ROBOT TOOL BEHAVIOR:
A DEVELOPMENTAL APPROACH TO AUTONOMOUS TOOL USE

A Dissertation
Presented to
The Academic Faculty

by

Alexander Stoytchev

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Computer Science

College of Computing
Georgia Institute of Technology
August 2007

ROBOT TOOL BEHAVIOR:
A DEVELOPMENTAL APPROACH TO AUTONOMOUS TOOL USE

Approved by:

Ronald Arkin, Advisor
College of Computing
Georgia Institute of Technology

Aaron Bobick
College of Computing
Georgia Institute of Technology

Tucker Balch
College of Computing
Georgia Institute of Technology

Charles Isbell
College of Computing
Georgia Institute of Technology

Harvey Lipkin
Mechanical Engineering
Georgia Institute of Technology

Date Approved: 8 June, 2007

To my late grandfather,

who was a carpenter and taught me

how to use the tools in his workshop when I was a child.

He wanted me to become an architect. I hope that being

a roboticist would have been a distant second on his list.

PREFACE

I was fortunate to stumble upon the topic of autonomous tool use in robots for my dissertation. When I joined the robotics lab at Georgia Tech I began to look for a good dissertation topic. For almost two years I was reading the robotics literature with that goal in mind. I enjoyed reading but the more I read the more I felt that all the good dissertation topics have already been taken. So I started reading books from other fields of science in hope of finding an inspiration. Finally, the years of reading paid off when I stumbled across Thomas Power's book entitled "Play and Exploration in Children and Animals."

At first I was angry at myself for even borrowing a book from the library about children playing with toys (that's what the picture on the front cover showed). Nevertheless, I decided to browse through the pages. Several pages contained tables that compared the known object exploration strategies in animals and humans. The book also contained a reprint of Benjamin Beck's taxonomy of tool using modes in animals (see Table 2.3). It immediately struck me that the problem of autonomous tool use has not been well addressed in robotics. Within five minutes I had already decided to abandon all of my other research ideas and start working in this new direction.

The more I read about this new topic the more I liked it. The more I read, however, the harder it was to figure out how to even begin to address this enormous topic. From the very beginning I was interested in the developmental aspect of tool use. One of the biggest challenges was to focus my ideas on creating a well defined developmental trajectory that the robot can take in order to learn to use tools autonomously, which is the topic of this dissertation.

ACKNOWLEDGEMENTS

One of the main reasons why I picked robotics as my area of specialization is because it is inherently multi-disciplinary. I never expected, however, that my own research will take on a journey through so many different disciplines: ethology, anthropology, primatology, neuroscience, psychophysics, developmental psychology, computer graphics, computer vision, dynamics, and, of course, robotics. Ultimately, it was a very satisfying experience although I have to admit that it did not always feel that way. There are many people that I would like to thank for acknowledge their support during my years as a graduate student.

First of all, I would like to thank the members of my dissertation committee: Ron Arkin, Aaron Bobick, Tucker Balch, Charles Isbell, and Harvey Lipkin.

My advisor, Ron Arkin, deserves a special thank you for his seemingly infinite patience which I tested on many occasions. He understood the importance of the problem that I picked for my dissertation and gave me unlimited academic freedom to pursue it. At the same time, however, he kept me on my toes by asking for new experimental results and conference papers which were not always forthcoming. I appreciated Ron's dependability and was impressed by his time management skills. I knew I could always find him in his office at 6am, even on the first day he came back from sabbatical.

Aaron Bobick was always a good mentor and supporter. In my discussions with him he was always one step ahead of me. He challenged me and tried to push my research in directions that would make a compelling demonstration of its potential. I also enjoyed his classes and I learned a lot from him about what it takes to be a good teacher.

Tucker Balch is partially responsible for me coming to Atlanta. I got to know him even before I came to the US as he was one of the few people that took the time to reply to my ridiculously long e-mail with questions about the Ph.D. program in robotics at Georgia Tech. Tucker is also an alumni of the Mobile Robotics Lab at Tech and his advice on lab-related issues has been invaluable to me.

Charles Isbell actually volunteered to be on my dissertation committee because he liked my topic. I really enjoyed our discussions about machine learning and its applications to

robotics.

Harvey Lipkin is the only professor in my long career as a student that tried to do something about my messy handwriting. He tactfully handed back my midterm exam with a pencil and an eraser attached to it. This nice gesture might have come too late to improve my handwriting, but I surely learned a lot about robot control and dynamics in his class. This knowledge came in very handy when I was programming my dynamics robot simulator.

I also would like to thank Chris Atkeson and Sven Koenig who served on my qualifying exam committee. Sven also served on my dissertation proposal committee. Formerly at Georgia Tech, they have since moved to Carnegie Mellon and USC, respectively, but I still miss our conversations.

I also would like to thank the funding agencies and sponsors that paid for my graduate student salary. The Yamaha Motor Company sponsored my first project as a graduate student; the research work on that project solidified my belief that robotics is the field that I want to be in. The Humanoids research group at Honda R&D in Tokyo, Japan sponsored my second research project. In April of 2000 I was fortunate to be among the few people in the world to see a live demo of their humanoid robot P3. This first-hand experience made me a believer that one day our world would be inhabited by robots walking and working beside us. I strove to make a contribution that will bring us closer to that vision. Next, I worked on three DARPA-funded projects: Tactical Mobile Robotics (TMR), Mobile Autonomous Robot Software (MARS), and Mobile Autonomous Robot Software vision 2020 (MARS-2020). I am confident that there are few experiences in life that can match the excitement of a large-scale DARPA demo. I was fortunate to take part in two such demos at Rockville, Maryland and Fort Benning, Georgia.

Many thanks to my fellow graduate students from the Mobile Robotics Lab: Yoichiro Endo, Zsolt Kira, Alan Wagner, Patrick Ulam, Lilia Moshkina, Eric Martinson, Matt Poweres, Michael Kaess, and Ananth Ranganathan. The long hours in the lab and the cold mornings of DARPA demo days were more enjoyable when you were around.

In the Fall of 2005 I had the opportunity and the pleasure of teaching the first ever graduate class in Developmental Robotics. This dissertation, which was finished after the class was over, is a lot more coherent and readable, I think, because of this class. As they say you never know something until you have to explain it in the classroom. Therefore, I would like to thank the students in my graduate class (Adam, Adrian, Allen, Bradley, Dae-Ki, Dustin, Flavian, Georgi, Jacob, Jesse, Jie, Jivko, John, Kevin, Kewei, Lou, Matthew M., Matthew P., Michael, Oksana, Ryan, Tanasha, and Tyler) for the many stimulating class discussions and insightful answers to the open questions in Developmental Robotics that I sneaked into their homework assignments. I also would like to thank the chair of the Computer Science Department at Iowa State University, Carl Chang, for giving me the opportunity to teach this class. Many thanks to my ISU colleague Vasant Honavar who encouraged me to teach a new experimental class in robotics as my first class.

Many thanks are also due to my colleagues from the Virtual Reality Applications Center at Iowa State University: Jim Oliver, Derrick Parkhurst, and Eliot Winer. They provided me with perfect working conditions at the start of my career and gave me enough breathing room to finish this dissertation.

I also would like to thank the GVU center at Georgia Tech, the National Science Foundation (NSF), and the American Association for Artificial Intelligence (AAAI) for awarding me several travel grants, which allowed me to attend academic conferences in my area. The Bulgarian branch of the *Open Society Foundation* paid for my first airplane ticket to the United States and for two years of my undergraduate education at the American University in Bulgaria, which is deeply appreciated.

I would like to thank my Bulgarian friends at Tech, Ivan Ganev and Yavor Angelov, for their technical help and encouragement over the years.

I also would like to thank my mother and my two sisters for believing in my abilities and encouraging me to pursue my dreams. They were certainly not happy about me going to graduate school seven thousand miles away from home but they knew that this was the best thing for me and did not try to stop me. My late father did not live to see me off to America. He would have been very proud of me.

Finally, I would like to thank Daniela. Only she knows what I had to go through to get to this point. Her love and support kept me going through all these years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The ability to use tools is one of the hallmarks of intelligence. Tool use is fundamental to human life and has been for at least the last two million years. We use tools to extend our reach, to amplify our physical strength, and to achieve many other tasks. A large number of animals have also been observed to use tools. Despite the widespread use of tools in the animal world, however, studies of autonomous robotic tool use are still rare.

This dissertation examines the problem of autonomous tool use in robots from the point of view of developmental robotics. Therefore, the main focus is not on optimizing robotic solutions for specific tool tasks but on designing algorithms and representations that a robot can use to develop tool-using abilities.

The dissertation describes a developmental sequence/trajectory that a robot can take in order to learn how to use tools autonomously. The developmental sequence begins with learning a model of the robot's body since the body is the most consistent and predictable part of the environment. Specifically, the robot learns which perceptual features are associated with its own body and which with the environment. Next, the robot can begin to identify certain patterns exhibited by the body itself and to learn a robot body schema model which can also be used to encode goal-oriented behaviors. The robot can also use its body as a well defined reference frame from which the properties of environmental objects can be explored by relating them to the body. Finally, the robot can begin to relate two environmental objects to one another and to learn that certain actions with the first object can affect the second object, i.e., the first object can be used as a tool.

The main contributions of the dissertation can be broadly summarized as follows: it demonstrates a method for autonomous self-detection in robots; it demonstrates a model for extendable robot body schema which can be used to achieve goal-oriented behaviors, including video-guided behaviors; it demonstrates a behavior-grounded method for learning the affordances of tools which can also be used to solve tool-using tasks.

CHAPTER I

INTRODUCTION

1.1 Tool use in animals and robots

The ability to use tools is one of the hallmarks of intelligence. Tools and tool use are fundamental to human life and have been for at least the last two million years (Reed, 1988, p. 77). We use tools to extend our reach, to amplify our physical strength, to transfer objects and liquids, and to achieve many other everyday tasks. Some hand tools like the axe, knife, and needle are almost indispensable for any human society (Campbell, 1984). It is hard to even imagine what our lives would be like without tools.

For many years it was believed that humans were the only tool-using species on Earth. In fact, tool use was considered to be one of the key features that distinguished us from other animals. Starting with Darwin (1871), however, this belief was challenged more and more vigorously.

During the last century, substantial evidence has been collected which clearly demonstrates that a large number of animals from different phyletic groups use tools both in captivity and in the wild (Beck, 1980). Some birds, for example, use twigs or cactus pines to probe for larvae in crevices in the bark of trees which they cannot reach with their beaks (Lack, 1953). Egyptian vultures use stones to break the hard shells of ostrich eggs (Goodall and van Lawick, 1966; Alcock, 1970). Chimpanzees use stones to crack nuts open (Struhsaker and Hunkeler, 1971; Boesch and Boesch, 1983, 1990), sticks to reach food (Köhler, 1931), dig holes, or attack predators (van Lawick-Goodall, 1970). Orangutans fish for termites with twigs and grass blades (Ellis, 1975; Parker, 1968) and use sticks as levers to pry open boxes or crevices (Ellis, 1975; Darwin, 1871). Sea otters use stones to open hard-shelled mussels (Hall and Schaller, 1964; Riedman, 1990). Horses and elephants use sticks to scratch their bodies (Chevalier-Skolnikoff and Liska, 1993). These examples and many more (see (Beck, 1980) and (van Lawick-Goodall, 1970) for a detailed overview) suggest that the ability to use tools is an adaptation mechanism used by many organisms to overcome the limitations imposed on them by their anatomy.

Despite the widespread use of tools in the animal world, however, studies of *autonomous* robotic tool use are still rare. There are industrial robots that use tools for tasks such as welding, cutting, and painting, but these operations are carefully scripted by a human programmer. Robot hardware capabilities, however, continue to increase at a remarkable rate. Humanoid robots such as Honda's Asimo, NASA's Robonaut, and Sony's Qrio feature motor capabilities similar to those of humans (Hirai et al., 1998; Ambrose et al., 2000; Fujita et al., 2003). In the near future similar robots will be working side by side with humans in homes, offices, hospitals, and in outer space. However, it is difficult to imagine how these robots that will look like us, act like us, and live in the same physical environment like us, will be very useful if they are not capable of something so innate to human culture as the ability to use tools. Because of their humanoid "anatomy" these robots undoubtedly will have to use external objects in a variety of tasks, for instance, to improve their reach or to increase their physical strength. These important problems, however, have not been well addressed by the robotics community to date.

This dissertation investigates if robots can be added to the list of tool-using "species". More specifically, the dissertation investigates computational representations and algorithms which facilitate the development of tool-using abilities in autonomous robots.

The experiments described in this dissertation were inspired and influenced by the long history of tool-using experiments with animals (summarized in Section 2.1.3). The methods and experiments described here were also influenced by psychological, ethological, and neuroscience research. No claim is made, however, that this dissertation attempts to model how human or animal tool-using abilities are encoded or developed. The related work in these fields of science serves only as an inspiration to this robotics research.

1.2 Research Questions

Research Question

Can autonomous robots be effective tool-using agents?

The problem of autonomous tool use in robots may be addressed in multiple ways. This dissertation, however, examines this problem from the point of view of developmental robotics, which is one of the newest branches of robotics (Weng et al., 2001; Zlatev and Balkenius, 2001). Therefore, the focus of this dissertation is not on optimizing robotic solutions for specific tool tasks but on designing algorithms and representations that a robot can use to develop tool-using abilities.

In order to answer the main research question this work investigates three subsidiary questions.

Subsidiary Question 1

Can a robot identify which sensory stimuli are produced by its own body and which are produced by the external world?

Several major developmental theories have proposed that development requires "an initial investment in the task of differentiating the self from the external world" (Watson, 1994). In other words, normal development requires self-detection abilities in which the "self" emerges from actual experience and is not innately predetermined (Watson, 1994). There is also evidence from primate studies that the most proficient tool-using species are those that can make a clear distinction between the tool and their own bodies (Povinelli et al., 2000).

This research question explores a method for autonomous self-detection in robots. This question also evaluates if the results of the self-detection method can be used by the robot to classify visual stimuli as either "self" or "other."

Subsidiary Question 2

Can a robot learn a pliable sensorimotor model of its own body and morph this model to facilitate goal-oriented tasks?

The neuroscience literature tells us that the brain keeps and constantly updates a model of the body called *body schema*. This model is not static and can be extended by external objects such as tools in a matter of seconds. For example, Iriki et al. (1996) have shown that the body representation of a monkey can be extended when the monkey is holding a tool and this extension may be important for tool-using behaviors. A similar extension of the body occurs when a monkey observes its own hand in a TV monitor (Iriki et al., 2001). This extension of the body allows the monkey to perform video-guided behaviors during which it can observe its hand movements only through the TV monitor.

This research question investigates if a robot can learn a sensorimotor model of its own body from self-observation data. This question also investigates if the pliability of this representation can help the robot achieve tool-using tasks and video-guided behaviors.

Subsidiary Question 3

Can a robot use exploratory behaviors to both learn and represent the functional properties or affordances of tools?

The related work on animal object exploration indicates that animals use stereotyped exploratory behaviors when faced with a new object (see Section 2.1.1). For some species of animals these tests include almost their entire behavioral repertoire (Lorenz, 1996). Recent studies with human subjects also suggest that the internal model that the brain uses to represent a new tool might be encoded in terms of specific past experiences (Mah and Mussa-Ivaldi, 2003).

This research question evaluates whether a robot can use exploratory behaviors to autonomously learn the functional properties or affordances (Gibson, 1979) of tools. This question also investigates if a robot can use this behavior-grounded affordance representation to solve tool-using tasks.

1.3 Contributions of the Dissertation

This dissertation makes the following contributions to the field of robotics:

- It demonstrates a method for autonomous self-detection in robots (Chapter 5).
- It demonstrates a model for extendable robot body schema (Chapter 6).
- It demonstrates a method for learning the affordances of tools by a robot (Chapter 7).

Although researchers in robotics have addressed issues such as manipulation and grasping of objects they generally have not treated these objects as tools. Rather, they have addressed the problem of holding an object for the purposes of transporting it. Some industrial applications of robot arms use specialized instruments, but they are usually firmly attached to the end effector and thus do not qualify as tools (see definitions in Section 2.1). Furthermore, the control laws for these instruments are usually provided by a human programmer or in some cases learned from human demonstration. Thus, no existing work to date in the robotics literature, to my knowledge, has attempted to investigate in a principled manner the general problem of handling external objects as tools and learning their affordances.

This dissertation makes a step toward the goal of building intelligent robots that can adapt to their environment by extending their capabilities through the use of tools. Robots that are capable of so doing will be far more useful than the robots of today. Abilities for autonomous tool use would be especially useful for humanoid robots operating in human-inhabited environments. Because of their humanoid anatomy, these robots will be faced with many of the same challenges that humans face in their daily lives. Overcoming these challenges will undoubtedly require the use of external objects as tools in a variety of tasks.

Planetary exploration missions may also benefit from research on autonomous robot tool use. For example, some NASA plans for space exploration missions call for collecting soil and rock samples from distant planets and bringing them back to Earth. This task requires using a shovel or tongs (Li et al., 1996). In some cases a hammer must be used to expose the internal geological makeup of the rock samples (Li et al., 1996). In all of these scenarios it may be possible to engineer acceptable solutions by providing the robot with preprogrammed behaviors for tool use. However, the available tools may break or become deformed. Autonomous adaptation would be critical for the success of the mission in these situations.

In the near future, research on autonomous robotic tool use may play a major role in answering some of the fundamental questions about the tool-using abilities of animals and humans. After ninety years of tool-using experiments with animals (see Section 2.1.3) there is still no comprehensive theory attempting to explain the origins, development, and learning of tool behaviors in living organisms.

1.4 Overview

The rest of this document is organized as follows. Chapter 2 surveys the existing literature on tool use in several fields of science: ethology, psychology, anthropology, neuroscience, robotics, and artificial intelligence. Chapter 3 formulates some basic principles which are used to formulate a developmental approach to autonomous tool use in robots. Chapter 4 describes the evaluation platforms that were used to perform experiments. Chapter 5 describes an algorithm for autonomous self-detection in robots and the experimental conditions under which it was tested. Chapter 6 describes a computational model for an extendable robot body schema and shows how it can be used to facilitate tool-using tasks and video-guided behaviors. Chapter 7 describes a behavior-grounded approach to autonomous learning of tool affordances and shows how the learned affordances can be used to solve tool-using tasks. Chapter 8 draws conclusions and suggests directions for future work.

CHAPTER II

RELATED WORK

The interdisciplinary nature of this dissertation mandates an extensive overview of the existing body of literature in ethology, psychology, anthropology, robotics, and artificial intelligence. The goal of this chapter is to establish a solid theoretical foundation for the work presented in later chapters as well as to differentiate this research from previous work.

2.1 Tool Use

Before the related work is presented it is necessary to give a working definition for tool use. Several definitions for tool use have been given in the literature. The definition that is adopted throughout this work is the one given by Beck:

> *"Tool use is the external employment of an unattached environmental object to alter more efficiently the form, position, or condition of another object, another organism, or the user itself when the user holds or carries the tool during or just prior to use and is responsible for the proper and effective orientation of the tool."* (Beck, 1980, p. 10).

According to this definition, an object is considered to be a tool if it is not a part of the user's body. The user must be in physical contact with the tool (i.e., hold or carry it) during or right before its use. The tool must be used to act on an object, another organism, or the user itself. And finally, the user must orient the tool in a position that is effective for the current task.

Alternative definitions for tool use have been given by Alcock (1972), van Lawick-Goodall (1970), and Parker and Gibson (1977). They are similar to Beck's definition and therefore are listed here only for completeness.

> *"[Tool use is] the manipulation of an inanimate object, not internally manufactured, with the effect of improving the animal's efficiency in altering the form or position of some separate object."* (Alcock, 1972, p. 464)

> *"Tool use is the use of an external object as a functional extension of mouth or beak, hand or claw, in the attainment of an immediate goal."* (van Lawick-Goodall, 1970, p. 195).

> *"[Goal directed] manipulation of one detached object relative to another (and in some cases through a force field) involving subsequent change of state of one or both of the objects, e.g., hitting one object with another, either directly or by throwing, raking in one object with another, opening one object with another as a lever."* (Parker and Gibson, 1977).

2.1.1 Object Exploration

The definition for tool use given above is broad enough to allow for almost any environmental object to be a used as a tool if it is manipulable by the user. For this to happen, however, both the physical and functional properties of the object need to be explored and understood by its user through some form of object manipulation. The psychological literature distinguishes three main forms of such object manipulation: exploration, object play, and tool use (Power, 2000, p. 19-22).

- **Exploration** is used to reduce uncertainty about the object through information gathering. Exploration typically occurs during the first encounter with an object or a new environment. Its purpose is to answer the intrinsic question of "What does this object do?" Behaviorally it consists of a long stereotyped sequence of behaviors that rely on the synchrony between visual and tactile sensors.

- **Object Play** occurs after initial exploration of the object. It tries to answer the implicit query of "What can I do with this object?" It involves a series of short behavioral sequences that are highly variable and idiosyncratic in nature. In object play it is typical to observe little coordination between different sensory modalities.

- **Tool Use** is goal-directed behavior that has a clear objective to use the object as a means to an end. The object is used to achieve, more easily, specific behavioral goals.

For most common objects the first two modes of object manipulation occur in both animals and humans during the first years of life (Power, 2000). However, even adult

Table 2.1: Types of object manipulation behaviors identified in primates in multi-species studies. From Power (2000, p. 25).

Investigating	**Relating**	**Procuring**
Sniffing	Rubbing	Buccal prehension
Poking/touching	Rolling	Picking up
Mouthing/licking/biting	Draping/Wrapping	Holding
Scratching	Hitting/Striking	Carrying
Rotating	Dropping	Transferring
Bringing to Eyes	Lining Up	
Transforming	**Other large-motor behaviors**	
Tearing/breaking	Waving/shaking	
Twisting/untwisting	Pushing/ pulling	
Wadding/forming	Throwing	

animals and humans use exploratory behaviors when faced with a new object. Table 2.1 lists some commonly used object manipulation routines which are instrumental in gathering information about object properties such as mass, durability, stiffness, and ease of handling.

