirza, N. A., Nehaniv, C. L., Dautenhahn, K. and te Boekhorst, R. (2005*a*), Using sensory-motor phase-plots to characterise robot-environment interactions, *in* 'Proc. of 6th IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA2005)', pp. 581–586.

Mirza, N. A., Nehaniv, C. L., Dautenhahn, K. and te Boekhorst, R. (2005*b*), Using temporal information distance to locate sensorimotor experience in a metric space, *in* 'Proc. 2005 IEEE Congress on Evolutionary Computation', Vol. 1, IEEE Press, Edinburgh, Scotland, pp. 150–157.

Mirza, N. A., Nehaniv, C. L., te Boekhorst, R. and Dautenhahn, K. (2005), Robot self-characterisation of experience using trajectories in sensory-motor phase space, *in* L. Berthouze, F. Kaplan, H. Kozima, H. Yano, J. Konczak, G. Metta, J. Nadel, G. Sandini, G. Stojanov and C. Balkenius, eds, 'Proc. 5th International Workshop on Epigenetic Robotics', Lund University Cognitive Studies.

Mitchell, T. (1997), *Machine Learning*, McGraw-Hill.

Montague, D. P. F. and Walker-Andrews, A. S. (2001), 'Peekaboo: A new look at infant's perception of emotion expression', *Developmental Psychology* **37**(6), 826–838.

Nehaniv, C. L. (1999*a*), Narrative for artifacts: Transcending context and self, *in* P. Sengers and M. Mateas, eds, 'Narrative Intelligence: Papers from the 1999 AAAI Fall Symposium', American Association for Artificial Intelligence, North Falmouth, Massachusetts, pp. 101–104.

Nehaniv, C. L. (1999*b*), Story-telling and emotion: Cognitive technology considerations in networking temporally and affectively grounded minds, *in* 'Third International Conference on Cognitive Technology: Networked Minds (CT'99)', San Francisco/Silicon Valley, USA, pp. 313–322.

Nehaniv, C. L. (2005), Sensorimotor experience and its metrics, *in* 'Proc. 2005 IEEE Congress on Evolutionary Computation', Vol. 1, IEEE Press, Edinburgh, Scotland, pp. 142–149.

Nehaniv, C. L. and Dautenhahn, K., eds (2007), *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, Cambridge Univ. Press.

Nehaniv, C. L., Dautenhahn, K. and Loomes, M. J. (1999), Constructive biology and approaches to temporal grounding in post-reactive robotics, *in* G. T. McKee and P. Schenker, eds, 'Sensor Fusion and Decentralized Control in Robotics Systems II (September 19-20, 1999, Boston, Massachusetts), Proceedings of SPIE', Vol. 3839, pp. 156–167.

Nehaniv, C. L., Polani, D., Dautenhahn, K., te Boekhorst, R. and Cañamero, L. (2002), Meaningful information, sensor evolution, and the temporal horizon of embodied organisms, *in* 'Artificial Life VIII', MIT Press, pp. 345–349.

Nehmzow, U. (2003), 'Quantitative analysis of robot-environment interaction - towards scientific mobile robotics', *International Journal of Robotics and Autonomous Systems* **44**, 55–68.

Nehmzow, U. and Smithers, T. (1991), Mapbuilding using self-organising networks in "really useful robots", *in* J. Meyer and S. Wilson, eds, 'From Animals to Animats, Proc. 1st conference on Simulation of Adaptive Behaviour', MIT Press.

Nuxoll, A. and Laird, J. E. (2004), A cognitive model of episodic memory integrated with a general cognitive architecture, *in* M. Lovett, C. Schunn, C. Lebiere and P. Munro, eds, 'Proceedings of the Sixth International Conference on Cognitive Modeling', Lawrence Erlbaum Associates, pp. 220–225.

Oates, T., Schmill, M. D. and Cohen, P. R. (2000), A method for clustering the experiences of a mobile robot that accords with human judgements, *in* 'Proceedings of the Seventeenth International Conference on Artificial Intelligence', AAAI Press, Menlo Park, California, pp. 846–851.

Ogino, M., Ooide, T., Watanabe, A. and Asada, M. (2007), Acquiring peekaboo communication: Early communication model based on reward prediction, *in* 'Proceedings of the 6th IEEE International Conference on Developoment and Learning', IEEE CDROM.

Olsson, L. A. (2006), Information Self-structuring for Developmental Robotics: Organization, Adaptation and Integration, PhD thesis, University of Hertfordshire.