It has long been recognized that the opportunity to explore can serve as a very powerful motivator (Girdner, 1953; Montgomery and Segall, 1955). Numerous psychological theorists have written about the intense interest of infants and young children in everyday objects (Piaget, 1952; White, 1959; Hunt, 1965; Wenar, 1976; J., 1988). Several theories have postulated that animals explore in order to satisfy their "stimulus hunger" (Barnett, 1958) or that they seek stimuli "for their own sake" (Berlyne, 1966). As mentioned above, exploration typically occurs during an organism's first exposure to an object or when some change in the environment is observed (Berlyne, 1950; Inglis, 1983; Montgomery, 1953). In this respect exploration is similar to two other typical responses: fear and avoidance (Power, 2000, p. 19). Which of the three responses is chosen often depends on the degree of *neophobia* (i.e., fear of new things) induced by the object (Power, 2000, p. 19).

In any case, if exploration is chosen an interesting thing occurs: the object is subjected to a battery of tests in the form of exploratory behaviors. For some species of animals these tests include almost the entire behavioral repertoire of the animal. The following quote from Conrad Lorenz, one of the founding fathers of ethology, summarizes this point well:

> *"A young corvide bird, confronted with an object it has never seen, runs through practically all of its behavioral patterns, except social and sexual ones. It treats the object first as a predator to be mobbed, then as a dangerous prey to be killed, then as a dead prey to be pulled to pieces, then as food to be tasted and hidden, and finally as indifferent material that can be used to hide food under, to perch upon, and so on.*
>
> *[...] The appetite for new situations, which we usually call curiosity, supplies a motivation as strong as that of any other appetitive behavior, and the only situation that assuages it is the ultimately established familiarity with the new object - in other words, new knowledge.*
>
> *[...] In fact, most of the difference between man and all the other organisms is founded on the new possibilities of cognition that are opened by exploratory behavior."* (Lorenz, 1996, p. 44)

Table 2.2 lists some exploratory procedures and links them to object properties that can be learned by applying these procedures to the object.

Table 2.2: Postulated links between knowledge about objects and exploratory procedures that may be used to gain this knowledge. From Power (2000, p. 69).

KNOWLEDGE ABOUT OBJECT	**EXPLORATORY PROCEDURE**
Substance-related properties	
Texture	Lateral motion
Hardness	Pressure
Temperature	Static contact
Weight	Unsupported holding
Structure-related properties	
Weight	Unsupported holding
Volume	Enclosure, contour following
Global shape	Enclosure
Exact shape	Contour following
Functional properties	
Part motion	Part motion test
Specific motion	Function test

Power's and Lorenz's observations that object exploration can be achieved through active experimentation with stereotyped behaviors motivates our use of a similar approach to robotic tool exploration. Specifically, waving, pushing, and pulling behaviors will be used to learn about tools and their affordances.

2.1.2 Modes of Tool Use

The last mode of object manipulation listed in the previous section is tool use. According to Beck (1980), whose taxonomy is widely adopted today, most animals use tools for four different functions: 1) to extend their reach; 2) to amplify the mechanical force that they can exert on the environment; 3) to enhance the effectiveness of antagonistic display behaviors; and 4) to control the flow of liquids. Table 2.3 lists 19 modes of tool use that fit into one of these four categories (only two others do not fit). The limited number of additional uses shows that tool use has some well defined evolutionary advantages that are readily adopted by many species (Beck, 1980, p. 123).

The first function of tool use, extension of reach, is applicable to situations in which the animal is somehow prevented from getting close enough to some incentive. The tool is used to bring the incentive within the sphere of reach of the animal. If the incentive is located above the animal, the tool can be propped against a supporting structure in order to climb to the incentive, or several objects can be stacked on top of each other to achieve the same objective. If the incentive is another living animal, baiting can be used to catch it without scaring it away. Inserting and probing is used when the incentive is located in narrow crevices or holes which may be within the sphere of reach but are inaccessible because the animal's prehensive structures are too thick (Beck, 1980, p. 123).

The second function of tool use is to amplify the mechanical force that an animal can exert on the environment. The six modes listed under this function in Table 2.3 work by increasing the mass, decreasing the elasticity, and/or increasing the speed of the animal's prehensive structures. For example, clubbing or hitting increases the speed and thus the force of a swing (Beck, 1980, p. 123). Prying uses leverage to amplify the applied force. Prodding and jabbing deliver the force more effectively by concentrating it on a small area.

Another common use of tools in the animal world is to amplify aggressive or antagonistic behaviors. The modes that fall into this category are: drop or throw down, unaimed throw, brandish or wave, and drag, kick or roll (see Table 2.3). Orangutans, for example, have been observed to throw stones and sticks at human observers (Harrisson, 1963). Some squirrels try to scare away garter and rattle snakes by kicking sand into their faces (Owings et al., 1977; Owings and Cross, 1977).

The fourth function of tool-use is to control liquids more effectively. This mode includes containment, transportation and wiping of liquids. For example, chimpanzees use clumps of leaves to wipe mud, sticky foods, and blood from their bodies (Goodal, 1964; van Lawick-Goodall, 1970). They also use leaves as sponges to absorb water (Goodal, 1964; van Lawick-Goodall, 1970; McGrew, 1977). A laboratory crow at the University of Chicago was observed to carry water in a plastic cup in order to moisten its dry food (Beck, 1980, p. 29).

The robot tool tasks described in later chapters fall in the extension of reach category in Beck's taxonomy. Tasks from this category have been used for the last 90 years to test and formulate theories of tool use in animals. The next section reviews the most prominent of these studies.

Table 2.3: Modes of tool use in animals. An 'x' indicates that this mode has been observed in the members of a specific phyletic group. From Beck (1980, p.120-1).

Mode of Tool Use	Arthropods and Mollusks	Fish	Reptiles and Amphibians	Birds	Nonprimate Mammals	New World Monkeys	Old World Monkeys	Apes
Extend Reach								
Reach				x	x	x	x	x
Prop and Climb					x			x
Balance and climb						x		x
Stack						x		x
Bait				x	x	x	x	x
Insert and probe				x		x	x	x
Amplify Mechanical Force								
Pound, Hammer	x			x	x	x	x	x
Aimed throw	x	x		x	x	x	x	x
Club, Hit					x	x		x
Pry	x			x		x		x
Dig				x	x		x	x
Prod, Jab						x		x
Augment Antagonistic Display								
Drop, Throw Down				x		x	x	x
Unaimed Throw					x		x	x
Brandish, Wave	x			x	x		x	x
Drag, Kick, Roll							x	x
Control Flow of Liquids								
Wipe				x	x	x	x	x
Contain				x			x	x
Sponge, Absorb	x			x				x
Not Fitting Any of the Above								
Drape, Affix	x				x			x
Hang and swing							x	x

2.1.3 Animal Tool Use

There are numerous accounts of animal tool use, some of which were listed in Chapter 1. Table 2.3 (see previous section) lists the observed modes of tool use among the members of many phyletic groups of animals. Because many animal species, not only mammals, have been observed to use tools, the phenomenon "apparently does not require a highly evolved central nervous system" (Campbell, 1985, p. 277). Yet it must be cognitively complex since man and chimpanzees are the only truly proficient tool users (Tomasello and Call, 1997, p. 58). While many animals are capable of tool use, not all of them are considered to be *intelligent tool users* since their use of tools is limited to narrowly specialized feeding adaptations (Parker and Gibson, 1977). Therefore, the majority of tool use experiments have been performed with primates.

Wolfgang Köhler was the first to systematically study the tool behavior of chimpanzees. The goals of his research were to establish if chimpanzees "behave with intelligence and insight under conditions which require such behavior" (Köhler, 1931, p. 1). Köhler views apes as good experimental subjects because they make mistakes that expose their limited cognitive abilities and thus can serve as the basis for building a theoretical model of the "nature of intelligent acts." He notes that humans will not be good subjects since they rarely encounter simple tasks for the first time and even in unfamiliar situations they perform the task mentally before they act so their behaviors cannot be easily observed.[1]

Köhler performed a large number of experiments with nine chimpanzees from 1913 to 1917 while he was stranded on an island during the First World War. The experimental designs were quite elaborate and required use of a variety of tools: straight sticks, L-sticks, T-sticks, ladders, boxes, rocks, ribbons, ropes, and coils of wire. The incentive for the animal was a banana or a piece of apple which could not be reached without using one or more of the available tools. The experimental methodology was to let the animals freely experiment with the available tools for a limited time period. If the problem was not solved during that time, the experiment was terminated and repeated at some later time.

[1] He noted, however, that experiments with children (which had not been performed at the time of his writing) would also be valuable (Köhler, 1931, p. 268).

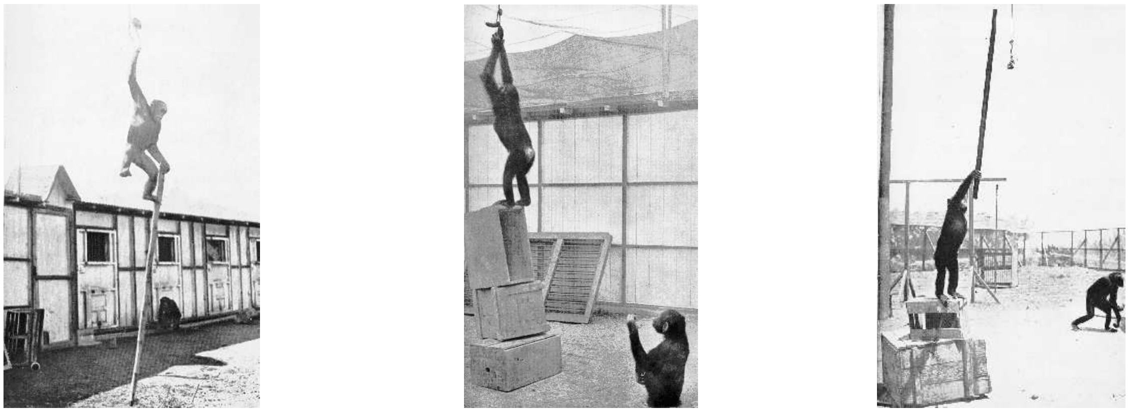


Figure 2.1: Three of the experiments performed by Köhler. The tools (boxes and sticks in these cases) were randomly placed in the environment. The chimpanzees had to learn how to use and/or arrange them to get the bananas.

In order to explain the abilities of apes to achieve complex tasks Köhler formulated a theory which he called a *theory of chance*. It states that tasks are often solved by a "lucky accident" which may occur while performing an action possibly unrelated to the current objective. When solving a task, the animals try different actions in an attempt to solve a "half-understood" problem. The solution may arise by some "chance outcome" of these actions (Köhler, 1931, p. 193). Köhler also observed that "impatience and anger in complicated cases take over" (Köhler, 1931, p. 262). The same was observed by Klüver (1933). Beck (1980) also witnessed the same phenomena and hypothesized that observing the outcomes of misplaced aggression and anger can lead to accidental discovery of tool use.

Köhler divides the errors that his subjects made into three main categories from the point of view of the observer: 1) *good errors* - those that make a "favorable impression" to the observer; the animal almost gets the task done; 2) *errors caused by complete lack of comprehension of the conditions of the task* - as if the animal has some "innocent limitation" which prevents it from grasping the task; and 3) *"crude stupidities"* - arising from habits in seemingly simple situations in which the animal should be able to achieve the task.

Other pioneering studies were performed by Bingham (1929), Yerkes and Yerkes (1929), Klüver (1933), and Yerkes (1943). The conclusions reached by them were in agreement with Köhler's finding. Klüver (1933), for example, concluded that the monkeys often achieved success after performing a series of attempts in which they displayed a variety of behaviors

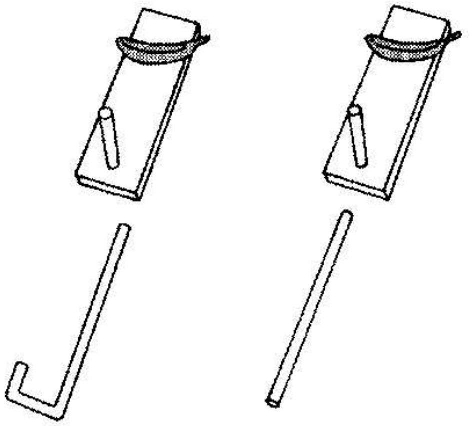


Figure 2.2: An experiment designed to test if chimpanzees appreciate the hook affordance of the tool. The task is to bring one of the platforms within reach using one of the tools and get the banana. See text for details. From Povinelli et al. (2000, p. 223).

with external objects. One of these behaviors eventually produced the desired result. However, more complicated problems requiring fine manipulation or precise ordering of actions often remained unsolved.

In more recent experimental work, Povinelli et al. (2000) replicated many of the experiments performed by Köhler and used statistical techniques to analyze the results. One experiment, shown on Figure 2.2, required the chimpanzees to choose the appropriate tool for bringing a platform with bananas within reach. The two tools were a hook positioned with the hook towards the ape and a straight stick. While this task may seem trivial, five of the seven chimpanzees chose the straight stick on their first trial. On subsequent trials they still did not show a clear preference for the hook which is the better tool for this task. Even more interestingly, the chimpanzees that picked the hook tool used it first without reorienting it. After their apparent lack of progress some of them reoriented the tool and tried again. The ones that picked the straight stick, however, also reoriented it even though this provides no advantage. Thus, the chimpanzees did not show that they understand the advantage that the hook offers in this task (Povinelli et al., 2000, p. 231).

The main conclusion reached by the researchers was that chimpanzees do not understand the functionalities of tools. Instead the solutions to the tool tasks may be due to simple rules extracted from experience like "contact between objects is necessary and sufficient to

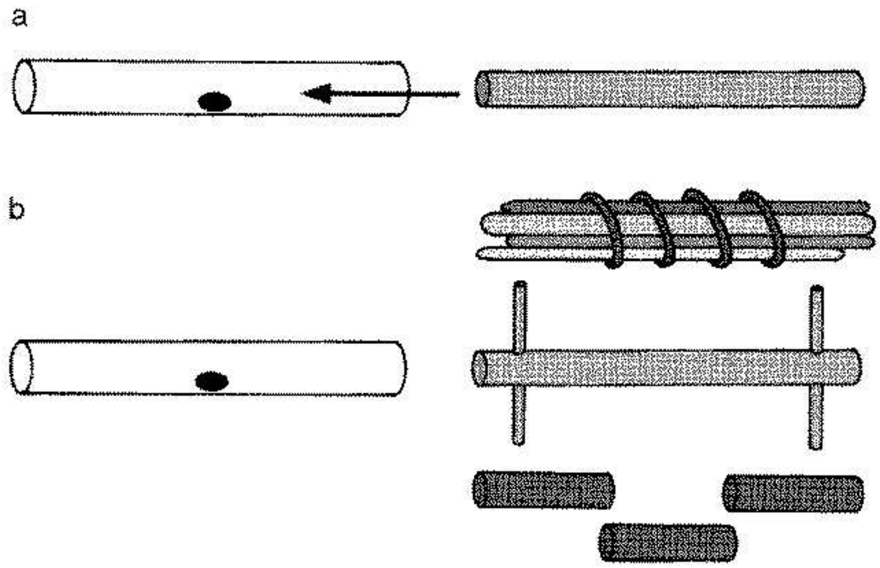


Figure 2.3: The tube problem described by Visalberghi (1993). (a) A peanut is placed inside a transparent plastic tube. To get the lure the monkey must use a stick to push it out of the tube. (b) The test tools given to the monkeys after training with the straight stick. See text for details.

establish covariation in movement" (Povinelli et al., 2000, p. 305) and "reorient the tool if there is no progress and try again" (Povinelli et al., 2000, p. 231). Thus, the chimpanzees may have only tried to establish contact between the tool and the platform regardless of the shape of the tool. When the tool did not cause the desired movement of the platform they either switched tools or reoriented the current tool (Povinelli et al., 2000, p. 305). Furthermore, it was concluded that chimpanzees do not reason about their own actions and tool tasks in terms of abstract unobservable phenomena such as force and gravity. Even the notion of contact that they have is that of "visual contact" and not "physical contact" or "support" (Povinelli et al., 2000, p. 260).

In another recent study, Visalberghi and Trinca (1989) tested capuchin monkeys on a tube task: a peanut was placed in the middle of a transparent tube whose length and diameter were such that the animal could not grab the peanut directly. After training with a straight stick a variety of tools were provided to the monkeys that could be used to push the lure out of the tube – short and long sticks, thick sticks, bundled sticks, and H-sticks with blocked ends (Figure 2.3). Although many of the test subjects eventually succeeded

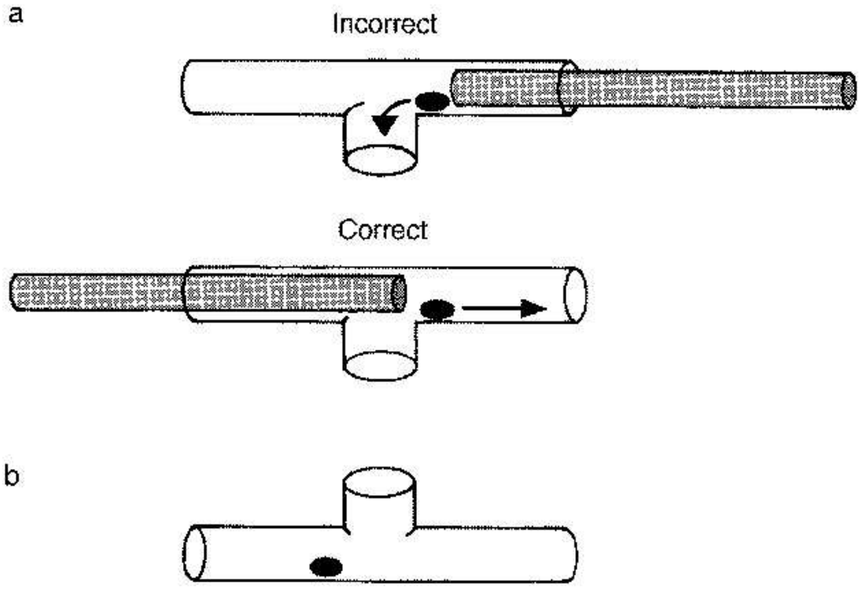


Figure 2.4: (a) The trap tube problem described by Visalberghi and Limongelli (1994); the end from which the stick is inserted affects the final outcome. (b) The inverted trap tube problem; the end from which the stick is inserted does not affect the final outcome.

in performing this task, they made a number of errors, for example, inserting a short stick into the tube and then inserting another short stick in the opposite side of the tube or freeing one end of an H-stick and then inserting the other end into the tube. This led the researchers to conclude that capuchin monkeys “do not abstract at a representational level the characteristics of the tool required by the task they face”. They also concluded that capuchins “do not acquire a general set of rules concerning the properties of the stick” (Visalberghi, 1993).

In a subsequent study the tube was modified to have a small trap hole right in the middle (Figure 2.4) so that if the food is pushed inside the hole it cannot be taken out (Visalberghi and Limongelli, 1994). Initially, it seemed that the animals learned the task. However, they did not learn which is the better end of the tube for inserting the stick and inserted it equiprobably into either side of the tube. Corrections were made based on the movement of the food and its location relative to the trap. Interestingly, the monkeys continued to avoid the trap even when the tube was rotated 180 degrees so that the trap was upside-down and could no longer capture the food. It seems that performance in this task was guided entirely by the position of the food relative to the hole. When it was observed that the food is moving towards the hole, the stick was inserted in the other direction (Visalberghi and Limongelli, 1994).

The results of the experiments described in this section seem to indicate that in many situations non-human primates do not have a clear idea about the functional properties of a tool and yet they are capable of solving tool tasks. It seems that they base their solutions on heuristics extracted from observable phenomena of their actions. In many cases movement of the attractor object is the most important thing to which they pay attention. This movement also serves as an indicator of their progress in the task. Another conclusion is that most experiments with new tools and tasks were solved purely by accident after a series of seemingly random exploratory behaviors.

Many of these observations are taken into account in Chapter 3 which formulates a theoretical model of autonomous robotic tool use. The robot experiments described in later chapters were inspired by the primate experiments described in this section.

2.2 Origins of Tool-Using Behaviors

Ethologists and anthropologists have identified several factors as important or contributing to the origins of animal and human tool use. The next two subsections describe the currently adopted theories in these two fields of science.

2.2.1 Ethological Viewpoint

Beck's explanation for discovery of tool use is trial-and-error learning. He argues that it is a process in which "the animal makes a variety of responses, one of which fortuitously produces reinforcement." (Beck, 1980, p. 141-2). This is similar to the operant conditioning training methods (Skinner, 1938) except that in the latter case the trainer uses a variety of techniques to elicit the first response. However, in the case of tool discovery, often there is no teacher to provide reinforcement.

In many cases it seems that the origins of a tool-using behavior may involve only a small change like a stimulus substitution in an existing behavioral sequence (Beck, 1980, p. 181). Another way to achieve the same result is to transfer an existing behavioral pattern to a slightly novel stimulus situation (Beck, 1980, p. 181). It is still debatable what triggers this substitution or transfer. Beck, however, suggests that observing the results of frustration and displaced aggression may lead to an accidental discovery of tool use (Beck, 1980, p. 181).

In the animal, world tool behaviors can also be learned through active tuition by a parent or through imitation (Kawamura, 1959; McGrew, 1978; de Waal, 2001). Active tuition, however, is very rare and some authors have even questioned whether it exists (Povinelli et al., 2000, p. 302). The woodpecker finch, for example, can learn to use twigs or cactus spines to push arthropods out of tree holes without social learning (Tebbich et al., 2001). In fact, this behavior can be acquired even by finches whose parents have never used tools to get their prey. Instead, the motivation provided by the environmental conditions seems to have a greater impact: dry environments have less food and stimulate the development of the behavior, while wet climates suppress it (Tebbich et al., 2001).

There are many ecological conditions that may have contributed to the evolution of tool use. The one condition on which most scientists agree, however, involves *extractive foraging* on *embedded foods* (Parker and Gibson, 1977; Beck, 1980). Examples of embedded foods include nuts, mollusks, and insects that dwell in underground nests, i.e., foods whose edible portions are protected by a hard shell or are hard to reach.

The evolutionary significance of tools for tool-using species has been seen by others as a compensation for the animal's "lack of biological equipment" or as a substitute for adaptation through morphology (Alcock, 1972; Parker and Gibson, 1977; Wilson, 1975). These species resort to the functionally equivalent behavior of tool use to compensate for their lack of anatomical specializations (Alcock, 1975).

A tool behavior can give a significant competitive advantage to the species that use that behavior. Alcock suggests that tool-using species have been able to invade ecological niches that are not at all characteristic of their phylogenetic groups (Alcock, 1972, p. 472). Tools may also provide a key advantage in the competition for resources with other species living in the same ecological niche. The tool-using species may obtain a monopoly or near monopoly of a specific resource and thus eliminate all competitors (Campbell, 1985, p. 277). Tool behaviors that increase or diversify food supplies indirectly improve the animal's reproductive chances and thus the survival of the species.

Another important factor in the evolution of tool behaviors is the presence of a powerful motivator. As mentioned above, food extraction is probably the most powerful motivator in the animal world. However, if food resources are plentiful and food can be obtained with little effort and bare hands tool behaviors are unlikely to evolve. This may have been the case with a group of wild African chimpanzees that use almost no tools and live in an area where food supplies are abundant (Poirier and McKee, 1999, p. 66).

These findings are insightful and at the same time quite relevant to robotics. The robots of today are becoming more capable and they are beginning to spread to environments inhabited by other species – the humans (Menzel and D'Aluisio, 2000). Since humans are tool-using species, the robots will have to "adapt" to the demands of their new ecological niches and become tool-using "species" themselves in order to compete successfully. Thus, in order to achieve the vision of mass adoption of robots into our everyday lives (discussed in Chapter 1) the fundamental questions of how to autonomously learn, represent, and use the functional properties of tools need to be addressed. Finding answers to these questions is the main motivation behind this dissertation.

2.2.2 Anthropological Viewpoint

The archaeological record left behind by our ancestors clearly indicates that tools have been an essential part of humanity for at least the last two million years. The oldest stone tools found date to between 2 to 2.5 million years ago. They are simple chips or pebbles from river gravels. Many of them were never shaped and can only be identified as tools if found together with shaped stones in caves where they do not occur naturally (Washburn, 1979, p. 9).

Over the ages, both the design of the tools and the methods of manufacture increased at a steady pace. Figure 2.5 shows four tools made during different stages of human evolution. Both the sophistication of the design and the number of operations needed to manufacture the tool increase from left to right. The most common manufacturing techniques also evolved from the simple striking of a stone with another stone, to using hard and soft materials as hammers to better control the flaking, and finally to grinding and polishing (Tattersall et al., 1988, p. 542-548). Producing a tool such as the one shown on the rightmost image in Figure 2.5 required nine major manufacturing steps and approximately 251 individual operations (Campbell, 1985, p. 398).

There is no archaeological evidence to suggest that stone tools were used earlier than 2.5 million years ago. However, it seems likely that other tools were used before that but they were not preserved. These tools were made of bio-degradable materials like bone, horn, teeth, leather, wood, bark, and leaves and stems of plants (Campbell, 1985, p. 278). Similar tools are used by modern day hunter-gatherer communities living in Africa and Australia. The most common tool used by these modern foragers is the digging stick which would not be preserved because of its composition (Poirier and McKee, 1999, p. 175). In fact, stone tools, because of their hardness, may have been invented in order to cut softer raw

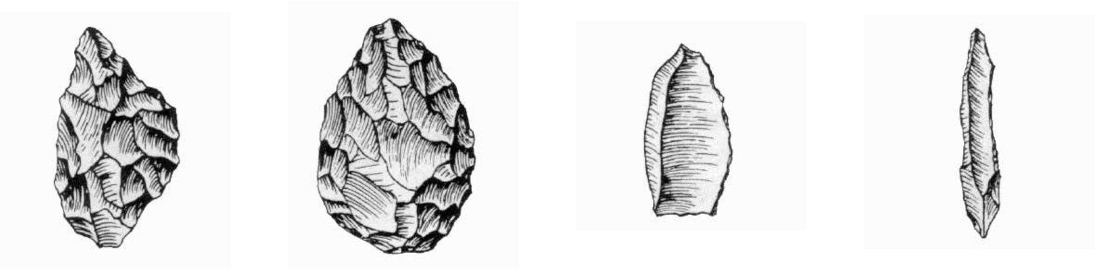


Figure 2.5: Progression of stone tool technology. Both the sophistication of the design and the number of operations needed to manufacture the tool increase as technology improves. From left to right: two hand axes made by Homo erectus; a scraping tool made in Neanderthal times; a stone knife made by Paleolithic people of modern type. From Campbell (1985, p. 398).

materials like wood and bone which were already being used for toolmaking (Campbell, 1985, p. 279).

The ability to use tools may have evolved gradually but the advantages that it offered were quite substantial. The quality of life made possible by tools changed the pressures of natural selection and even changed the structure of man (Washburn, 1979, p. 11). Darwin (1871) was among the first to suggest that tool use required free hands and thus may have been a selective evolutionary advantage towards the adoption of two-legged locomotion. Stone tools may have played an important role in the adoption of a carnivorous diet in the early hominids as well. Initially our ancestors had a vegetarian diet but gradually began scavenging the remains of dead animals killed by other carnivores. These leftovers had to be cut and it is unlikely that bare hands and teeth were sufficient for this job (Collins, 1976). In fact, it has been shown that some of the early stone tools have been used to cut meat by analyzing the microwear patterns of their edges (Keeley, 1979).

The benefits of tool use may have lead to the gradual increase of localization abilities and memory capacity for the purpose of remembering the location of raw tool materials. This would have been especially important in harsh environments where such materials are sparse. Some theories propose that certain gatherings of stone tools in large numbers were, in fact, early tool repositories (Potts, 1984). Remembering the location of these *stone* *caches* would be crucial for the survival of the tribe, especially for a hunter-gatherer tribe that may traverse tens of kilometers per day.

Calvin (1983) proposed an interesting theory according to which stone throwing (a specific mode of tool use) may have played an important role in the lateralization of the brain and even the development of language. Throwing a stone requires precise coordination of sequential one-handed motor skills and thus may have required increased cognitive abilities which concentrated in one of the brain's hemispheres. Language is another sequential skill that may have benefited from the cognitive abilities developed for stone throwing. In fact, the brain centers for throwing and language are located close to each other in the left hemisphere of the brain and may have evolved simultaneously (Kimura, 1979; Lieberman, 1991). Language may have evolved for another reason also connected with tools. The increasing importance of tools for the survival of the species may have put evolutionary pressure towards the development of a communication system. Without communication it is hard to refer to specific tools in daily activities and even harder to pass the ability to use and manufacture tools from one generation to the next (Poirier and McKee, 1999, p. 135).

The combined effect of increased hand manipulation required for tool use, carnivorous diet, and increasing requirements on the mental and communication abilities undoubtedly contributed to the increase in brain size (Poirier and McKee, 1999, p. 138). This provided more "computational resources" that could be used to invent new tools for other tasks.

The increasing sophistication of tools and tool-making technology over the last two million years has resulted in today's technological society. Almost every aspect of our lives today is affected by our abilities to use and manufacture tools. Table 2.4 divides human technology into seven main categories. According to this categorization, the invention of stone tools marks the beginning of human technology. Even though our ancestors may have been using tools long before that, technology did not start until they realized how to use tools to manufacture new tools. In fact, most primate animals living today are capable of tool use and tool modification but only man has been known to manufacture new tools using existing tools (Poirier and McKee, 1999, p. 68). A similar distinction is made between fire use and fire making. While the early hominids may have been using fire created by natural

Table 2.4: Categories of technology identified by Campbell (1985, p. 279).

1. Prototechnology	a) Tool use
	b) Tool modification
2. Technology	a) Tool manufacture
	b) Stone technology (and secondary tools)
3. Pyrotechnology	a) Fire use
	b) Fire control
	c) Fire making
	d) Metal industries (smelting, casting, forging)
4. Facilities	a) Containers, cords, etc.
	b) Energy control
5. Machines	
6. Instruments	
7. Computers	

means such as lightning and forest fires, many years would pass before they were capable of starting fire on their own (Poirier and McKee, 1999, p. 244). With the ability to make fire, metallurgy emerged which made the production of more complicated and more robust tools possible. From then on, human technology has advanced very rapidly to such an extent that our modern lives seem impossible without automobiles, computers, electricity, and telephones.

During the evolution of humankind, tools were used for different purposes (Table 2.4). The first tools were used to extend or amplify motor and physical abilities. Pounding, cutting, digging, and scraping provided a definite advantage; tools that could be used for these actions were in high demand (Washburn, 1979, p. 9). This trend was changed much later towards producing tools that amplify our sensing abilities. Instruments like the telescope and the microscope allowed people to see objects at great distances and magnify hard-to-see objects (Campbell, 1985, p. 397). Another trend has begun with the invention of the computer which promises to revolutionize our computational and planning abilities.

Similar to the history of evolution of human tool behavior, the robot experiments described in later chapters involve simple tools like sticks that extend the physical capabilities of the robot. Another reason for choosing the stick as the primary tool in those experiments comes from the psychological work discussed in the next section. According to some psychological theories the first intelligent behaviors in children are those that use a stick to bring distant objects within grasping range.

2.3 Psychological Influences

Psychologists have long been interested in the abilities of humans and animals to use tools to manipulate external objects. According to some psychological theories interactions with external objects are so important that they directly influence the development of human intelligence (Piaget, 1952). Other theories have focused on the capacity of tools to transform organisms into more capable ones (Gibson, 1979). Yet others, have focused on the differences between animal and human tool-using behaviors in search of the origins of human intelligence (Vauclair, 1984; Power, 2000). This section provides a brief summary of the psychological theories of tool use that have influenced this dissertation work.

2.3.1 Piaget

Jean Piaget formulated probably one of the most influential theories of child development. According to his theory, external objects play an important role in the development of human sensorimotor abilities. Piaget's theory also suggests that intelligent behaviors are first developed in the process of interaction with objects. The first manifestation of *inventive intelligence*, according to Piaget, is observed when a child successfully manages to bring distant objects closer by pulling the support on which they are placed or by using a stick to bring them within its field of prehension (Piaget, 1952, p. 280).

> *"The desire to strike or swing objects fortuitously reveals to the child the power of the stick when by chance the latter extends the action of the hand. [...] Thereafter, when the child aims to reach an object situated outside his field of prehension, it is natural that his desire should arouse the schemata in question. [...] It is not enough to strike an object with a stick in order to draw it to oneself [therefore] it is necessary to discover how to give an object an appropriate movement. [...] [T]he child, as soon as he sees the object being displaced a little under the influence of the stick's blows, understands the possibility of utilizing these displacements with the view of drawing the object in question*

> *to him. This comprehension is not only due to the initial schemata which are at the root of the subject's searching (schema of grasping and that of striking) [...] but it is also due to the "auxiliary" schemata which join themselves to the former."* (Piaget, 1952, p. 301-2).

According to Piaget, even the mathematical abilities of humans have their origins in object interaction as children first learn the concept of a number by counting external objects.

Piaget's theory divides the first two years of human life into six distinct stages (Piaget, 1952). Most of the perceptual and motor skills of the child are developed during these two years. With each additional stage, the behaviors of the child progress from simple to more intelligent ones. The role that external objects play in the development of the child also increases with each additional stage. A brief summary of the major developments in each stage is provided below.

- **Stage I: Reflex Structures (0-1 Month)**

  Piaget suggests that at birth children have no cognitive structures. Instead they have reflex structures for sucking, grasping, and crying. For example, newborn children close their hands when their palms are touched. Similarly, children start sucking any object that comes into contact with their lips (Piaget, 1952, p. 89).

- **Stage II: Primary Circular Reactions (1-4 Months)**

  The infant's reflex structures are gradually transformed into sensorimotor action schemas, which Piaget calls *primary circular reactions*. This happens after repeated use of the reflex structures, which the baby would apply to any object. For example, babies would grasp blankets, pillows, fingers, etc. Stage II infants, however, are not concerned with the objects around them and would not pay attention to the effects of their actions on the external world. They would execute an action even if it is not applied to any object. It is not uncommon for them to open and close their hands in mid-air. The repeated use of the action forms the primary circular reaction.

- **Stage III: Secondary Circular Reactions (4-8 Months)**

At the end of Stage II, infants are more capable of exploring their world. They can form associations between their actions and the results produced in the external environment. The child actively tries to reproduce or prolong these results. Through this repetition the child discovers and generalizes behavioral patterns that “produce and make interesting sights last” (Piaget, 1952, p. 171). Piaget calls these behavioral patterns *secondary circular reactions*. External objects play an important role in stage III but differences between objects are not noticed as the infant handles them in the same way.

- **Stage IV: Coordination of Secondary Schemes (8-12 Months)**

Stage IV marks the beginning of problem solving. Babies are now capable of coordinating their secondary schemas in a means-ends fashion. Instead of merely reproducing the results of their actions, children would use one schema to achieve a specific goal. That goal can simply be to use another schema. For example, in order to play with a toy covered with a cloth the child would first remove the cloth and then pick up the toy. Overcoming these obstacles with intermediate means and unforeseen difficulties requires adaptation of familiar schemas to the realities of new situations (Piaget, 1952, p. 213).

The coordination of secondary schemas gradually leads to their generalization and applicability to a growing number of objects (Piaget, 1952, p. 238). Children become capable of applying them to many new situations and that accelerates their exploration capabilities. One limitation of Stage IV is that infants are capable of employing only familiar schemes. If a task requires a new way of interaction with the world, the baby simply cannot solve it.

- **Stage V: Tertiary Circular Reactions (12-18 Months)**

Stage V babies are real explorers. They systematically vary and gradate their behaviors and pay close attention to the observed results. They discover new means through such active experimentation. Piaget calls these experiments *tertiary circular reactions* (Piaget, 1952, p. 267). The experiments are guided by the search of novelty as a goal in and of itself which is why Piaget also calls them ”experiments in order to see” (Piaget, 1952, p. 264). For the first time the child can truly adapt to unfamiliar situations by using existing schemas as well as actively seeking and finding new methods (Piaget, 1952, p. 265).

Stage V also marks the beginning of intelligent tool use. According to Piaget, the first manifestation of that is the child's ability to bring distant objects within reach. This is achieved by pulling the support on which the objects are standing or by using a stick to push them within the field of prehension. This complex action is formed by the coordination of different schemas for grasping, striking, and bringing to oneself. The first two schemas serve as means to achieving the final goal and are directed by the last schema.

- **Stage VI: Invention of New Means Through Mental Combinations (18 Months)**

Exploration and problem solving in Stage VI continues much in the same way as in stage V with one important difference: the actual experimentation now occurs mentally. The child only pauses to think for a while before solving a problem.

The theoretical approach to robotic tool use, presented in Chapter 3, was inspired by Piaget's ideas. The computational method for learning the affordances of tools presented in Chapter 7 resembles the tertiary circular reactions described by Piaget. Also, Piaget's suggestion that one of the first intelligent acts in children is the use of a stick to bring out-of-reach objects within grasping range has inspired the choice of tools and tool tasks in the robot experiments.

2.3.2 Gibson

James Gibson (1979) was a proponent of the ecological approach to perception which advocates that perception is more a *direct process* than a *cognitive process*. At the time of Gibson's writing, the predominant view in psychology was that objects have properties or qualities associated with them and in order to perceive and object an organism should perceive *all* of its properties. These properties are: color, texture, composition, size, shape, and features of shape, mass, elasticity, rigidity, and mobility among others.

Gibson proposed an alternative theory according to which objects are perceived in terms of their affordances, not their qualities. An *affordance* is defined as an "invariant combination of variables." Presumably it is easier to perceive this invariant than to perceive all the variables separately. In Gibson's own words: "it is never necessary to distinguish all the features of an object and, in fact, it would be impossible to do so; perception is economical" (Gibson, 1979, p. 134-5). The most important property of a chair, for example, is that it affords sitting; its exact shape, color, and material of which it is made are only secondary in importance. Gibson's theory does not suggest that we are not capable of distinguishing object properties if we have to. It only states that when perceiving objects we first observe their affordances and then their properties. According to Gibson, the same process occurs in child development: infants first notice the affordances of objects and only later do they begin to recognize their properties (Gibson, 1979, p. 134).

While Gibson is not specific about the way in which object affordances are learned he seems to suggest that some affordances are learned in infancy when the child experiments with objects. For example, an object affords throwing if it can be grasped and moved away from one's body with a swift action of the hand and then letting it go. The perceptual invariant in this case is the shrinking of the visual angle of the object as it is flying through the air. This highly interesting "zoom" effect will draw the attention of the child (Gibson, 1979, p. 235).

Gibson divides environmental objects into two main categories: attached and detached. *Attached objects* are defined as substances "partially or wholly surrounded by the medium" which cannot be displaced without becoming detached (Gibson, 1979, p. 241). *Detached objects*, on the other hand, are objects that can be displaced; they are portable and afford carrying.

Detached objects must be comparable in size with the animal under consideration in order to afford behavior. For example, an object is graspable if it is approximately "hand size" (Gibson, 1979, p. 234) or has opposable surfaces the distance between which is less than the span of the hand (Gibson, 1979, p. 133). Thus, an object affords different things to people with different body sizes; an object might be graspable for an adult but may not be graspable for a child. Therefore, Gibson suggests that a child learns "his scale of sizes as commensurate with his body, not with a measuring stick" (Gibson, 1979, p. 235).

Using the above definitions, Gibson defines tools as detached objects that are graspable, portable, manipulable, and usually rigid (Gibson, 1979, p. 40). A hammer, for example, is an elongated object that is graspable at one end, weighted at the other end, and affords hitting or hammering. A knife, on the other hand, is a graspable object with a sharp blade that affords cutting. A writing tool like a pencil leaves traces when applied to surfaces and thus affords trace-making (Gibson, 1979, p. 134).

Gibson views tool-using as a rule governed activity. The rules are extensions to the rules used to control our hands and bodies when we perform a similar task without a tool. For example, the "rule for [using] pliers is analogous to that for prehending" (Gibson, 1979, p. 235). This line of reasoning is further extended to suggest that tools can be viewed as extensions to our hands and bodies and that they blur the boundary between us and the environment:

> *"When in use a tool is a sort of extension of the hand, almost an attachment to it or a part of the user's own body, and thus is no longer a part of the environment of the user. But when not in use the tool is simply a detached object of the environment, graspable and portable, to be sure, but nevertheless external to the observer. This capacity to attach something to the body suggests that the boundary between the animal and the environment is not fixed at the surface of the skin but can shift."* (Gibson, 1979, p. 41)

Note that this extension qualifies some everyday objects like clothes as tools. Although clothes are rarely qualified as tools, this extension makes perfect sense if we think of clothes as tools that afford us to maintain our body temperature. When being worn, clothes become part of the wearer's body and are no longer part of the environment. When not being worn, clothes are just detached objects made of fabric (Gibson, 1979, p. 41).

Gibson is not the only one that has postulated that tools can act as extensions of the human body. Neuroscientists have suggested that the *body schema* (i.e., the representation of the body in the brain) is very pliable and can be modified by the use of external objects such as tools (see Section 6.2). Chapter 6 describes a computational model for a robot body schema that has extensibility properties similar to its biological analog.

2.4 AI and Robotics

As it was mentioned in Chapter 1, studies of *autonomous* tool use in robotics and AI are still rare. Nevertheless, multiple researchers have addressed different aspects of the broader problem of robot tool use. The computational aspects of tool use published in the literature can be divided into three main categories: *Tool Modeling*, *Tool Recognition*, and *Tool Application* (Figure 2.6). This section reviews the related work in these areas.

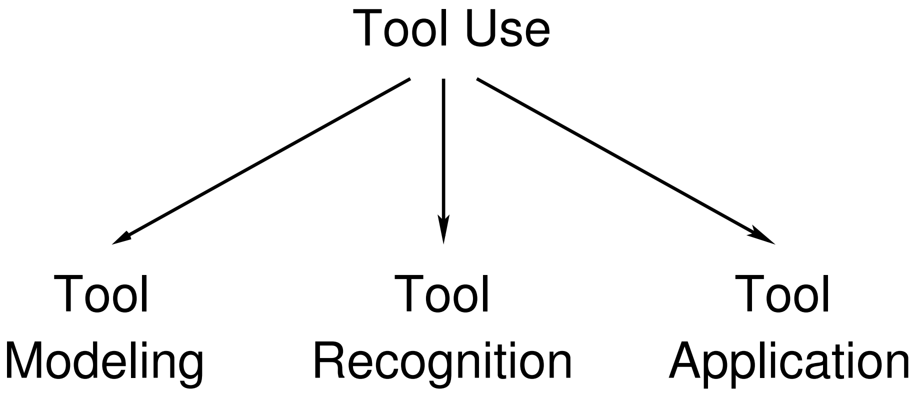


Figure 2.6: The computational aspects of tool use published in the Robotics and AI literature can be divided into three main categories.

The first aspect of tool use found in the literature is *tool modeling*. Tool models are computational representations and data structures that capture the shape and/or functionality of a tool. In many instances tool shape and tool functionality are treated as separate entities and additional data structures are required to perform the mapping between the two. Depending on the assumptions made by different approaches this mapping can be one-to-one, one-to-many, many-to-one, or many-to-many. Shape representation methods can be divided into three types depending on the detail with which they model the boundary or volume of the tool: exact boundary, exact volume, or an approximation of the two. Function representation methods use a library of functional primitives, often described in terms of geometric relationships between object parts, to represent the object. Figure 2.7 shows this classification graphically.

The second component of tool use described in the literature is *tool recognition*. The goal of the recognition process is to identify the type of tool and to extract its functional

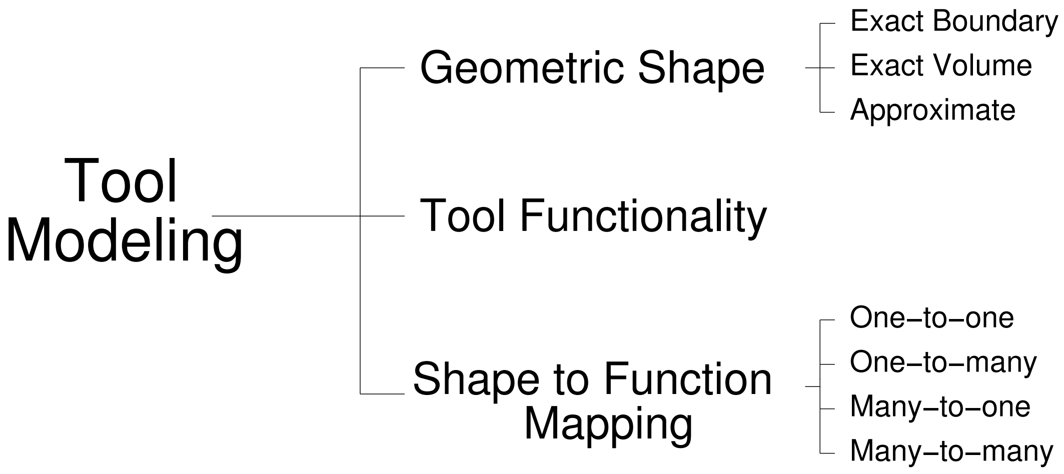


Figure 2.7: Tool models are representations that capture the shape and functionality of a tool. Additional representations and data structures are often necessary to perform the mapping between shape and functionality.

properties so that the tool can be successfully applied to a task. This problem is a subset of the more general problem of object recognition. Therefore, most of the techniques in this category come from the field of computer vision. The algorithms used in the recognition process fall into two main categories: shape-based and function-based. Until recently almost all of the approaches to object recognition were shape-based. More recently, however, function-based approaches have been gaining popularity. This is especially true for recognizing objects that have clearly defined functionality like hand tools. The sensing modality used to perceive the tool for recognition purposes can be visual, haptic, or a combination of the two. This classification is shown on Figure 2.8.