Olsson, L., Nehaniv, C. L. and Polani, D. (2004), Sensory channel grouping and structure from uninterpreted sensor data, *in* '2004 NASA/DoD Conference on Evolvable Hardware June 24-26, 2004 Seattle, Washington, USA', IEEE Computer Society Press, pp. 153–160.

Olsson, L., Nehaniv, C. L. and Polani, D. (2006*a*), 'From unknown sensors and actuators to actions grounded in sensorimotor perceptions', *Connection Science* **18**(2), 121–144.

Olsson, L., Nehaniv, C. L. and Polani, D. (2006*b*), Measuring informational distances between sensors and sensor integration, *in* 'Artificial Life X', MIT Press.

OpenCV (2000), 'Open computer vision library (gpl licence)', http://sourceforge.net/projects/opencvlibrary/.

Oudeyer, P.-Y., Kaplan, F., Hafner, V. and Whyte, A. (2005), The playground experiment: Task-independent development of a curious robot, *in* D. Bank and L. Meeden, eds, 'Proc. of the AAAI Spring Symposium on Developmental Robotics, 2005', pp. 42–47.

Pea, R. D. (2004), 'The social and technological dimensions of scaffolding and related theoretical concepts for learning, education and human activity', *The Journal of the Learning Sciences* **13**(3), 423–451.

Pfeifer, R. and Bongard, J. (2007), *How the Body Shapes the Way We Think: A New View of Intelligence*, MIT Press.

Pfeifer, R. and Scheier, C. (1999), *Understanding intelligence*, MIT Press, Cambridge, MA, USA.

Poelz, P. M. and Prem, E. (2003), 'Concept acquisition using isomap on sensorimotor experiences of a mobiole robot', **101**, 175–176.

Prem, E., Hoertnagl, E. and Poelz, P. (2003), Anchoring symbols in hybrid autonomous systems using isomap sequences, *in* 'Design and application of hybrid intelligent systems', IOS Press, Amsterdam, The Netherlands, The Netherlands, pp. 1114–1123.

Quick, T., Dautenhahn, K., Nehaniv, C. L. and Roberts, G. (1999), The essence of embodiment: A framework for understanding and exploiting structural coupling between system and environment, *in* 'Proc. Third International Conference on Computing Anticipatory Systems (CASYS'99)', Liège, Belgium.

Ram, A. and Santamaria, J. C. (1997), 'Continuous case-based reasoning', *Artificial Intelligence* **90**(1-2), 25–77.

Rizzolatti, G., Fadiga, L., Gallese, V. and Fogassi, L. (1996), 'Premotor cortex and the recognition of motor actions', *Cognitive Brain Research* **3**(2), 131–141.

Robins, B., Dautenhahn, K., Nehaniv, C. L., Mirza, N. A., François, D. and Olsson, L. (2005), Sustaining interaction dynamics and engagement in dyadic child-robot interaction kinesics: Lessons learnt from an exploratory study, *in* 'Proc. of the 14th IEEE International Workshop on Robot and Human Interactive Communication, RO-MAN2005'.

Rochat, P., Querido, J. G. and Striano, T. (1999), 'Emerging sensitivity to the timing and structure of protoconversation in early infancy', *Developmental Psychology* **35**(4), 950–957.

Rolls, E. T. (1999), *The Brain and Emotion*, Oxford University Press.

Romano, D. and Wong, A. (2004), Personality model of a social character, *in* 'Empathy in Human Computer Interaction Workshop, HCI 2004, the 18th British HCI Group Annual Conference'.

Rosenbloom, P. S., Laird, J. E. and Newell, A., eds (1993), *The Soar Papers: Research on Integrated Intelligence*, MIT Press, Cambridge, MA.

Rosenfield, I. (1988), *The Invention of Memory: A New View of the Brain*, Basic Books: New York.

Rosenlicht, M. (1985), *Introduction to Analysis*, new edn, Dover Publications.

Rosson, M. B. and Carroll, J. M. (2002), *Usability Engineering: Scenario-based Development of Human-Computer Interaction*, Morgan Kaufmann Publishers.

Rylatt, R. M. and Czarnecki, C. (2000), 'Embedding connectionist autonomous agents in time: The 'road sign problem'', *Neural Processing Letters* **12**(2), 145–158.

Schank, R. C. (1999), *Dynamic Memory Revisited*, Cambridge University Press.

Schank, R. C. and Abelson, R. P. (1977), *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*, L. Erlbaum, Hillsdale, NJ.