The third component of robotic tool use is *tool application*. Tool application is the act of using the tool to achieve a task. The approaches described in the literature can be classified

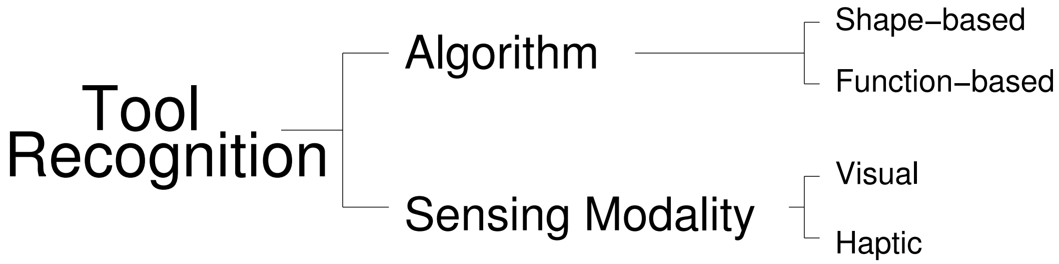


Figure 2.8: Tool recognition approaches can be classified based on the type of the recognition algorithms used and the type of sensing modality used to perceive the tool.

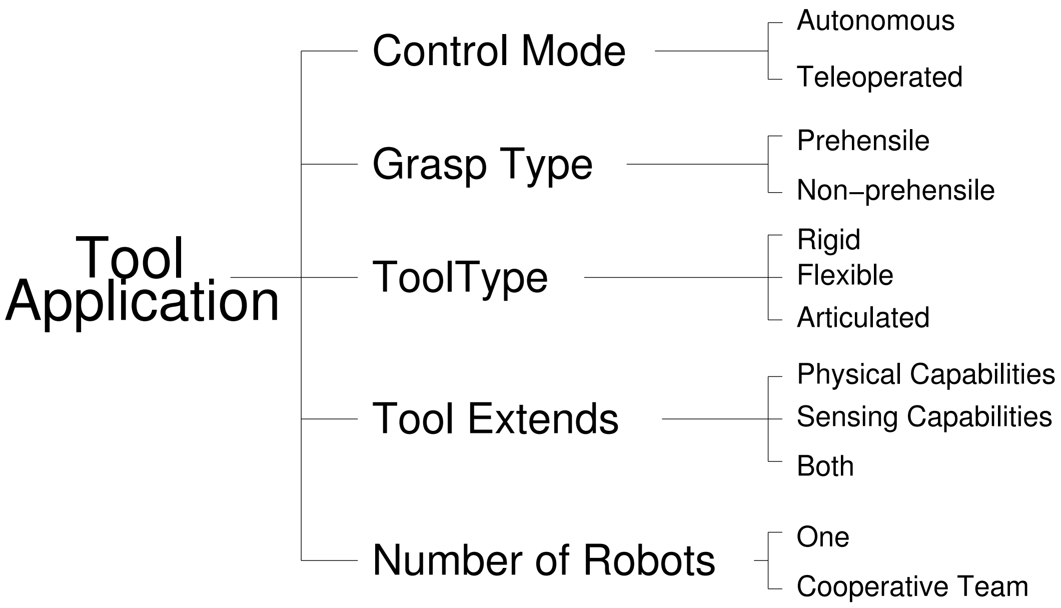


Figure 2.9: Robotic tool application can be classified based on the five criteria shown here.

based on five criteria: control mode, grasp type, tool type, robot capabilities that the tool extends, and number of robots using the tool (Figure 2.9). The control mode can be either autonomous or teleoperated. There are two main types of grasps used to control a tool: prehensile and non-prehensile. The tools used can be rigid objects, flexible objects like ropes, or articulated objects like scissors that have moving parts. A tool usually extends the physical capabilities of a robot like the ability to reach further, however, a tool can also be used to extend the sensing capabilities of the robot (e.g., a stick can be used to hit an object to discover its acoustic properties (Krotkov, 1995)). In most instances tool use is performed by a single robot but the manipulation of large objects sometimes requires a cooperative team of robots.

The following subsections summarize the related work in these three areas.

2.4.1 Tool Modeling

A variety of shape representation techniques for 2D and 3D objects have been developed over the years (Ballard and Brown, 1982; Haralick and Shapiro, 1993). Most of them were invented for computer graphics and computer vision applications, but some have their origins in early AI work. Shape representation methods can be divided into three main categories as described below.

The first category includes methods that describe objects in terms of their boundaries. Some examples include: *polylines, chain codes, B-splines, wire frames,* and *surface-edge-vertex* representations. The methods in the second category focus on representing the area (or volume in the 3D case) of objects. Example methods from this category are: *spatial occupancy arrays, quad-trees*, and *oct-trees.* The methods in the last category represent either the boundary or the volume of the object but use approximation techniques which are more economical and often give better results in object recognition tasks. Some examples are: *skeleton or stick figure* representations, *sticks, plates and blobs* (Shapiro et al., 1984; Mulgaonkar et al., 1984), *generalized cylinders* (Brooks, 1980), *superquadratics* (Barr, 1981), and *superellipses* (Gardiner, 1965).

Finding the mapping between object shape and object functionality is a difficult problem. The nature of this mapping is not always clear since an object usually has more than one functionality. Furthermore, the functionality depends on the intended task. Under some assumptions, however, it is possible to use reasoning and infer the functionality of an object from its shape. The most prominent approaches in AI that have addressed this problem in the context of tool use are reviewed below. Section 2.4.2 describes additional approaches to this problem used in the context of tool recognition.

In an early conceptual work, Lowry (1982) describes a framework for reasoning between structural and functional relationships of objects. In this framework, structure is described with a hierarchy of generalized cylinders and motion sequences of the spines of the generalized cylinders. Function, on the other hand, is expressed by a hierarchy of kinematic primitives, functional primitives, and causal networks. Analogical reasoning and representation constraints are used to link the functional and topological hierarchies describing an object. Both qualitative and quantitative reasoning can be used to extract relationships between the two hierarchies. For example, structural symmetry usually is a good predictor of functional symmetry.

Brady et al. (1985) present an outline for a system intended to assist people with construction and assembly work. The project, named the *mechanic's mate*, has three main

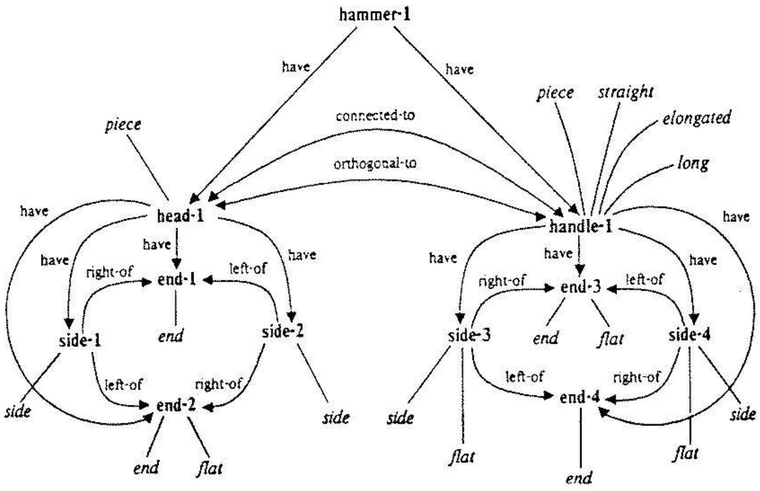


Figure 2.10: Semantic network representation for a hammer. From Connell and Brady (1987).

objectives: 1) to understand the interplay between planning and reasoning in the domain of hand tools and fasteners; 2) to explore different geometric tool representations; and 3) to explore qualitative and quantitative representations that capture the dynamics of using tools, fasteners and objects. Smoothed local symmetries are suggested as shape representation primitives for this system (Brady and Asada, 1984). Case-based planning is suggested as a way of specializing existing tool-plans to new situations.

Connell and Brady (1987) describe a system which performs analogical reasoning about shapes in the domain of hand tools. They use *semantic nets* to describe the structural relations between the object subparts. A hammer, for example, has a head and a handle both of which have two ends and two sides. These descriptions can be learned from positive and negative examples of objects. However, these representations can be complex (see Figure 2.10) and small errors due to poor learning or object recognition may lead to completely different object functionality. To minimize such errors an algorithm based on gray coding was used so that small changes or errors in the symbolic representation cause only small changes in the semantic representation. Viana and Jaulent (1991) describe a different approach to bridging the gap between shape and functionality descriptions. They use fuzzy set representations to compute the compatibility between an object shape and a required function.

The EDISON system (Hodges, 1992, 1993, 1995) uses *naive* mechanical knowledge to reason about the function of devices with moving parts such as a nutcracker and a can opener. A Functional Ontology for Naive Mechanics (FONM) is defined which includes physical, behavioral, and functional properties of objects and devices. Semantic networks are used to describe the shapes of the devices. As discussed above, this representation has the disadvantage of creating a many-to-one mapping from form to function. Hodges, however, sees this as an advantage and claims that this diversity can help with finding innovative uses of the device. Reasoning is used to find functional equivalences between devices or some of their parts. These equivalences can later be exploited in problem solving.

A recent overview of the different approaches to reasoning about the functionality of tools is given by Bicici and St. Amant (2007). The authors conclude that relying only on the shape of an object to determine its functionality is limited. They suggest that "an agent that is interested in learning how a tool can be used [...] needs to look for the changes it can achieve in the physical world by using the tool" (Bicici and St. Amant, 2007, p. 20). The tool representation described in Chapter 7 is consistent with this observation.

2.4.2 Tool Recognition

Many studies in computer vision and AI have focused on object recognition. One specific area of research which is relevant to the current discussion is function-based object recognition. Systems that subscribe to this methodology use information about the shape of an object to reason about the purposes that the object might serve, thus helping the recognition process (DiManzo et al., 1989; Stark and Bowyer, 1991, 1996; Stark et al., 1997; Sutton et al., 1993; Green et al., 1995). These approaches have proven to be useful for recognition of objects with clearly defined functionality like hand tools. This subsection reviews only the approaches that have explicitly dealt with the domain of tools.

One of the most detailed and complete function-based recognition systems to date is GRUFF (Generic Recognition Using Form and Function) (Stark and Bowyer, 1996). The GRUFF project began in 1987 and evolved through four different generations including more object categories with each generation (Stark and Bowyer, 1991; Sutton et al., 1992, 1993, 1994). GRUFF-1 uses function-based definitions of chairs only; GRUFF-2 includes objects in the “furniture” category; GRUFF-3 incorporates “dishes”, “cups”, “bowls”, “plates”, “pans”, and “pitchers”; and GRUFF-4 includes objects in the “hand-tools” category.

GRUFF takes as input a CAD description of an object and outputs a list of possible shape interpretations for this object. A set of knowledge primitives describing basic physical properties are used to reason about the input shape. The primitives take as input portions of object descriptions and output a value between 0 and 1. A global evaluation measure is obtained by combining all primitive measurements. The following knowledge primitives are currently used: *relative orientation, dimensions, proximity, clearance, stability*, and *enclosure.* Each primitive can take several parameters which modify its output. For example, the *relative orientation* primitive takes the normals of two surfaces and compares if the angle between the normals falls within a specified range. This primitive can also be used as a measure of the transformation required to align the two surfaces so that the normals are parallel.

In the GRUFF project object categories are described with definition trees. The root of the tree represents the most general category (e.g., chair). The nodes of the tree represent different sub-categories (e.g., lounge chair, balance chair, high chair). The leaves of the tree represent invocations of the knowledge primitives that describe a specific functional property (e.g., provides arm support, provides back support). When evaluating object functionality GRUFF tries to evaluate if the object can function as an instance of some category instead of looking at its intended purpose (e.g., GRUFF can recognize that a trashcan turned upside down can serve as a chair).

Fine-tuning the object membership functions which rely on many knowledge primitive functions proved to be increasingly more difficult as more objects were added to the CAD database. Because of this difficulty, a learning variant of GRUFF called OMLET was developed (Stark et al., 1997). OMLET learns the membership functions from correctly classified examples of objects.

Rivlin et al. (1995) describe another computer vision system that recognizes simple hand tools. Unlike GRUFF, however, they are not trying to reason about global object

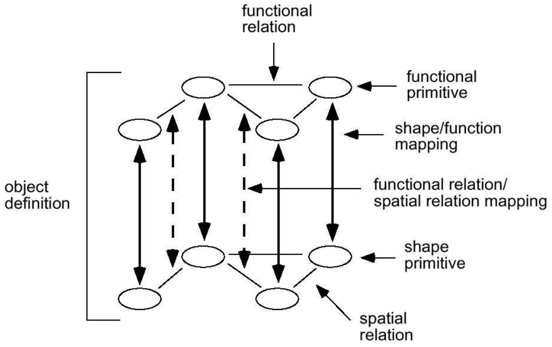


Figure 2.11: Object functionality representation used by Rivlin et al. (1995).

characteristics such as stability, height, or existence of large horizontal surfaces. Instead, they reason about object parts defined qualitatively with an extended version of the *sticks, plates, and blobs* paradigm (Shapiro et al., 1984; Mulgaonkar et al., 1984). The spatial relations between primitives are also expressed quantitatively. Angle of joining, for example, can be *oblique, perpendicular, tangential*, etc. The position of the joint on each surface is also qualitatively described as *near the middle, near the side, near the corner*, or *near the end of a surface.* Functional primitives are defined and linked by functional relations. The mapping between shape primitives (and their relations) and functional primitives is achieved by a one-to-one function (see figure 2.11). In general this mapping falls in the many-to-many category but the authors make the assumption that hand tools have only one functionality. Another assumption in this research is that hand tools have clearly defined *end effectors* (those parts which deliver the action) and *handles* (those parts which provide the interface between the agent and the end-effector).

Kise et al. (1993) also describe a system that recognizes objects in the hand-tools category (e.g., wrenches and bottle openers). They introduce the concept of "functands", which are defined as recipients of direct effects of functional primitives (Kise et al., 1993). The functional model of a tool consists of functands and a variety of functional primitives. A chair, for example, provides back support which is a functional primitive. The recipient of this primitive is a functand which in this case is the back of a human body (Kise et al., 1994). Similarly, a wrench is defined as a "lever working on a nut and a hand". The main goal of this approach is to verify a hypothesized functionality for an object. However, the approach seems computationally complex since it requires testing all possible combinations of functands against all functional primitives. In the case of chair, for example, this requires estimating the degree of support offered by the object for all possible human body postures.

Bogoni and Bajcsy describe a system that evaluates the applicability of differently shaped tools for cutting and piercing operations (Bogoni and Bajcsy, 1995; Bogoni, 1995). A robot manipulator is used to move the tool into contact with various materials (e.g., wood, sponge, plasticine) while a computer vision system tracks the outline of the tool and measures its penetration into the material. The outlines of the tools are modeled by superquadratics and clustering algorithms are used to identify interesting properties of successful tools (see Section 2.4.3 for more details).

All of the methods discussed so far use either visual information derived from a camera or shape information obtained form a CAD model. Several studies, however, have used haptic information as the primary sensing modality (Allen, 1990; Stansfield, 1987). Allen uses a robotic arm and a library of haptic exploratory procedures to derive models of objects that can be used for recognition purposes. For example, contour following hand movements are used to generate a generalized cylinders model of the object, while grasping by containment is used to generate a superquadratics model of the object (Allen, 1990). A hybrid visual-haptic approach to object recognition is described in (Stansfield, 1987).

While these systems may perform well on object recognition tasks, only the work of Bogoni and Bajcsy (1995) has been tested on a real robot platform to perform a tool task. The functional models of tools used by these systems are more suitable for object recognition than for robotic tasks since they do not incorporate the properties of robot-tool and tool-object interaction.

2.4.3 Tool Application

St. Amant and Wood (2005) provide an overview of the literature on physical tool use and argue that creating artificial *habile* agents (i.e., tool-using agents) is an important challenge for robotics and AI. St. Amant and Wood (2005) also introduce the concept of a "tooling test" - a variant of the Turing Test which involves a physical interaction with the world. A robot can pass this test if an independent human observer cannot distinguish whether the tool-using robot is acting on its own or is remotely controlled by another human.

The work of Bogoni and Bajcsy (1995) is the only example, to my knowledge, that has attempted to study object functionality with the intention of using the object as a tool for a robot. In his Ph.D. dissertation Bogoni (1995) uses a robot arm to manipulate a variety of pointed objects in order to determine their applicability for cutting and piercing operations.

The tools used in his work consist of two clearly detectable parts: a handle and an endpoint. The physical tools are three-dimensional objects but their shape, extracted from a camera image, is approximated by a pair of two-dimensional superellipses (one for the handle and one for the endpoint). Each superellipse is defined by four parameters, which control the tapering, width, and length of the shape.

The robot-tool interaction is scripted by a discrete event system (a graph of robot actions and perceptual events). Sensory routines that look for specific events are defined (e.g., the output of a touch sensor is used to to determine when the tool has touched the surface of the object).

The robot experiments evaluate the applicability of differently shaped tools for cutting and piercing tasks. The tools are applied to different types of materials like styrofoam, pine wood, plasticine, and sponge. The outcome of each experiment is either a success or a failure. Clustering algorithms are used to identify the *interesting properties* of the successful tools. Specifically, interesting properties are identified by the parameters of the superellipses for which the difference between the class means ($\mu_{success}$ and $\mu_{failire}$) is the largest relative to the class standard deviation for the given parameter.

Experimental data is used to build *force-shape maps* for each tool. These maps provide a way of classifying a tool with respect to its shape, the force exerted on it, and the depth of penetration for a given material. The maps are represented by interpolated three-dimensional surfaces constructed by fitting second order curves to the empirical data. The three axis of the force-shape map represent force, depth of penetration, and tool sharpness.

A shortcoming of this approach is that both the shape and the functionality of the tool are represented by the same superquadratic (i.e., the mapping between tool shape and tool functionality is one-to-one). For the chopping and piercing tasks, this is not a problem since the only critical functionality of the tool is its sharpness which can be extracted directly from its shape. Another shortcoming is that the limitations and capabilities of the robot are not taken into consideration; the robot manipulator only serves to lower the tool along the vertical direction. In this way, the point of contact between the tool and the object is always predetermined - the edge of the chopper or the tip of the piercer. In more general tool tasks, however, the tool and the object are likely to come into contact at more than one point along their boundaries and the outcome of the task may depend on the point of contact.

Asada and Asari (1988) describe a method for learning manipulator tool tasks from human demonstration. Both the displacement of the tool and the force exerted by the human are stored and used to approximate a robot controller. If $F = (F_1, ..., F_6)^T$ is the force and moment applied to the tool by the human and $x = (x_1, ..., x_6)^T$ is the position and orientation of the tool, the goal is to come up with a robot control law $F = \hat{g}(x, x', x'', ..., x^{(n)})$ that approximates the performance of the human expert. The function $\hat{g}$ must approximate the human data well and yet should be easy to compute to allow real time robot control. This method has been applied to a grinding task in which the tool is moved vertically depending on the surface's unevenness.

Donald, Gariepy, and Rus (1999, 2000) conducted a series of robot experiments in *constrained prehensile manipulation* using ropes. This work is an example of collaborative tool use because a team of robots is used to achieve the task of moving large objects tied with a rope. The tool in this case is a flexible object, which makes the task of the robots more complex. This complexity is reduced by carefully coordinating the actions of the robots and by defining three basic manipulation skills. These skills are: tieing a rope around an object, translating a bound object by pulling it, and rotating a bound object using flossing movements of the rope.

MacKenzie and Arkin (1996) used a behavior-based approach to control a mobile manipulator for a drum sampling task. A sampling instrument is attached to the end-effector of the robot manipulator, which allows the robot to inspect the contents of drums for potentially hazardous materials. Motor schemas are used to control the direction of movement of the sampling instrument. The schemas are controlled by perceptual features of drums and open bung holes extracted from camera images. The pseudo-forces calculated by the motor schemas are applied to the tip of the sampling instrument. The Jacobian of the manipulator is then used to convert these forces into corresponding joint torques (Cameron et al., 1993). In this way the instrument, the arm, and the base move as a cohesive unit.

Krotkov (1995) notes that relatively little robotics research has been geared towards discovering external objects' properties other than shape and position. In particular, Krotkov's research is concerned with robotic identification of material properties of objects. Some of the exploration methods employed by the robot in Krotkov's work use tools coupled with sensory routines to discover object properties. For example, the "whack and watch" method uses a wooden pendulum to strike an object and estimate its mass and coefficient of sliding friction based on the displacement and acceleration of the object after impact. The "hit and listen" method uses a blind person's cane to determine acoustic properties of objects. The cane is dropped from fixed heights on a variety of objects and the sound frequency patterns detected after impact are used to classify the types of materials the objects are made of (Durst and Krotkov, 1993).

Fitzpatrick et al. (2003) used a similar approach to program a robot to poke objects with its arm (without using a tool) and learn the rolling properties of the objects from the resulting displacements. In their work they used a single poking behavior parameterized by four possible starting positions for the robot's arm. The robot learns a model of how each object slides (e.g., toy cars tend to slide in the direction of their elongated axis while balls can slide in any direction).

Tool tasks generally require grasping of the tool. The robotics literature offers numerous examples of robotic grasping of objects. Of particular interest to the current discussion are several studies that have addressed the problem of grasping objects based on their intended functionality. Cutkosky (1989) formulated a grasp taxonomy based on observations of tool grasps used by human mechanics. His taxonomy is based on the type of tool task and relates grip power to object size and dexterity of manipulation. The taxonomy can be used to find suitable grasp choices depending on a task requirements (e.g,. force, mobility, sensitivity) and tool/object attributes (e.g., geometry, texture, fragility). Stansfield (1991) describes a knowledge-based system that uses simple heuristics to choose grasps for initially unknown objects. The system uses a structured light to extract a fixed set of view-dependent object features (Stansfield, 1988). This line of work is similar to the general function-based recognition problem. Kang and Ikeuchi use human input gathered through a data glove to teach a robot different grasping strategies through demonstration (Kang and Ikeuchi, 1995; Kang, 1994).

Several approaches to object manipulation, however, do not use grasping. Examples of such *non-prehensile manipulation* include pushing, throwing, rolling, tumbling, pivoting, snatching, sliding, and slipping (Mason, 1982, 1986; Lynch and Mason, 1999; Lynch, 1998; Zumel and Erdmann, 1996; Aiyama et al., 1993). Other examples include *impulse manipulation* where the robot applies an instantaneous force at a specific location on the object causing the object to move slightly in that direction (Huang et al., 1995); *palm manipulation* in which the entire surface of the robot end manipulator is used as opposed to the use of the fingertips alone (Erdmann, 1996); and *sensorless manipulation* in which the robot indirectly manipulates objects confined in a container or a tray by sliding the objects along walls, into walls, and into corners to reduce the number of possible orientations of the objects (Erdmann and Mason, 1986). These manipulations are very useful in many practical situations and the fact that grasping is not necessary simplifies robot hardware. The hardware simplification, however, comes at the expense of increased complexity of the control and planning algorithms. While similar to tool use this area does not address the problem of using objects as tools. Instead, it focuses on the direct manipulation of objects without using grasping.

There are numerous other examples of robot systems performing tasks, which although not qualified as tool use by their authors, fall into that category based on the definition given in Section 2.1. These tasks include: juggling a tennis ball with a paddle (Aboaf et al., 1989), juggling a devil's stick (Schaal and Atkeson, 1994), using a stick for drumming (Kotosaka and Schaal, 2001), and playing air hockey (Bentivegna and Atkeson, 2001). In general, the solutions to these tasks are tuned to the particular domain by providing detailed kinematic and dynamic models to the robot. In some cases these models are learned from human demonstration which serves as a way of reducing the complexity of the learning task.

Despite the advances in robotics described above there are still a number of domains that require more intelligent tool-using robots. Many problems in these domains are not solvable with existing robot technology. NASA, for example, has demonstrated that there is a palpable need for tool-using robots in high-risk high-cost missions like planetary exploration. This domain requires collecting soil and rock samples using a shovel or tongs. In some cases a hammer must to be used to expose the internal geological makeup of rock samples (Li et al., 1996). The twin NASA rovers, Spirit and Opportunity, that landed on Mars in 2004 carried rock abrasion tools on board. The operation of these tools was controlled remotely from Earth. This may not be feasible, however, for more distant planets.

Another domain for tool-using robots is space station and satellite maintenance. The construction of the International Space Station requires thousands of hours of exhausting and dangerous space walks. Therefore, NASA is developing a humanoid robot that will help with this task (Ambrose et al., 2000). Since all space station components were initially designed to be assembled by humans, the *robonaut* must be capable of handling the tools required for the assembly tasks (Li et al., 1996). The initial goal of NASA, however, is to use the robot in a teleoperated mode.

CHAPTER III

A DEVELOPMENTAL APPROACH TO AUTONOMOUS TOOL USE BY ROBOTS

This chapter formulates five basic principles of developmental robotics and gives an example of how these principles can be applied to the problem of autonomous tool use in robots. The five principles are formulated based on some of the recurring themes in the developmental learning literature and in the author's own research. These principles follow logically from the verification principle (see Section 3.2) which is assumed to be self-evident.

3.1 Introduction

Developmental robotics is one of the newest branches of robotics (Weng et al., 2001; Zlatev and Balkenius, 2001). The basic research assumption of this field is that "true intelligence in natural and (possibly) artificial systems presupposes three crucial properties: *embodiment* of the system, *situatedness* in a physical or social environment, and a prolonged *epigenetic developmental process* through which increasingly more complex cognitive structures emerge in the system as a result of interactions with the physical or social environment" (Zlatev and Balkenius, 2001).

The study of autonomous tool-using abilities in robots is a task that is ideally suited for the methods of developmental robotics. There are three main reasons why. First, tool-using abilities develop relatively early in the life cycles of animals and humans: 1.5-2 years after birth (Piaget, 1952; Tomasello and Call, 1997; Power, 2000). This time is an order of magnitude shorter than the time required for full maturation, which in humans takes approximately 20 years. For robots that learn developmentally this translates to shorter learning times. Second, the ability to use tools precedes the development of other cognitive abilities that are potentially more difficult to model with existing AI techniques (e.g., the development of tool-using abilities in human infants precedes the development of language). And finally, the developmental sequence leading to autonomous tool use is surprisingly uniform across different primate species (Tomasello and Call, 1997; Power, 2000). It follows the developmental sequence outlined by Piaget (see Section 2.3.1). Most of the observed variations between species deal with the duration of the individual stages and not with their order (Tomasello and Call, 1997; Power, 2000). This lack of variation suggests that evolution has stumbled upon a developmental solution to the problem of autonomous tool use which works for many different organisms. This increases the probability that the same sequence may work for robots as well.

Many fields of science are organized around a small set of fundamental laws, e.g., physics has Newton's laws and thermodynamics has its fundamental laws as well. Progress in a field without any fundamental laws tends to be slow and incoherent. Once the fundamental laws are formulated, however, the field can thrive by building upon them. This progress continues until the laws are found to be insufficient to explain the latest experimental evidence. At that point the old laws must be rejected and new laws must be formulated so the scientific progress can continue.

In some fields of science, however, it is not possible to formulate fundamental laws because it would be impossible to prove them, empirically or otherwise. Nevertheless, it is still possible to get around this obstacle by formulating a set of basic principles that are stated in the form of postulates or axioms, i.e., statements that are presented without proof because they are considered to be self-evident. The most famous example of this approach, of course, is Euclid's formulation of the fundamental axioms of Geometry.

Developmental robotics is still in its infancy, however, and it would be premature to try to come up with the fundamental laws or axioms of the field. There are some recurring themes in the developmental learning literature and in the author's own research, however, that can be used to formulate some basic principles. These principles are neither laws (as they cannot be proved at this point) nor axioms (as it would be hard to argue at this point that they are self-evident and/or form a consistent set). Nevertheless, these basic principles can be used to guide future research until they are found to be inadequate and it is time to modify or reject them. Five basic principles are described below.

3.2 The Verification Principle

Developmental Robotics emerged as a field partly as a reaction to the inability of traditional robot architectures to scale up to tasks that require close to human levels of intelligence. One of the primary reasons for scalability problems is that the amount of programming and knowledge engineering that the robot designers have to perform grows very rapidly with the complexity of the robot's tasks. There is mounting evidence that pre-programming cannot be the solution to the scalability problem. The environments in which the robots are expected to operate are simply too complex and unpredictable. It is naive to think that this complexity can be captured in code before the robot is allowed to experience the world through its own sensors and effectors.

Consider the task of programming a household robot, for example, with the ability to handle all possible objects that it can encounter inside a home. It is simply not possible for any robot designer to predict the number of objects that the robot may encounter and the contexts in which they can be used over the robot's projected service time.

There is yet another fundamental problem that pre-programming not only cannot address, but actually makes worse. The problem is that programmers introduce too many hidden assumptions in the robot's code. If the assumptions fail, and they almost always do, the robot begins to act strangely and the programmers are sent back to the drawing board to try and fix what is wrong. The robot has no way of testing and verifying these hidden assumptions because they are not made explicit. Therefore, the robot is not capable of autonomously adapting to situations that violate these assumptions. The only way to overcome this problem is to put the robot in charge of testing and verifying everything that it learns.

After this introduction, the first basic principle can be stated. It is the so-called *verification principle* that was first postulated by Richard Sutton in a series of on-line essays in 2001 (Sutton, 2001a,b). The principle is stated as follows:

> **The Verification Principle:** *An AI system can create and maintain knowledge only to the extent that it can verify that knowledge itself.* (Sutton, 2001b)

According to Sutton, “the key to a successful AI is that it can tell for itself whether or not it is working correctly”(Sutton, 2001b). The only reasonable way to achieve this goal is to put the AI system in charge of its own learning using the verification principle. If verification is not possible for some concept then the AI system should not attempt to learn that concept. In other words, all AI systems and AI learning algorithms should follow the motto: *No verification, No learning.*

Sutton also points out that the verification principle eventually will be adopted by many AI practitioners because it offers fundamental practical advantages over alternative methods when it comes to scalability. Another way of saying the same thing is: “Never program anything bigger than your head”(Sutton, 2001b). Thus, the verification principle stands for autonomous testing and verification performed by the robot and for the robot. As explained above, it would be unrealistic to expect the robot programmers to fix their robots every time when the robots encounter a problem due to a hidden assumption. In fact, it should be the robots telling their programmers what is wrong with them and not the other way around! This point is also mentioned by Dennett (Dennett, 1989) who points out that any sufficiently complicated system, almost by default, must be considered intelligent. Furthermore, when something goes wrong with any sufficiently complex system the people in charge of operating it have no choice other than to accept the system’s own explanation for what is wrong with it.

Sutton was the first researcher in AI to state the verification principle explicitly. However, the origins of the verification principle go back to the ideas of the *logical positivists* philosophers of the 1930’s. The two most prominent among them were Rudolf Carnap and Alfred Ayer. They both argued that statements that cannot be either proved or disproved by experience (i.e., metaphysical statements) are meaningless. Ayer defined two types of verifiability, “strong” and “weak”, which he formulated as follows:

> *“a proposition is said to be verifiable, in the strong sense of the term, if and only if, its truth could be conclusively established in experience. But it is verifiable, in the weak sense, if it is possible for experience to render it probable.”* (Ayer, 1952, p. 37)

Thus, in order to verify something in the "strong" sense one would have to physically perform the verification sequence. On the other hand, to verify something in the "weak" sense one does not have to perform the verification sequence directly but one must have the prerequisite sensors, effectors, and abilities to perform the verification sequence if necessary.

For example, a blind person may be able to verify in the "strong" sense the statement "this object is soft" by physically touching the object and testing its softness. He can also verify this statement in the "weak" sense as he is physically capable of performing the verification procedure if necessary. However, in Ayer's view, the same blind person will not be able to verify, neither in the "strong" nor in the "weak" sense, the statement "this object is red" as he does not have the ability to see and thus to perceive colors. In Ayer's own words:

> *"But there remain a number of significant propositions, concerning matters of fact, which we could not verify even if we chose; simply because we lack the practical means of placing ourselves in the situation where the relevant observations could be made."* (Ayer, 1952, p. 36)

The verification principle is easy to state. However, once a commitment is made to follow this principle the implications are far-reaching. In fact, the principle is so different from the practices of traditional autonomous robotics that it changes almost everything. In particular, it forces the programmer to rethink the ways in which learnable quantities are encoded in the robot architecture as anything that is potentially learnable must also be autonomously verifiable.

The verification principle is so profound that the remaining four principles can be considered as its corollaries. As the connection may not be intuitively obvious, however, they will be stated as separate principles.

3.3 The Principle of Embodiment

An important implication of the verification principle is that the robot must have the ability to verify everything that it learns. Because verification cannot be performed in the absence of actions the robot must have some means of affecting the world, i.e., it must have a body.

The principle of embodiment has been defended many times in the literature, e.g., (Varela et al., 1991; Brooks and Stein, 1994; Brooks et al., 1999; Clark, 1997; Pfeifer and Scheier, 1999; Gibbs, 2006). It seems that at least in robotics there is a consensus that this principle must be followed. After all, there aren't any robots without bodies.

Most of the arguments in favor of the embodiment principle that have been put forward by roboticists, however, are about justifying this principle to its opponents (e.g., Brooks and Stein (1994); Brooks et al. (1999)). The reasons for this are historical. The early AI systems (or as Brooks calls them Good Old Fashioned AI - GOFAI) were disembodied and consisted of learning algorithms that manipulated data in the computer's memory without the need to interact with the external world.

As a result of this historic debate most of the arguments in favor of embodiment miss the main point. The debate should not be about whether or not to embrace the principle of embodiment. Instead, the debate should be about the different ways that can be used to program truly embodied robots. Gibbs makes a similar observation about the current state of the art in AI and robotics: "Despite embracing both embodiment and situatedness in designing enactive robots, most systems fail to capture the way bodily mechanisms are truly embedded in their environments."(Gibbs, 2006, p. 73).

Some of the arguments used to justify the embodiment principle can easily be explained from the point of view of the verification principle. Nevertheless, the connection between the two has not been made explicit so far. Instead of rehashing the debate in favor of embodiment – which has been argued very eloquently by others, e.g., (Varela et al., 1991; Gibbs, 2006) – I am only going to focus on a slightly different interpretation of embodiment in light of the verification principle.

In my opinion, most arguments in favor of the embodiment principle make a distinction between the body and the world and treat the body as something special. In other words, they make the body/world boundary explicit. This distinction, however, is artificial. The only reason why the body may seem special is because the body is the most consistent, the most predictable, and the most verifiable part of the environment. Other that that, there should be no difference between the body and the external world. To the brain the body may seem special but that is just because “the brain is the body’s captive audience” (Demasio, 1994, p. 160). In other words, the body is always there and we can’t run away from it.

According to the new interpretation of the embodiment principle described here, the body is still required for the sake of verification. However, the verification principle must also be applicable to the properties of the body. That is to say, the properties of the body must be autonomously verifiable as well. Therefore, the learning and exploration principles that the robot uses to explore the external world must be the same as the ones that it uses to explore the properties of its own body.

This interpretation reduces the special status of the body. Instead of treating the body as something special, the new interpretation treats the body as simply the most consistent, the most predictable, and the most verifiable part of the environment. Because of that the body can be easily distinguished from the environment. Furthermore, in any developmental trajectory the body must be explored first.

In fact, distinguishing the body from the external world should be relatively easy because there are certain events that only the owner of the body can experience and no one else. Rochat (2003) calls these events *self-specifying* and lists three such events: 1) efferent-afferent loops (e.g., moving ones hand and seeing it move); 2) double touch (e.g., touching one’s two index fingers together); 3) vocalization behaviors followed by hearing their results (e.g., crying and hearing oneself cry). These events are characterized by the fact that they are multimodal, i.e., they involve more than one sensory or motor modality. Also, these events are autonomously verifiable because you can always repeat the action and observe the same result.

Because the body is constructed from actual verifiable experience, in theory, it should be possible to change one’s body representation. In fact, it turns out that this is surprisingly easy to do. Some experiments have shown that the body/world boundary is very pliable and can be altered in a matter of seconds (Ramachandran and Blakeslee, 1998; Iriki et al., 1996). For example, it comes as a total surprise for many people to realize that what they normally think of as their own body is just a phantom created by their brains. There is a very simple experiment which can be performed without any special equipment that exposes the phantom body (Ramachandran and Blakeslee, 1998). The experiment goes like this: a subject places his arm under a table. The person conducting the experiment sits right next to the subject and uses both of his hands to deliver simultaneous taps and strokes to both the subject's arm (which is under the table) and the surface of the table. If the taps and strokes are delivered synchronously then after about 2 minutes the subject will have the bizarre sensation that the table is part of his body and that part of his skin is stretched out to lie on the surface of the table. Similar extensions and re-mappings of the body have been reported by others (Botvinick and Cohen, 1998; Iriki et al., 1996; Ishibashi et al., 2000).

The conclusions from these studies may seem strange because typically one would assume that embodiment implies that there is a solid representation of the body somewhere in the brain. One possible reason for the phantom body is that the body itself is not constant but changes over time. Our bodies change with age. They change as we gain or lose weight. They change when we suffer the results of injuries or accidents. In short, our bodies are constantly changing. Thus, it seems impossible that the brain should keep a fixed representation for the body. If this representation is not flexible then sooner or later it will become obsolete and useless.

Another possible reason for the phantom body is that it may be impossible for the brain to predict all complicated events that occur within the body. Therefore, the composition of the body must be constructed continuously from the latest available information. This is eloquently stated by Demasio:

> *"Moreover, the brain is not likely to predict how all the commands - neural and chemical, but especially the latter- will play out in the body, because the play-out and the resulting states depend on local biochemical contexts and on numerous variables within the body itself which are not fully represented neurally. What is played out in the body is constructed anew, moment by moment, and is not an exact replica of anything that happened before. I suspect that the body states are not algorithmically predictable by the brain, but rather that the brain waits for the body to report what actually has transpired."* (Demasio, 1994, p. 158)

Chapter 5 uses the insights from this section to formulate an algorithm for autonomous self-detection by a robot. The algorithm uses proprioceptive-visual efferent-afferent loops as self-specifying events to identify which visual features belong to the robot's body.

Chapter 6 describes a computational representation for a robot body schema (RBS). This representation is learned by the robot from self-observation data. The RBS representation meets the requirements of both the verification principle and the embodiment principle as the robot builds a model for its own body from self-observation data that is repeatably observable.

3.4 The Principle of Subjectivity

The principle of subjectivity also follows quite naturally from the verification principle. If a robot is allowed to learn and maintain only knowledge that it can autonomously verify for itself then it follows that what the robot learns must be a function of what the robot has experienced through its own sensors and effectors, i.e., its learning must be a function of experience. As a consequence, two robots with the same control architectures but with different histories of interactions could have two totally different representations for the same object. In other words, the two representations will be subjective.

Ayer was probably the first one to recognize that the verification principle implies subjectivity. He observed that if all knowledge must be verifiable through experience then it follows that all knowledge is subjective as it has to be formed through individual experiences (Ayer, 1952, p. 125-126). Thus, what is learned depends entirely on the capabilities of the learner and the history of interactions between the learner and the environment or between the learner and its own body. Furthermore, if the learner does not have the capacity to perform a specific verification procedure then the learner would never be able to learn something that depends on that procedure (as in the blind person example given above). Thus, subjectivity may be for developmental learning what relativity is for physics – a fundamental limitation that cannot be avoided or circumvented.

The subjectivity principle captures very well the subjective nature of object affordances. A similar notion was suggested by Gibson who stated that a child learns "his scale of sizes as commensurate with his body, not with a measuring stick" (Gibson, 1979, p. 235). Thus, an object affords different things to people with different body sizes; an object might be graspable for an adult but may not be graspable for a child. Noë has recently given a modern interpretation of Gibson's ideas and has stressed that affordances are also skill relative:

> *"Affordances are animal-relative, depending, for example, on the size and shape of the animal. It is worth noting that they are also skill-relative. To give an example, a good hitter in baseball is someone for whom a thrown pitch affords certain possibilities for movement. The excellence of a hitter does not consist primarily in having excellent vision. But it may very well consist in the mastery of sensorimotor skills, the possession of which enables a situation to afford an opportunity for action not otherwise available."* (Noë, 2004, p. 106)

From what has been said so far one can infer that the essence of the principle of subjectivity is that it imposes limitations on what it potentially learnable by a specific agent. In particular, there are two types of limitations: sensorimotor and experiential. Each of them is discussed below along with the adaptation mechanisms that have been adopted by animals and humans to reduce the impact of these limitations.

3.4.1 Sensorimotor Limitations

The first limitation imposed on the robot by the subjectivity principle is that what is potentially learnable is determined by the sensorimotor capabilities of the robot's body. In other words, the subjectivity principle implies that all learning is pre-conditioned on what the body is capable of doing. For example, a blind robot cannot learn what is the meaning of the color red because it does not have the ability to perceive colors.

While it may be impossible to learn something that is beyond the sensorimotor limitations of the body, it is certainly possible to push these limits farther by building tools and instruments. It seems that a common theme in the history of human technological progress is the constant augmentation and extension of the existing capabilities of our bodies. For example, Campbell outlines several technological milestones which have essentially pushed one body limit after another (Campbell, 1985). The technological progression described by Campbell starts with tools that augment our physical abilities (e.g., sticks, stone axes, and spears), then moves to tools and instruments that augment our perceptual abilities (e.g., telescopes and microscopes), and it is currently at the stage of tools that augment our cognitive abilities (e.g., computers and PDAs).

Regardless of how complicated these tools and instruments are, however, their capabilities will always be learned, conceptualized, and understood *relative to our own sensorimotor capabilities*. In other words, the tools and instruments are nothing more than prosthetic devices that can only be used if they are somehow tied to the pre-existing capabilities of our bodies. Furthermore, this tool-body connection can only be established through the verification principle. The only way in which we can understand how a new tool works is by expressing its functionality in terms of our own sensorimotor repertoire. This is true even for tools and instruments that substitute one sensing modality for another. For example, humans have no natural means of reading magnetic fields but we have invented the compass which allows us to do that. The compass, however, does not convert the direction of the magnetic field into a modality that we can't interpret, e.g., infrared light. Instead, it converts it to human readable form with the help of a needle.

The exploration process involved in learning the functional properties or affordances of a new tool is not always straight forward. Typically this process involves active trial and error. Probably the most interesting aspect of this exploration, however, is that the functional properties of the new tool are learned in relation to the existing behavioral repertoire of the learner.

The related work on animal object exploration indicates that animals use stereotyped exploratory behaviors when faced with a new object (Power, 2000; Lorenz, 1996). This set of behaviors is species specific and may be genetically predetermined. For some species of animals these tests include almost their entire behavioral repertoire: "A young corvide bird, confronted with an object it has never seen, runs through practically all of its behavioral patterns, except social and sexual ones."(Lorenz, 1996, p. 44)

Unlike crows, adult humans rarely explore a new object by subjecting it to all possible behaviors in their behavioral repertoire. Human object exploration tends to be more focused although that is not always the case with human infants (Power, 2000). Nevertheless, an extensive exploration process similar to the one displayed by crows can sometimes be observed in adult humans as well. This process is easily observed in the members of technologically "primitive" societies when they are exposed for the first time to an object from a technologically advanced society (Diamond, 1999, p. 246).

Chapter 7 describes a method for autonomous learning of object affordances by a robot. The robot learns the affordances of different tools in terms of the expected outcomes of specific exploratory behaviors. The affordance representation is inherently subjective as it is expressed in terms of the behavioral repertoire of the robot (i.e., it is skill relative). The affordance representation is also subjective because the affordances are expressed relative to the capabilities of the robot's body. For example, if an object is too thick to be grasped by the robot the robot learns that the object is not graspable even though it might be graspable for a different robot with a larger gripper (Stoytchev, 2005).