Scheult, M. and Logan, B. (2001), Affective vs deliberative agent control, *in* 'Proceedings AISB 2001', pp. 1–10.

Shanahan, M. (2006), 'A cognitive architecture that combines internal simulation with a global workspace', *Consciousness and Cognition* **15**, 433–449.

Siebra, S., Salgado, A. and Tedesco, P. (2007), 'A contextualised learning interaction memory', *Journal of the Brazilian Computer Society* **13**(3), 51–66.

Sporns, O. and Pegors, Teresa, K. (2004), Information-theoretical aspects of embodied artificial intelligence, *in* F. Iida, R. Pfeifer, L. Steels and Y. Kuniyoshi, eds, ‘Embodied Artificial Intelligence’, Vol. 3139 of *Lecture Notes in Artificial Intelligence*, Springer Berlin / Heidelberg, pp. 74–85.

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

Tarapore, D., Lungarella, M. and Gómez, G. (2004), Fingerprinting agent-environment interaction via information theory, *in* F. Groen, N. Amato, A. Bonarini, E. Yoshida and B. Kröse, eds, ‘Proc. of the 8th International Conference on Intelligent Autonomous Systems’, Amsterdam, The Netherlands, pp. 512–520.

Tarapore, D., Lungarella, M. and Gómez, G. (2006), ‘Quantifying patterns of agent–environment interaction’, *Robotics and Autonomous Systems* **54**(2), 150–158.

te Boekhorst, I. J. A., Lungarella, M. and Pfeifer, R. (2003), Dimensionality reduction through sensori-motor coordination, *in* O. Kaynak, E. Alpaydin, E. Öja and L. Xu, eds, ‘Proc. of the Joint International Conference on Artificial Neural Networks and Neural Information Processing. Istanbul, Turkey’, Lecture Notes in Computer Science 2114, Springer, Berlin, pp. 496–503.

Thelen, E. and Smith, L. B. (1994), *A Dynamic Systems Approach to the Development of Cognition and Action*, MIT Press.

Thieme, M. and Ziemke, T. (2002), The road sign problem revisited: handling delayed response tasks with neural robot controllers, *in* ‘ICSAB: Proceedings of the seventh international conference on simulation of adaptive behavior on From animals to animats’, MIT Press, Cambridge, MA, USA, pp. 228–229.

Tidwell, J. (1999), ‘Common ground: A pattern language for human-computer interface design’, http://www.mit.edu/~jtidwell/interaction_patterns.html.

Tomlinson, B. and Blumberg, B. (2003), Alphawolf: Social learning, emotion and development in autonomous virtual agents, *in* ‘Innovative Concepts for Agent-

Based Systems', Vol. 2564 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 35–45.

Touretzky, D. and Tira-Thompson, E. (2004), 'Tekkotsu', *Carnegie Mellon University, Online Open Source, http://www.tekkotsu.org* .

Tulving, E. (1983), *Elements of Episodic Memory*, Oxford: Clarendon Press.

Tulving, E. and Craik, F., eds (2005), *The Oxford Handbook of Memory*, Oxford University Press.

van Duijn, M., Keijzer, F. and Franken, D. (2006), 'Principles of minimal cognition: Casting cognition as sensorimotor coordination', *Adaptive Behavior* **14**(2), 157–170.

Varela, F. J., Thompson, E. and Rosch, E. (1991), *The embodied mind: Cognitive science and human experience*, MIT Press, Cambridge, MA, USA.

Veatch, T. C. (1998), 'A theory of humour', *International Journal of Humour Research* **11**(2), 161–175.

Vogel, D. (2005), 'A neural network model of memory and higher cognitive functions', *International Journal of Psychophysiology* **55**(1), 3–21.

von Hofsten, C. (1993), 'Prospective control : A basic aspect of action development', *Human Development* **36**, 253–270.

Vygotsky, L. (1978), *Mind and society: The development of higher mental processes*, Harvard University Press., Cambridge, MA.

Weng, J., Evans, C., Hwang, W. S. and Lee, Y. B. (1999), The developmental approach to artificial intelligence: Concepts, developmental algorithms and experimental results, *in* 'NSF Design & Manufacturing Grantees Conference'.

Wermter, S. and Elshaw, M. (2003), 'Learning robot actions based on self-organising language memory', *Neural Networks* .

Wisconsin Medical Society, T. (2007), ‘Savant syndrome: Islands of genius’.