3.4.2 Experiential Limitations

In addition to sensorimotor limitations the subjectivity principle also imposes experiential limitations on the robot. Experiential limitations restrict what is potentially learnable simply because learning depends on the history of interactions between the robot and the environment, i.e., it depends on experience. Because, among other things, experience is a function of time this limitation is essentially due to the finite amount of time that is available for any type of learning. One interesting corollary of this is that: the more intelligent the life form the longer it has to spend in the developmental stage.

Time is a key factor in developmental learning. By default developmental learning requires interaction with the external world. There is a limit on how fast this interaction can occur which ultimately restricts the speed of learning. While the limitation of time cannot be avoided it is possible to speed up learning by relying on the experience of others. The reason why this does not violate the subjectivity principle is because verification can be performed in the "weak sense" and not only in the "strong sense." Humans, for example, often exploit this shortcut. Ever since writing was invented we have been able to experience places and events through the words and pictures of others. These vicarious experiences are essential for us.

Vicarious experiences, however, require some sort of basic overlap between our understanding of the world and that of others. Thus, the following question arises: if everything that is learned is subjective then how can two different people have a common understanding about anything? Obviously this is not a big issue for humans because otherwise our civilization will not be able to function normally. Nevertheless, this is one of the fundamental questions that many philosophers have grappled with.

To answer this question without violating the basic principles that have been stated so far we must allow for the fact that the representations that two agents have may be functionally different but nevertheless they can be qualitatively the same. Furthermore, the verification principle can be used to establish the qualitative equivalence between the representations of two different agents. This was well understood by Ayer who stated the following:

> *"For we define the qualitative identity and difference of two people's sense-experiences in terms of the similarity and dissimilarity of their reactions to empirical tests. To determine, for instance, whether two people have the same colour sense we observe whether they classify all the colour expanses with which they are confronted in the same way; and when we say that a man is color-blind, what we are asserting is that he classifies certain colour expanses in a different way from that in which they would be classified by the majority of people."* (Ayer, 1952, p. 132)

Another reason why two humans can understand each other even though they have totally different life experiences is because they have very similar physical bodies. While no two human bodies are exactly the same they still have very similar structure. Furthermore, our bodies have limits which determine how we can explore the world through them (e.g., we can only move our hands so fast). On the other hand, the world is also structured and imposes restrictions on how we can explore it through our actions (e.g., an object that is too wide may not be graspable). Because we have similar bodies and because we live in the same physical world there is a significant overlap which allows us to have a shared understanding. Similar ideas have been proposed in psychology and have been gaining popularity in recent years (Glenberg, 1997; O'Regan and Noë, 2001; Noë, 2004; Gibbs, 2006).

Consequently, experience must constantly shape or change all internal representations of the agent over time. Whatever representations are used they must be flexible enough to be able to change and adapt when new experience becomes available. There is a good amount of experimental evidence to suggest that such adaptation takes place in biological systems. For example, the representation of the fingers in the somatosensory cortex of a monkey depends on the pattern of their use (Wang et al., 1995). If two of the fingers are used more often than other fingers then the number of neurons in the somatosensory cortex that are used to encode these two fingers will increase (Wang et al., 1995).

The affordance representation described in Chapter 7 is influenced by the actual history of interactions between the robot and the tools. The affordance representation is pliable and can accommodate the latest empirical evidence about the properties of the tool. For example, the representation can accommodate tools that can break – a drastic change that significantly alters their affordances.

3.5 The Principle of Grounding

While the verification principle states that all things that the robot learns must be verifiable, the grounding principle describes what constitutes a valid verification. Grounding is very important because if the verification principle is left unchecked it can easily go into an infinite recursion. At some point there needs to be an indivisible entity which is not brought under further scrutiny, i.e., an entity which does not require additional verification. Thus, figuratively speaking, grounding puts the brakes on verification.

Grounding is a familiar problem in AI. In fact, one of the oldest open problems in AI is the so-called *symbol grounding problem* (Harnad, 1990). Grounding, however, is also a very loaded term. Unfortunately, it is difficult to come up with another term to replace it with. Therefore, for the purposes of this document the term grounding is used only to refer to the process or the outcome of the process which determines what constitutes a successful verification.

Despite the challenges in defining what constitutes grounding, if we follow the principles outlined so far we can arrive at the basic components of grounding. The motivation for stating the embodiment principle was that verification is impossible without the ability to affect the world. This implies that the first component that is necessary for successful verification (i.e., grounding) is an action or a behavior.

The action by itself, however, is not very useful for the purposes of successful verification (i.e., grounding) because it does not provide any sort of feedback. In order to verify anything the robot needs to be able to observe the outcomes of its own actions. Thus, the second component of any verification procedure must be the outcome or outcomes that are associated with the action that was performed.

This leads us to one of the main insights of this section, namely, that grounding consists of ACT-OUTCOME (or BEHAVIOR-OBSERVATION) pairs. In other words, grounding is achieved through the coupling of actions and their observable outcomes. Piaget expressed this idea when he said that "children are real explorers" and that "they perform experiments in order to see." Similar ideas have been proposed and defended by others, e.g., (Gibson, 1979, 1969; Varela et al., 1991; Noë, 2004; O'Regan and Noë, 2001; Gibbs, 2006).

Grounding of information based on a single act-outcome pair is not sufficient, however, as the outcome may be due to a lucky coincidence. Thus, before grounding can occur the outcome must be replicated at least several times in the same context. If the act-outcome pair can be replicated then the robot can build up probabilistic confidence that what was observed was not just due to pure coincidence but that there is a real relationship that can be reliably reproduced in the future.

Therefore, grounding requires that action-outcome pairs be coupled with some sort of probabilistic estimates of repeatability. Confidence can be built up over time if multiple executions of the same action lead to the same outcome under similar conditions. In many situations the robot should be able to repeat the action (or sequence of actions) that were executed just prior to the detection of the outcome. If the outcome can be replicated then the act-outcome pair is worth remembering as it is autonomously verifiable. Another way to achieve the same goal is to remember only long sequences of (possibly different) act-outcome pairs which are unlikely to occur in any other context due to the length of the sequence. This latter method is closer to Gibson's ideas for representing affordances.

Stating that grounding is performed in terms of act-outcome pairs coupled with a probabilistic estimate is a good start but leaves the formulation of grounding somewhat vague. Each action or behavior is possibly itself a very complicated process that involves multiple levels of detail. The same is true for the outcomes or observations. Thus, what remains to be addressed is how to identify the persistent features of a verification sequence that are constant across different contexts. In other words, one needs to identify the sensorimotor invariants. Because the invariants remain unchanged they are worth remembering and thus can be used for grounding.

While there could be potentially infinite number of ways to ground some information this section will focus on only one of them. It is arguably the easiest one to pick out from the sensorimotor flux and probably the first one to be discovered developmentally. This mechanism for grounding is based on detection of temporal contingency.

Temporal contingency is a very appealing method for grounding because it abstracts away the nature and complexity of the stimuli involved and reduces them to the relative time of their co-occurrence. The signals could come from different parts of the body and can have their origins in different sensors and actuators.

Temporal contingency is easy to calculate. The only requirement is to have a mechanism for reliable detection of the interval between two events. The events can be represented as binary and the detection can be performed only at the times in which these signals change from 0 to 1 or from 1 to 0. Furthermore, once the delay between two signals is estimated it can be used to predict future events.

Timing contingency detection is used in Chapter 5 to detect which perceptual features belong to the body of the robot. In order to do that, the robot learns the characteristic delay between its motor actions (efferent stimuli) and the movements of perceptual features in the environment (afferent stimuli). This delay can then be used to classify the perceptual stimuli that the robot can detect into "self" and "other".

Detection of temporal contingency is very important for the normal development of social skills as well. In fact, it has often been suggested that contingency alone is a powerful social signal that plays an important role in learning to imitate (Jones, 2006) and language acquisition (Goldstein et al., 2003). Watson (Watson, 1985) proposed that the "*contingency relation* between a behavior and a subsequent stimulus may serve as a social signal beyond (possibly even independent of) the signal value of the stimulus itself." Exploring this suggestion might be a fruitful direction for future work in social robotics.

3.6 The Principle of Gradual Exploration

The principle of gradual exploration recognizes the fact that it is impossible to learn everything at the same time. Before we learn to walk we must learn how to crawl. Before we learn to read we must learn to recognize individual letters. There is no way around that. Similarly, there are certain milestones or stages that must be achieved in developmental learning before development can continue to the next stage.

Every major developmental theory either assumes or explicitly states that development proceeds in stages (Piaget, 1952; Freud, 1965; Bowlby, 1969). These theories, however, often disagree about what causes the stages and what triggers the transitions between them. Variations in the timing of these stages have also been observed between the members of the same species. Therefore, the age limits set by Piaget and others about what developmental milestone should happen when must be treated as rough guidelines and not as fixed rules. "Although the stages correspond roughly to age levels (at least in the children studied [by Piaget]), their significance is not that they offer behavior norms for specific ages but that the sequence of the stages and the transition from one stage to the next is invariant."(Wolff, 1960, p. 37)

E.J. Gibson (who was J.J. Gibson's wife) also expressed some doubts about the usefulness of formulating stages in developmental learning: *"I want to look for trends in development, but I am very dubious about stages. [...] To repeat, trends do not imply stages in each of which a radically new process emerges, nor do they imply maturation in which a new direction exclusive of learning is created."*(Gibson, 1969, p. 450)

It seems that a more fruitful area of research these days is to compare and contrast the developmental sequences of different organisms. Comparative studies between primates and humans are useful precisely because they expose the major developmental differences between different species that follow Piaget's sequence in their development (Tomasello and Call, 1997; Power, 2000). Regardless of what causes the stages, one of the most important lessons that we can draw from these studies is that the final outcome depends not just on the stages but on their relative order and duration.

For example, the time during which autonomous locomotion emerges after birth in primates varies significantly between different species (Tomasello and Call, 1997; Power, 2000). In chimpanzees this is achieved fairly rapidly and then they begin to move about the environment on their own. In humans, on the other hand, independent locomotion does not emerge until about a year after birth. An important consequence of this is that human infants have a much longer developmental period during which they can manually explore and manipulate objects. They tend to play with objects, rotate them, chew them, throw them, relate them to one another, and bring them to their eyes to take a closer look. In contrast, chimpanzees are not as interested in sitting down and manually exploring objects because they learn to walk at a much younger age. To the extent that object exploration occurs in chimpanzees, it usually is performed when the objects are on the ground (Tomasello and Call, 1997; Power, 2000). Chimpanzees rarely pick up an object in order to bring it to the eyes and explore it (Power, 2000).

Another interesting result from comparative studies is that object exploration (and exploration in general) seems to be self-guided and does not require external reinforcement. What is not yet clear, however, is what process initiates exploration and what process terminates it.

The principle of *gradual exploration* states that exploration is self-regulated and always proceeds from the most verifiable to the least verifiable parts of the environment. In other words, the exploration is guided by an attention mechanism that is continually attracted to parts of the environment that exhibit medium levels of verifiability. Therefore, the exploration process can chart a developmental trajectory without external reinforcement because what is worth exploring next depends on what is being explored now.

The previous section described how temporal contingency can be used for successful verifiability (i.e., grounding). This section builds upon that example but also takes into account the level of contingency that is detected. At any point in time the parts of the environment that are the most interesting, and thus worth exploring, exhibit medium levels of contingency. To see why this might be the case consider the following insights reached by Watson (1985, 1994).

In his experiments Watson studied the attentional mechanisms of infants and how they change over time. In Watson's terminology the delay between motor commands (efferent signals) and observed movements of visual stimuli (afferent signals) is called *the perfect contingency*. Furthermore, Watson defines several levels of contingency based on how much the delay deviates from the delay associated with the perfect contingency. Watson (1985) observed that the level of contingency that is detected by the infants is very important. For example he observed that three-month-old infants paid attention only to stimuli that exhibit the perfect contingency (Watson, 1994) while 16-week-old infants paid attention to stimuli that exhibit imperfect contingencies (Watson, 1985). In his experiments the infants watched a TV monitor which showed a woman's face. The TV image was manipulated such that the woman's face would become animated for 2-second intervals after the infant kicked with his legs (Watson, 1985). The level of this contingency was varied by adjusting the timing delay between the infants' kicking movements and the animation. Somewhat surprisingly the 16-week-old infants in this study paid more attention to faces that did not show the perfect contingency (i.e., faces that did not move immediately after the infants' kicking movements). This result led Watson to conclude that the infant's attentional mechanisms may be modulated by an inverted U-shaped function based on the contingency of the stimulus (Watson, 1985).

An attention function that has these properties seems ideal for an autonomous robot. If a stimulus exhibits perfect contingency then it is not very interesting as the robot can already predict everything about that stimulus. On the other hand, if the stimulus exhibits very low levels of contingency then the robot cannot learn a predictive model of that stimulus which makes that stimulus uninteresting as well. Therefore, the really interesting stimuli are those that exhibit medium levels of contingency.

E.J. Gibson reached conclusions similar to those of Watson. She argued that perceptual systems are self-organized in such a way that they always try to reduce uncertainty. Furthermore, this search is self-regulated and does not require external reinforcement:

> *"The search is directed by the task and by intrinsic cognitive motives. The need*

> *to get information from the environment is as strong as to get food from it, and obviously useful for survival. The search is terminated not by externally provided rewards and punishments, but by internal reduction of uncertainty. The products of the search have the property of reducing the information to be processed. Perception is thus active, adaptive, and self-regulated."* (Gibson, 1969, p. 144)

Thus, the main message of this section is that roboticists should try to identify attention functions and intrinsic motivation functions for autonomous robots that have properties similar to the ones described above. Namely, functions that can guide the robot's gradual exploration of the environment from its the most verifiable to its least verifiable parts. This seems to be a promising area of future research.

3.7 Developmental Sequence for Autonomous Tool Use

This section provides an example that uses the five principles described above in a developmental sequence. This sequence can be used by autonomous robots to acquire tool using abilities. Following this sequence, a robot can explore progressively larger chunks of the initially unknown environment that surrounds it. Gradual exploration is achieved by detecting regularities that can be explained and replicated with the sensorimotor repertoire of the robot. This exploration proceeds from the most predictable to the least predictable parts of the environment.

The developmental sequence begins with learning a model of the robot's body since the body is the most consistent and predictable part of the environment. Internal models that reliably identify the sensorimotor contingencies associated with the robot's body are learned from self-observation data. For example, the robot can learn the characteristic delay between its motor actions (efferent stimuli) and the movements of perceptual features in the environment (afferent stimuli). By selecting the most consistently observed delay the robot can learn its own efferent-afferent delay. Furthermore, this delay can be used to classify the perceptual stimuli that the robot can detect into "self" and "other" (see Chapter 5).

Once the perceptual features associated with the robot's body are identified, the robot can begin to learn certain patterns exhibited by the body itself. For example, the features that belong to the body can be clustered into groups based on their movement contingencies.

These groups can then be used to form frames of reference (or body frames) which in turn can can be used to both control the movements of the robot as well as to predict the locations of certain stimuli (see Chapter 6).

During the next stage, the robot uses its body as a well defined reference frame from which the movements and positions of environmental objects can be observed. In particular, the robot can learn that certain behaviors (e.g., grasping) can reliably cause an environmental object to move in the same way as some part of the robot's body (e.g., its wrist) during subsequent robot behaviors. Thus, the robot can learn that the grasping behavior is necessary in order to control the position of the object reliably. This knowledge is used for subsequent tool-using behaviors. One method for learning these first-order (or binding) affordances is described in (Stoytchev, 2005).

Next the robot can use the previously explored properties of objects and relate them to other objects. In this way, the robot can learn that certain actions with objects can affect other objects, i.e., they can be used as tools. Using the principles of verification and grounding the robot can learn the affordances of tools. The robot can autonomously verify and correct these affordances if the tool changes or breaks (see Chapter 7).

3.8 Summary

This chapter proposed five basic principles of developmental robotics. These principles were formulated based on some of the recurring themes in the developmental learning literature and in the author's own research. The five principles follow logically from the verification principle (postulated by Richard Sutton) which is assumed to be self-evident.

The chapter also described an example of how these principles can be applied to autonomous tool use in robots. The chapters that follow describes the individual components of this sequence in more details.

CHAPTER IV

EVALUATION PLATFORMS

This chapter describes the evaluation platforms that were chosen for performing the experiments described in the following chapters. The two platforms, a dynamics robot simulator and a mobile robot manipulator, are described in detail below.

4.1 Dynamics Robot Simulator

The first experimental platform is a physics-based 3D dynamics robot simulator. The simulator was implemented in C++ and was used as a fast prototyping environment for verification and testing of the approach to autonomous robotic tool use. The simulator provides a rich set of routines for modeling, simulating, controlling, and drawing of simulated worlds, robots, and environmental objects. Figure 4.1 shows screen snapshots from several microworlds and robots that were used during different stages of this research.

Simulated robots are modeled as a collection of rigid bodies. The shape of each rigid body is described by one or more primitive geometric shapes. The primitive geometric shapes supported by the simulator are: parallelopiped, sphere, and capped-cylinder (a cylinder with two hemispheres at each end). Two rigid bodies can be connected with a joint which imposes constraints on the relative movements of the bodies. The following joints are currently supported: hinge, prismatic, ball and socket, and universal. The configuration of the robots is specified in plain-text configuration files. This feature allows for rapid prototyping of new robots and making changes to existing robots. The same process is used to build the objects and tools in the simulated worlds.

The geometric shapes of the rigid bodies are used to calculate the collision points between them which in turn are used to calculate the corresponding collision forces. The collision forces and other forces such as gravity, friction, and robot motor torques are included in the dynamics calculations to produce physically accurate movements of the simulated objects. To achieve this, the simulator computes the positions, rotations, accelerations, and torques for all rigid bodies at regular time intervals. The timestep for these calculations can be specified as a parameter which allows for more and more accurate simulations at the expense of computational time.

The dynamics calculations are performed using the Open Dynamics Engine (ODE) Library (Smith, 2003). ODE provides basic capabilities for dynamically simulating 3D rigid objects and calculating the collisions between them. However, it does not provide functionality for building and controlling robots. ODE also includes an OpenGL-based visualization library which was extended for the purposes of the simulator.

At least three other robot simulators are based on ODE: Gazebo, Webots, and UberSim. Gazebo is a 3D multiple robots simulator with dynamics developed at USC (the code is available at http://playerstage.sourceforge.net/gazebo/gazebo.html). It is geared toward mobile non-articulated robots and thus has limited robot modeling capabilities. Webots (Michael, 2004) offers robot building capabilities but it is a commercial product and users are limited by what they can modify without access to the source code. UberSim (Go et al., 2004) is an open source simulator geared toward robot soccer applications. It was developed at Carnegie Mellon and is similar to the other two simulators.

Another feature of the simulator used in this research (that is also present in the aforementioned simulators) is that computer vision routines are integrated into the simulator. The OpenGL rendered simulator window is treated as a camera image on which computer vision processing can be performed. The simulator supports multiple cameras which can be static or attached to a robot. The open source color segmentation library CMVision (Bruce et al., 2000) was used to track the movements of the robots, tools, and objects in the simulated microworlds. The objects and locations of interest were color coded to accommodate the tracking process.

The combination of physically realistic 3D dynamics simulation and computer vision routines from simulated images provides for a level of realism that could not be achieved with the previous generation of autonomous robot simulators (e.g., MissionLab (MacKenzie et al., 1997), TeamBots (Balch, 1998), Stage (Gerkey et al., 2003)). Despite these advances in simulation technology, however, a robot simulator is still only a *simulator*. A common problem with all simulators is that they fail to capture some of the complexities of the real world that they try to simulate. Thus, the simulation environment may provide a false sense of success. Therefore, it is necessary to validate the simulation results by repeating some (or all) experiments on a real robot. The next section describes the robot that was used for this purpose.

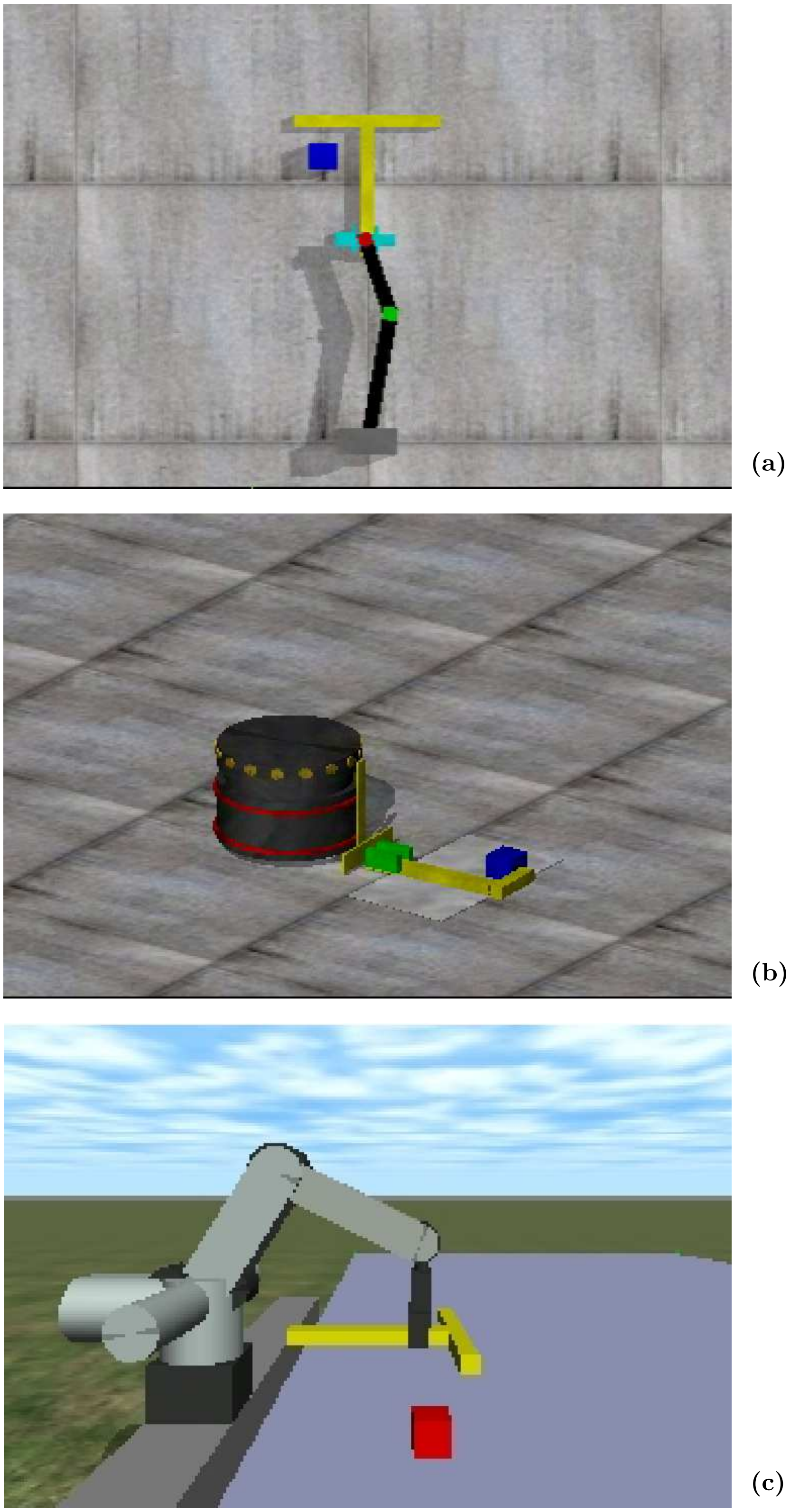


Figure 4.1: Screen snapshots from simulated microworlds and robots used at different stages of this research. a) A two joint robot with a gripper; b) A Nomad 150 robot; c) The simulation version of the CRS+ A251 mobile manipulator described in the next section.

4.2 Mobile Robot Manipulator

The second experimental platform is a CRS+ A251 manipulator arm. The robot has 5 degrees of freedom (waist roll, shoulder pitch, elbow pitch, wrist pitch, wrist roll) plus a gripper (Figure 4.2). The joint limits of the manipulator are shown in Figure 4.3. In addition to that, the arm was mounted on a Nomad 150 robot which allows the manipulator to move sideways (Figure 4.4). The Nomad 150 is a three-wheeled holonomic vehicle with separately steerable turret and base. In all experiments, however, its movements were restricted to simple translations parallel to the table on which the tools and attractors were placed. In other words, the Nomad 150 was used as a linear track for the manipulator.

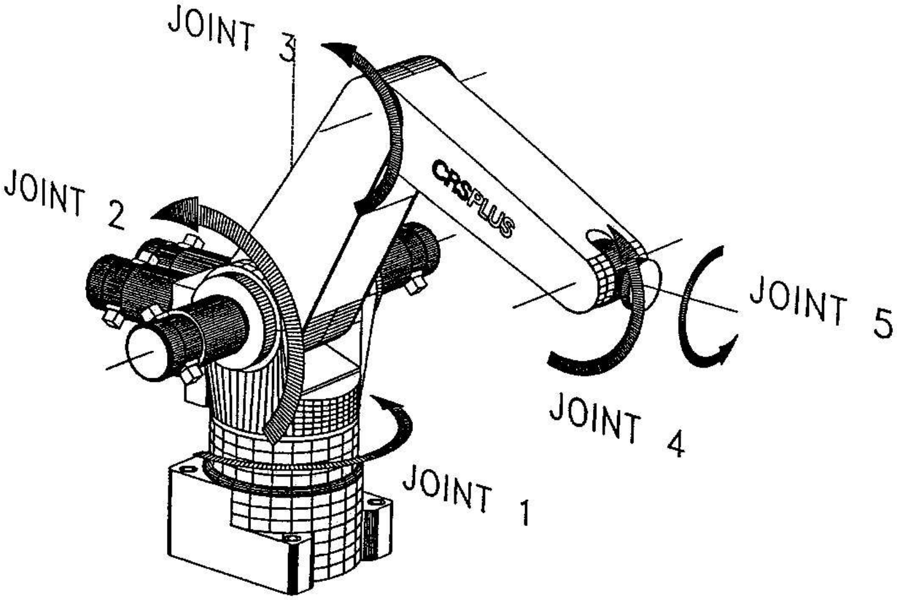


Figure 4.2: Joint configuration of the CRS+ A251 arm. From (CRSplus, 1990).

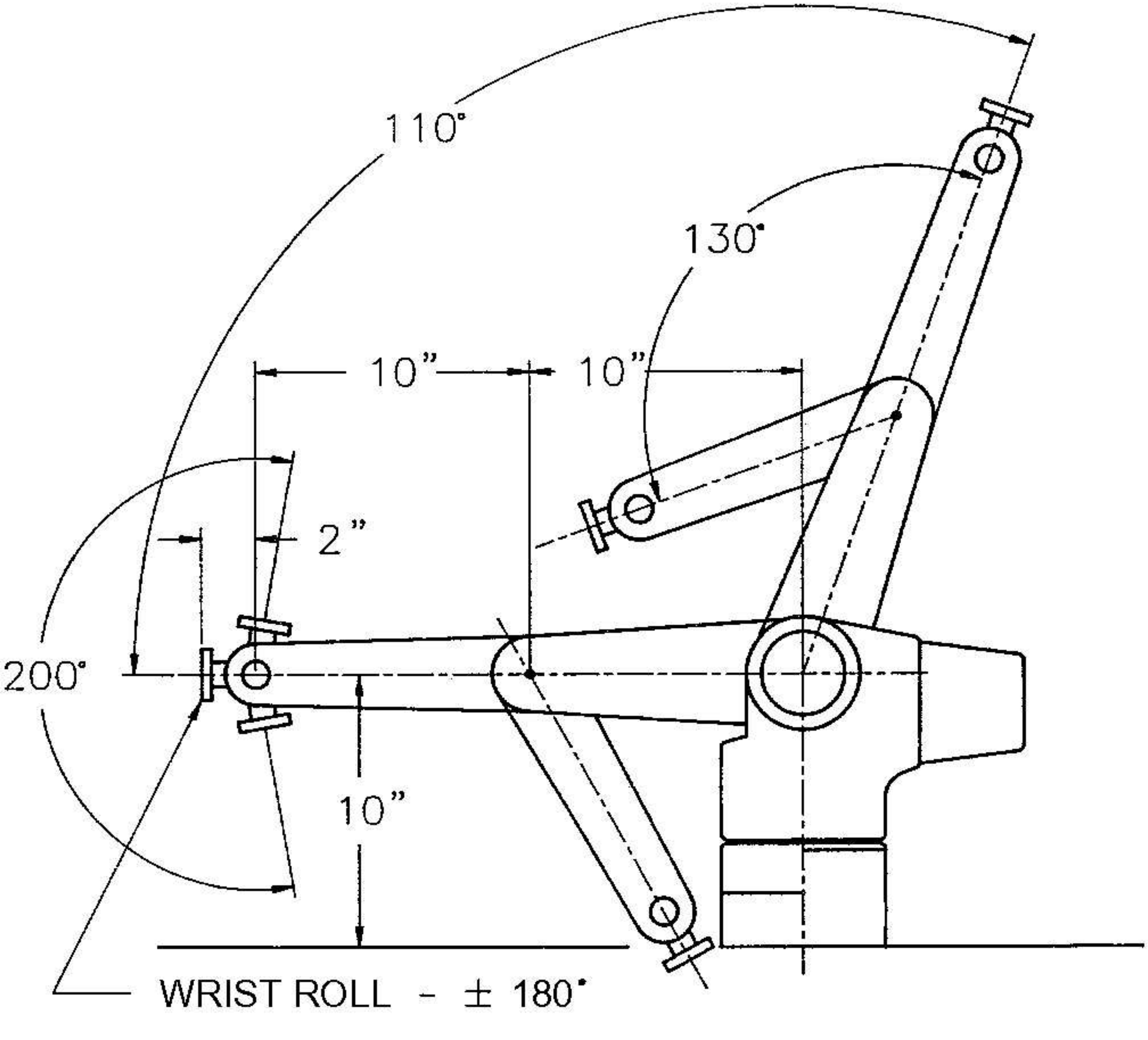


Figure 4.3: Joint limits of the CRS+ A251 manipulator. From (CRSplus, 1990).

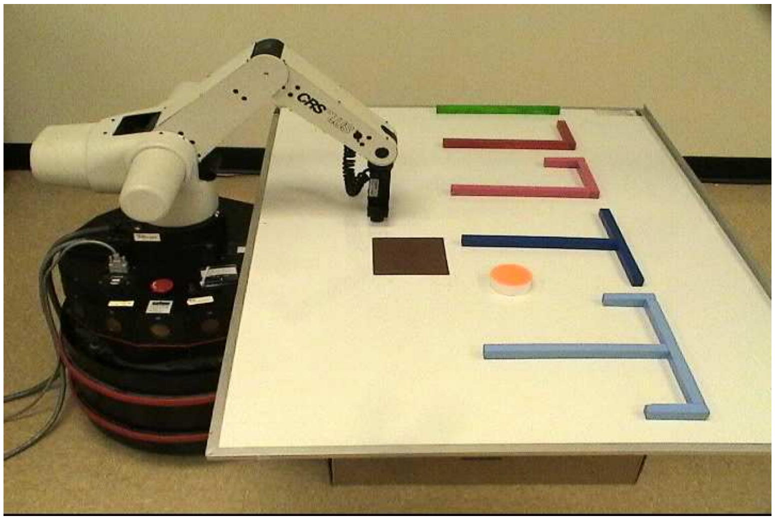


Figure 4.4: The mobile manipulator and five of the tools used in the experiments.

A computer camera was used to track the positions of the tools, objects, and the robot itself. The CMVision color tracking library that was integrated in the simulator was also used to track object in the real world. Two different positions were used for the camera depending on the task. These setups are described in the next chapters.

Both the manipulator and the Nomad 150 robots are controlled through a serial line. The robot control code and the color tracker were run on a Pentium IV machine (2.6 GHz, 1 GB RAM), running RedHat Linux 9.0.

CHAPTER V

SELF-DETECTION IN ROBOTS

5.1 Introduction

An important problem that many organisms have to solve early in their developmental cycles is how to distinguish between themselves and the surrounding environment. In other words, they must learn how to identify which sensory stimuli are produced by their own bodies and which are produced by the external world. Solving this problem is critically important for their normal development. For example, human infants which fail to develop self-detection abilities suffer from debilitating disorders such as infantile autism and Rett syndrome (Watson, 1994).

This chapter addresses the problem of autonomous self-detection by a robot. The chapter describes a methodology for autonomous learning of the characteristic delay between motor commands (efferent signals) and observed movements of visual stimuli (afferent signals). The robot estimates its own efferent-afferent delay (also called the perfect contingency) from self-observation data gathered by the robot while performing motor babbling. That is to say, the robot gathers this data while executing random rhythmic movements similar to the primary circular reactions described by Piaget. After the efferent-afferent delay is estimated, the robot imprints on that delay and can later use it to classify visual stimuli as either “self” or “other”. Results from robot experiments performed in environments with increasing degrees of difficulty are reported.

5.2 Related Work

This chapter was inspired by John Watson’s 1994 paper entitled “Detection of Self: The perfect algorithm.” Ever since I first read his paper I wanted to implement some of his ideas on a robot. At that time, however, this seemed so distant from my dissertation topic that I kept putting it off. Ironically, as my ideas about autonomous tool use kept evolving I came to realize that self-detection is an early developmental milestone that must be achieved before everything else can fall into place. This section reviews the main findings and theories about self-detection described in the literature.

5.2.1 Self-Detection in Humans

There are at least two fundamental questions about the self-detection abilities of humans: 1) What is the mechanism that is developed and used for self-detection?; and 2) When is this mechanism developed? This section summarizes the answers to these questions that are given in prior research.

Almost every major developmental theory recognizes the fact that normal development requires “an initial investment in the task of differentiating the self from the external world”(Watson, 1994). This is certainly the case for the two most influential theories of the 20-th century: Freud’s and Piaget’s. These theories, however, disagree about the ways in which the self-detection process is achieved. Freud and his followers believed that it is achieved very early in life through the gradual differentiation of the self from the mother (see Lewis and Brooks-Gunn (1979, p. 8-9)). Piaget, on the other hand, appears to ignore this question altogether. Nevertheless, he explicitly assumes that self-detection and the subsequent “self” versus “other” discrimination does occur because it is a necessary component of the secondary circular reactions (i.e., exploratory behaviors directed at external objects). Freud and Piaget, however, agree that the “self” emerges from actual experience and is not innately predetermined (Watson, 1994).

Modern theories of human development also seem to agree that the self is derived from actual experience. Furthermore, they identify the types of experience that are required for that: efferent-afferent loops that are coupled with some sort of probabilistic estimate of repeatability. The gist of these theories is summarized below with the help of three examples from Rochat (2003), Lewis and Brooks-Gunn (1979), and Watson (1994).

Rochat (2003) suggests that there are certain events that are *self-specifying*. These events can only be experienced by the owner of the body. The self-specifying events are also multimodal because they involve more than one sensory or motor modality. Because these events are unique to the owner of the body they are easy to identify and also to replicate. Rochat explicitly lists the following self-specifying events:

> "When infants experience their own crying, their own touch, or experience the perfect contingency between seen and felt bodily movements (e.g., the arm crossing the field of view), they perceive something that no one but themselves can perceive. The transport of the[ir] own hand to the face, very frequent at birth and even during the last trimester of pregnancy, is a unique tactile experience, unlike any other tactile experience as it entails a "double touch": the hand touches the face and simultaneously the face touching the hand. Same for the auditory experience of the[ir] own crying or the visual-proprioceptive experience accompanying self-produced movement. These basic perceptual (i.e., multimodal) experiences are indeed self-specifying, unlike any other perception experienced by the infant from birth and even prior to birth in the confines of the maternal womb."(Rochat, 2003, p. 723)

More than two decades earlier Lewis and Brooks-Gunn (1979) proposed a very similar process through which an infant can detect its own self. According to them the self is defined through action-outcome pairings (i.e., efferent-afferent loops) coupled with a probabilistic estimate of their regularity and consistency. Here is how they describe the emergence of what they call the "existential self", i.e., the self as a subject distinct from others and from the world:

> "This nonevaluative, existential self is developed from the consistency, regularity, and contingency of the infant's action and outcome in the world. The mechanism of reafferent feedback provides the first contingency information for the child; therefore, the kinesthetic feedback produced by the infant's own actions from the basis for the development of self. For example, each time a certain set of muscles operates (eyes close), it becomes black (cannot see). [...] These kinesthetic systems provide immediate and regular action-outcome pairings." (Lewis and Brooks-Gunn, 1979, p. 9)

The last example used to illustrate the self-detection abilities of humans comes from Watson (1994). Watson proposes that the process of self-detection is achieved by detecting the temporal contingency between efferent and afferent stimuli. The level of contingency that is detected serves as a filter that determines which stimuli are generated by the body and which ones are generated by the external world. In other words, the level of contingency is used as a measure of selfness. In Watson's own words:

> "Another option is that imperfect contingency between efferent and afferent activity implies out-of-body sources of stimulation, perfect contingency implies

> in-body sources, and non-contingent stimuli are ambiguous. In order to specify this mechanism, it is necessary to first specify what I mean by temporal contingency. What I mean is the temporal pattern between two events that potentially reflects the causal dependency between them." (Watson, 1994, p. 134)

All three examples suggest that the self is discovered quite naturally as it is the most predictable and the most consistent part of the environment. Furthermore, all seem to confirm that the self is constructed from self-specifying events which are essentially efferent-afferent loops or action-outcome pairs. There are many other studies that have reached similar conclusions. However, they are too numerous to be summarized here. For an extensive overview the reader is referred to: Lewis and Brooks-Gunn (1979) and Parker et al. (1994).

At least one study has tried to identify the minimum set of perceptual features that are required for self-detection. Flom and Bahrick (2001) showed that five-month-old infants can perceive the intermodal proprioceptive-visual relation on the basis of motion alone when all other information about the infants' legs was eliminated. In their experiments they fitted the infants with socks that contained luminescent dots. The camera image was preprocessed such that only the positions of the markers were projected on the TV monitor. In this way the infants could only observe a point-light display of their feet on the TV monitor placed in front of them. The experimental results showed that five-month-olds were able to differentiate between self-produced (i.e., contingent) leg motion and pre-recorded (i.e., non-contingent) motion produced by the legs of another infant. These results illustrate that only movement information alone might be sufficient for self-detection since all other features like edges and texture were eliminated in these experiments. The robot experiments described later use a similar experimental design as the robot's visual system has perceptual filters that allow the robot to see only the positions and movements of specific color markers placed on the robot's body. Similar to the infants in the dotted socks experiments, the robot can only see a point light display of its movements.

The second fundamental question about the self-detection abilities of humans is: When are these abilities developed? The developmental literature suggests that in human infants this task takes approximately 3 months (Lewis and Brooks-Gunn, 1979; Watson, 1985). This estimate is derived through experiments in which infants of various ages are tested for their looking preferences, i.e., for their preferences for stimuli. These experiments test whether or not infants can discriminate between the perfectly contingent movements of their own legs and the non-contingent movements of another baby's legs. The experimental setup typically consist of two TV monitors which the infant can observe at the same time. The first monitor shows the leg movements of the infant captured by the camera. The second monitor shows the leg movements of another infant recorded during a previous trial (Watson, 1985). The results of these experiments show that the distribution of looking preferences of 3-month-old infants is bimodal, i.e., half of the infants preferred to see their own movements and the other half preferred to see the movement of the other child (Watson, 1985, 1994). By the age of 5 months all children showed significant preferential fixation on the image that was not their contingent self.

Based on the results of these experiments, Watson (1994) has proposed that infants go through a developmental phase when they are about three-months-old during which they switch from self seeking to self avoiding. During the first 3 months the child is trying to estimate the perfect contingency between its own motor commands (efferent signals) and the resulting sensory perceptions (afferent signals). Once the perfect contingency is estimated the infant focuses its attention to other parts of the environment that show a less perfect contingency. Thus, sometime after the third month the attention mechanisms of the infant are modified to seek and interact with other parts of the environment (i.e., objects and people) that exhibit imperfect contingency.

Evidence for the existence of a self-seeking period followed by a self-avoidance period also comes from studies of children with genetic diseases like Rett syndrome (Watson, 1994). Children affected by this disorder appear to develop normally for a period of 6-16 months before they rapidly succumb to a form of severe mental retardation (Watson, 1994). This transition is very sudden and is accompanied by “increasingly repetitive hand-on-hand and hand-to-mouth” movements (Watson, 1994). Watson calls these movement patterns “deviant self seeking.” It seems that this genetic disorder flips a switch which reverses the developmental cycle and as a result the infants transition back to the self seeking patterns which they were performing during the first 3 months of their lives. There is also some evidence to suggest that the onset of schizophrenia in adult humans is sometimes accompanied by the loss of their capacity for self-recognition (Gallup, 1977, p. 331).

The notion of “self” has many other manifestations. Most of them are related to the social aspects of the self or what Lewis and Brooks-Gunn (1979) call the “categorical self.” Rochat (2003), for example, identified five levels of self-awareness as they unfold from the moment of birth to approximately 4-5 years of age. These levels are: 1) self-world differentiation; 2) a sense of how the body is situated in relation to other entities in the environment; 3) the development of the “Me” concept which children begin to use in order to distinguish between themselves and other people; 4) the development of the temporally extended “Me” concept that children can use to recognize themselves in photographs; 5) the development of theories of mind and representational abilities for others in relation to the self. With the exception of the first level, however, all of these are beyond the scope of this dissertation.

5.2.2 Self-Detection in Animals

Many studies have focused on the self-detection abilities of animals. Perhaps the most influential study was performed by Gallup (1970) who reported for the first time the abilities of chimpanzees to self-recognize in a mirror. Gallup anesthetized several chimpanzees and placed odorless color markers on their faces while they were asleep. Once the chimpanzees woke up from the anesthesia they were allowed to move about their cages. After a brief adaptation period, a mirror was introduced into their environment. Gallup measured that there was a sharp increase in the number of self-directed movements toward the spot where the marker was placed after the mirror was introduced. Furthermore, the chimps directed their exploratory actions not toward the image in the mirror but toward their own faces. This shows that they indeed understood the difference between the two. For a more recent treatment of the mirror test see (Gallup et al., 2002).

Gallup’s discovery was followed by a large number of studies that have attempted to test which species of animals can pass the mirror test. Somewhat surprisingly, the number turned out to be very small: chimpanzees, orangutans, and bonobos (one of the four great apes, often called the forgotten ape, see (de Waal, 1997)). There is also at least one study which has documented similar capabilities in bottlenose dolphins (Reiss and Marino, 2001). However, the dolphins in that study were not anesthetized so the results are not directly comparable with those of the other mirror experiments. Another recent study reported that one Asian elephant (out of three that were tested) conclusively passed the mirror test (Plotnik et al., 2006). Attempts to replicate the mirror test with other primate and non-primates species have failed.

Unfortunately, the scientific community is still divided as to what the mirror test is a test of (Povinelli and Cant, 1995; Barth et al., 2004; de Waal et al., 2005). It is unlikely that the process of natural selection would have favored an ability for self-recognition in a mirror when this carries no immediate advantages (not to mention that there were no mirrors in the jungles of pre-historic Africa). Therefore, other explanations are needed.

Some studies have shown that gorillas (which are the only great apes that don't pass the mirror test) and rhesus monkeys would show intense interest in exploring markers that are placed on their wrists and abdomen (i.e., places on their bodies that they can normally see). However, they will not pay attention to markers on their faces, i.e., markers which cannot be seen without a mirror (Shillito et al., 1999; Gallup et al., 1980). On the other hand, one of the first things that chimpanzees do when they are exposed to a mirror is to study parts of their bodies which they cannot normally see, e.g., their faces, the insides of their mouths and nostrils (Barth et al., 2004). Several studies have also reported that gorillas learn to control the image in the mirror by exploiting contingencies between their own movements and the movements of the mirror image (Parker, 1991). Similarly, juvenile chimpanzees have been observed to display contingent behaviors toward the mirror image without showing any self-exploratory behaviors (Eddy et al., 1996). Thus, it is plausible that, from a developmental perspective, the process of self-recognition goes through a stage of self-detection based on detecting temporal contingencies. Self-recognition abilities, however, probably require a much more detailed representation for the body.

Gallup (1977) has argued that the interspecies differences are probably due to different degrees of self-awareness. Another reason for these differences “may be due to the absence of a sufficiently well-integrated self-concept” (Gallup, 1977, p. 334). Yet another reason according to Gallup (1977) might be that the species that pass the mirror test can direct their attention both outward (towards the external world) and inwards (towards their own bodies), i.e., they can become “the subject of [their] own attention.” Humans, of course, have the most developed self-exploration abilities and have used them to create several branches of science, e.g., medicine, biology, and genetics.

Another hypothesis is that the ability to detect oneself in the mirror is only one small manifestation of a much more sophisticated system which is used to represent the body and its positions in space (Povinelli and Cant, 1995; Barth et al., 2004). The primary function of this system is to control and plan body movements. Barth et al. (2004) claim that this would explain why gorillas don’t pass the mirror test but orangutans do. Since gorillas are more evolutionarily advanced than orangutans one would expect that the opposite would be true. The answer, according to Barth et al. (2004), is that the ability to represent body movements at a fine level of detail was first developed in the last common ancestor of the two species which lived about 15-20 million years ago. Gorillas have since lost this ability as it was not required for their terrestrial lifestyles. Orangutans, on the other hand, have inherited that ability and have used it actively to support their arboreal lifestyles. Because of their large body weight, orangutans must plan their movements very precisely from branch to branch to avoid falling down from trees (Povinelli and Cant, 1995).

Furthermore, animals with more elaborate representation of their bodies (i.e., with more elaborate body schemas) are likely to be more proficient tool users (Povinelli et al., 2000, p. 333). In fact, McGrew (1992) discovered that the factor which best predicts the degree of tool-using abilities in different primate species is their ability to recognize themselves in the mirror, i.e., their ability to pass the mirror test. Habitat, food supplies, and other factors were not as predictive. Westergaard and Suomi (1995) have also concluded that “the psychological capacities that underlie the use of tools are associated with those that underlie mirror inspection”(p. 221).

Since many species of animals have been observed to use tools (see Figure 2.3) but only three species of primates pass the mirror test it seems that a detailed representation of the body is not strictly required for tool use. However, as Barth et al. (2004) point out: “the hypothesis is simply that the more explicitly the distinction between self and object can be represented, the more rapidly tool-use will emerge” (Barth et al., 2004, p. 33).

Similar conclusions have been reached by Westergaard and Suomi (1995) who made the following predictions based on their experiments with tufted capuchin monkeys: “We predict that at the intraspecific level the ability to use tools is positively correlated with mirror interest and that this relationship occurs independently of a capacity for self-recognition. We further predict that at the interspecific level the ability to use tools is a necessary, but not sufficient, condition for the emergence of self-recognition. We believe that mirror-aided self-recognition reflects a cognitive process that enables animals to fully comprehend the properties of a specific type of tool”(Westergaard and Suomi, 1995, p. 222).

5.2.3 Self-Detection in Robots

Self-detection experiments with robots are still rare. One of the few published studies on this subject was conducted by Michel, Gold, and Scassellati (2004). They implemented an approach to autonomous self-detection similar to the temporal contingency strategy described by Watson (1994). Their robot was successful in identifying movements that were generated by its own body. The robot was also able to identify the movements of its hand reflected in a mirror as self-generated motion because the reflection obeyed the same temporal contingency as the robot’s body.

One limitation of their study is that the self-detection is performed at the pixel level and the results are not carried over to high-level visual features of the robot’s body. Thus, there is no permanent trace of which visual features constitute the robot’s body. Because of this, the detection can only be performed when the robot is moving. The study presented in this chapter goes a step further and keeps a probabilistic estimate across the visual features that the robot can detect as to whether or not they belong to the robot’s body. In this way, the stimuli can be classified as either “self” or “other” even when the robot is not moving.

Another limitation of the Michel et al. (2004) study is that the training procedure used to estimate the efferent-afferent delay can only be performed if the robot is the only moving object in the environment. The algorithm described in Section 5.8 does not suffer from this limitation.

Another team of roboticists have attempted to perform self-detection experiments with robots based on a different self-specifying event: the so-called double touch (Yoshikawa et al., 2004). The double touch is a self-specifying event because it can only be experienced by the robot when it touches its own body. This event cannot be experienced if the robot touches an object or if somebody else touches the robot since both cases would correspond to a single touch event. From the papers that have been published by this team so far, however, it is not clear if the results of the self-detection have been used for some other purpose, e.g., to bootstrap the robot's learning of the properties of objects. Nevertheless, the self may be identified more robustly when inputs from double touch events and motor-visual efferent-afferent loops are combined. This seems a fruitful area for future research.

5.3 Problem Statement

The related work shows that the processes of self-detection and self-recognition are developed under different developmental schedules and are influenced by a number of factors. In humans, this developmental process can be quite complicated and requires many years of social interactions before all aspects of the self can be fully developed (Rochat, 2003). The robotics implementation described later in this chapter focuses only on the problem of autonomous self-detection. In other words, the robot only learns which visual features belong to its body and which do not. The problems of self-recognition and relating the self to other social agents in the environment are not addressed here.

For the sake of clarity the problem of autonomous self-detection by a robot will be stated explicitly using the following notation. Let the robot have a set of joints $J = \{j_1, j_2, \ldots, j_n\}$ with corresponding joint angles $\Theta = \{q_1, q_2, \ldots, q_n\}$. The joints connect a set of rigid bodies $B = \{b_1, b_2, \ldots, b_{n+1}\}$ and impose restrictions on how the bodies can move with respect to one another. For example, each joint, j_i, has lower and upper joint limits, q_i^L and q_i^U, which are either available to the robot's controller or can be inferred by it. Each joint, j_i, can be controlled by a motor command, $move(j_i, q_i, t)$, which takes a target joint angle, q_i, and a start time, t, and moves the joint to the target joint angle. More than one *move* command can be active at any given time.

Also, let there be a set of visual features $F = \{f_1, f_2, \ldots, f_k\}$ that the robot can detect and track over time. Some of these features belong to the robot's body, i.e., they are located on the outer surfaces of the set of rigid bodies, B. Other features belong to the external environment and the objects in it. The robot can detect the positions of visual features and detect whether or not they are moving at any given point in time. In other words, the robot has a set of perceptual functions $P = \{p_1, p_2, \ldots, p_k\}$, where $p_i(f_i, t) \rightarrow \{0, 1\}$. That is to say, the function p_i returns 1 if feature f_i is moving at time t, and 0 otherwise.

The goal of the robot is to classify the set of features, F, into either "self" or "other". In other words, the robot must split the set of features in two subsets, F_{self} and F_{other}, such that $F = F_{self} \cup F_{other}$.

5.4 Methodology

The problem of self-detection by a robot is divided into two separate problems as follows:

Sub-problem 1: How can a robot estimate its own efferent-afferent delay, i.e., the delay between the robot's motor actions and their perceived effects?

Sub-problem 2: How can a robot use its efferent-afferent delay to classify the visual features that it can detect into either "self" or "other"?

The methodology for solving these two sub-problems is illustrated by two figures. Figure 5.1 shows how the robot can estimate its efferent-afferent delay (sub-problem 1) by measuring the elapsed time from the start of a motor command to the start of visual movement. The approach relies on detecting the temporal contingency between motor commands and observed movements of visual features. To get a reliable estimate of the delay the robot gathers statistical information by executing multiple motor commands over an extended period of time. Section 5.8 shows that this approach is reliable even if there are other moving visual features in the environment as their movements are typically not correlated with the robot's motor commands. Once the delay is estimated the robot imprints on its value (i.e., remembers it irreversibly) and uses it to solve sub-problem 2.

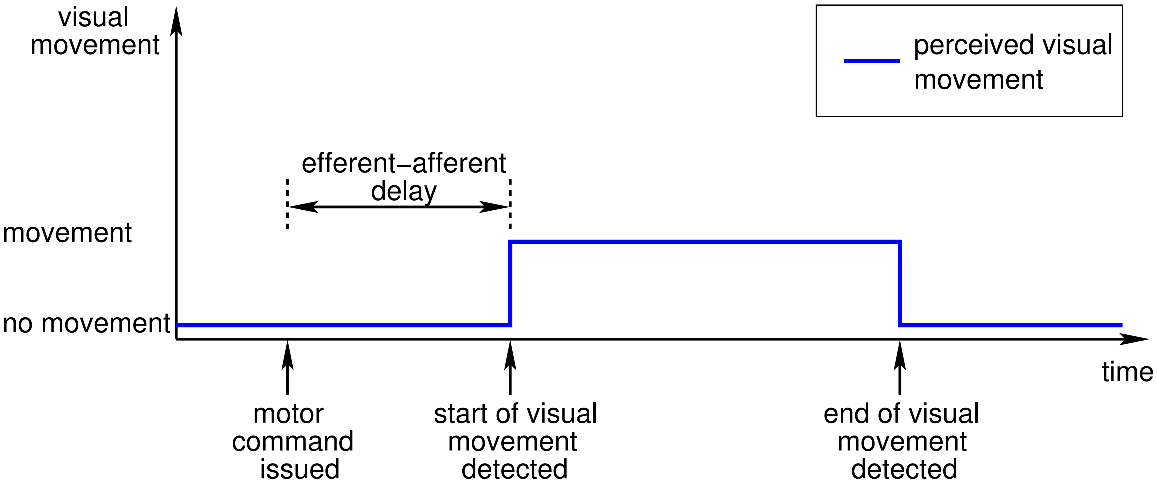

visual movement
perceived visual movement
efferent–afferent delay
movement
no movement
time
motor command issued
start of visual movement detected
end of visual movement detected

Figure 5.1: The efferent-afferent delay is defined as the time interval between the start of a motor command (efferent signal) and the detection of visual movement (afferent signal). The goal of the robot is to learn this characteristic delay (also called the perfect contingency) from self-observation data.

Figure 5.2 shows how the estimated efferent-afferent delay can be used to classify visual features as either “self” or “other” (sub-problem 2). The figure shows three visual features and their detected movements over time represented by red, green, and blue lines. Out of these three features only feature 3 (blue) can be classified as “self” as it is the only one that conforms to the perfect contingency. Feature 1 (red) begins to move too late after the motor command is issued and feature 2 (green) begins to move too soon after the movement command is issued.

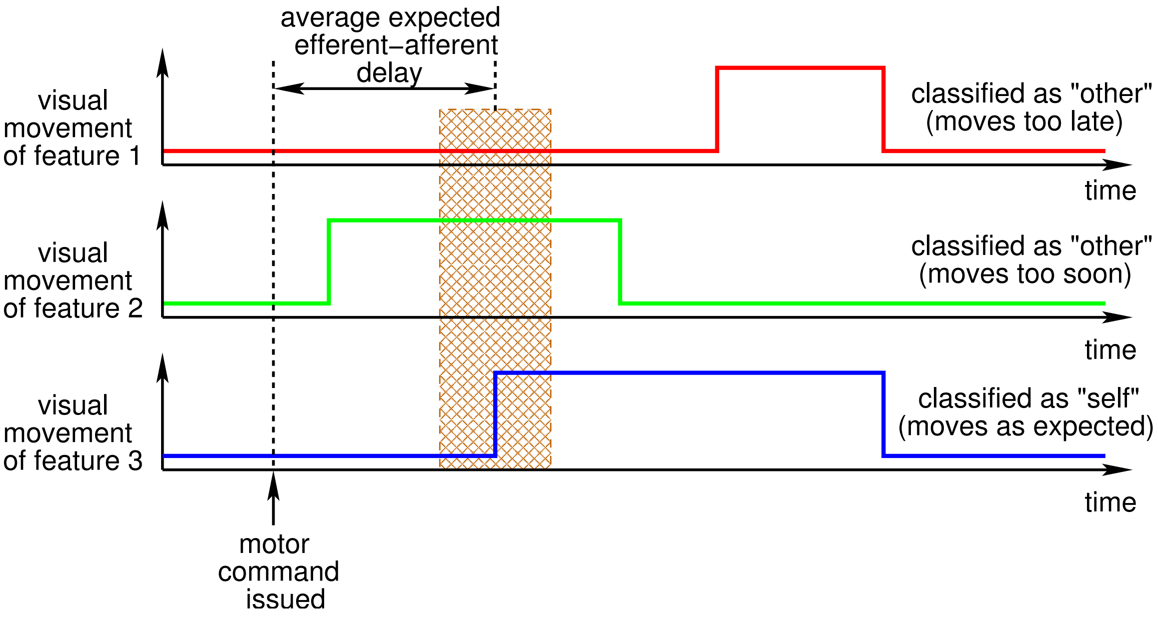


Figure 5.2: “Self” versus “Other” discrimination. Once the robot has learned its efferent-afferent delay it can use its value to classify the visual features that it can detect into “self” and “other.” In this figure, only feature 3 (blue) can be classified as self as it starts to move after the expected efferent-afferent delay plus or minus some tolerance (shown as the brown region). Features 1 and 2 are both classified as “other” since they start to move either too late (feature 1) or too soon (feature 2) after the motor command is issued.

So far, the methodology described here is similar to the one described by Michel et al. (2004). As mentioned in Section 5.2.3, however, their approach performs the detection on the pixel level and does not generalize the detection results to more high level perceptual features. The methodology presented here overcomes this problem as described below.

A classification based on a single observation can be unreliable due to sensory noise or a lucky coincidence in the movements of the features relative to the robot's motor command. Therefore, the robot maintains a probabilistic estimate for each feature as to whether or not it is a part of the robot's body. The probabilistic estimate is based on the sufficiency and necessity indices proposed by Watson (1994). The sufficiency index measures the probability that the stimulus (visual movement) will occur during some specified period of time after the action (motor command). The necessity index, on the other hand, measures the probability that the action (motor command) was performed during some specified period of time before the stimulus (visual movement) was observed. The robot continuously updates these two indexes for each feature as new evidence becomes available. Features for which both indexes are above a certain threshold are classified as "self." All others are classified as "other." Section 5.9 provides more details about this procedure.

The remaining sections in this chapter describe the individual components necessary to solve the two sub-problems of self-detection. Section 5.5 summarizes the method for detecting visual features. Section 5.6 describes the motor babbling procedure that is used by the robot to gather self-observation data. Section 5.7 explains the procedure for detecting the movements of visual features. Section 5.8 describes the method for learning the efferent-afferent delay of the robot and the experiments that were used to test this method. Finally, Section 5.9 presents the results of robot experiments for "self" versus "other" discrimination.

5.5 Detecting Visual Features

All experiments in this chapter were performed using the robot arm described in Section 4.2. The movements of the robot were restricted to the vertical plane. In other words, only joints 2, 3, and 4 (i.e., shoulder pitch, elbow pitch, and wrist pitch) were allowed to move (see Figure 4.2). Joints 1 and 5 (i.e., waist roll and wrist roll) were disabled and their joint angles were set to 0. The mobile base of the robot was also disabled (i.e., there were no linear translations).

The experimental setup is shown in Figure 5.3. Six color markers (also called body markers) were placed on the body of the robot as shown in Figure 5.4. The robot's body markers were located and tracked using color segmentation (see Figure 5.5). The position of each marker was determined by the centroid of the largest blob that matched the specific color. The color segmentation was performed using a computer vision code which performs histogram matching in HSV color space with the help of the openCV library (an open source computer vision package). The digital video camera (Sony EVI-D30) was mounted on a tripod and its field of view was adjusted so that it can see all body markers in all possible joint configurations of the robot. The image resolution was set to 640x480. For all experiments described in this chapter the frames were captured at 30 frames per second.

Color tracking is a notoriously difficult problem in computer vision. In the course of this research a disproportionately large amount of time was spent fine tuning the color tracker and selecting distinct colors for the experiments. It was established empirically that no more than 12 colors can be tracked reliably for extended periods of time. The following colors were selected: dark orange, dark red, dark green, dark blue, yellow, light green, pink, tan, orange, violet, light blue, and red. The color tracker was fine tuned to look for areas of the image that have these specific colors. All other areas were filtered out.

It is important to note that it is possible to use other colors than the ones listed above but it turned out to be impossible to track more than 12 colors at the same time in the light conditions in the lab. The tracking results depend on the ambient light as well as the amount of time that the camera has been turned on (most likely related to its operating temperature). The color limit is slightly higher (17 colors) for the simulator which does not suffer from transient lighting effects (the color limit is not much higher because the simulator supports textures and shading that can significantly change the appearance of uniform colored surfaces). For comparison, the robots in the Robocup legged league are limited to 8 colors (yellow, cyan, pink, green, orange, white, red, and blue) which are also very carefully selected to maximize detectability.

Nevertheless, color tracking is computationally fast and convenient method to use as it is very easy to place color markers on the robot and on the objects that the robot uses. An alternative, but more computationally expensive approach, would be to paint the body of the robot using different textures and track only locations with unique local features.

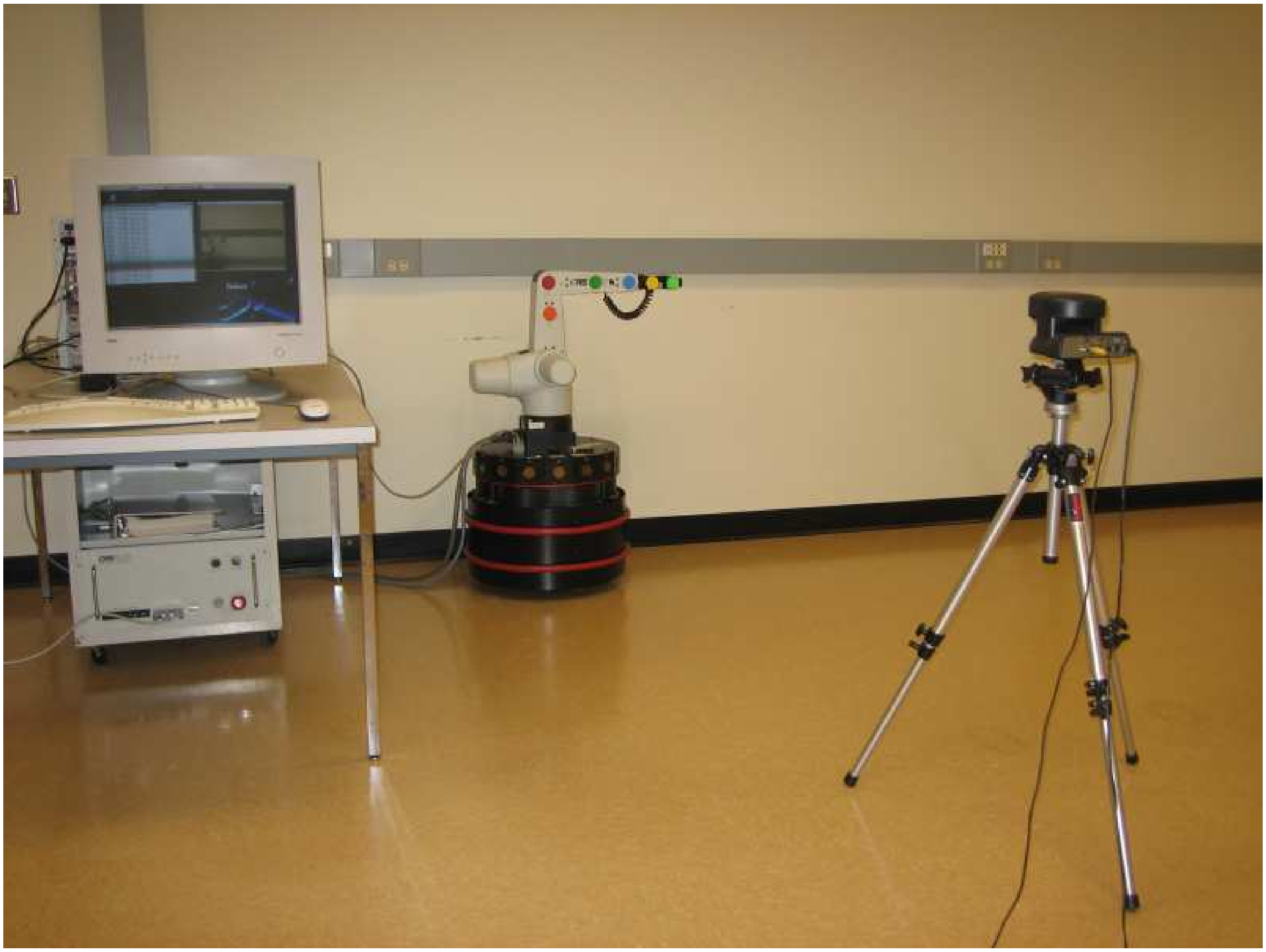


Figure 5.3: The experimental setup for most of the experiments described in this chapter.

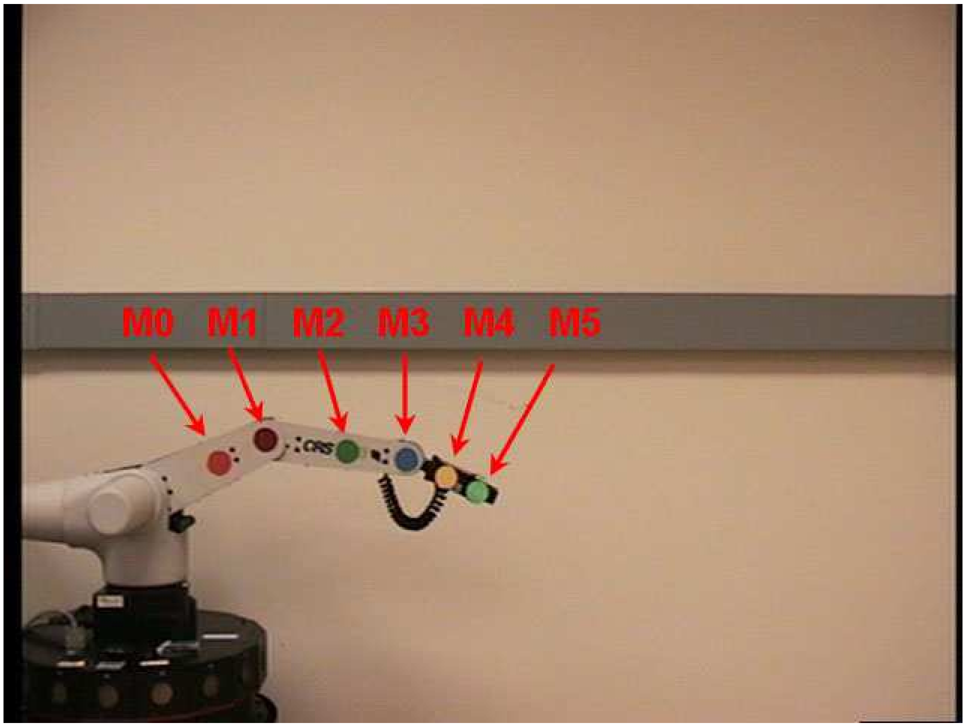


Figure 5.4: The figure shows the positions and colors of the six body markers. Each marker is assigned a number which is used to refer to this marker in the text and figures that follow. From left to right the markers have the following colors: 0) dark orange; 1) dark red; 2) dark green; 3) dark blue; 4) yellow; 5) light green.

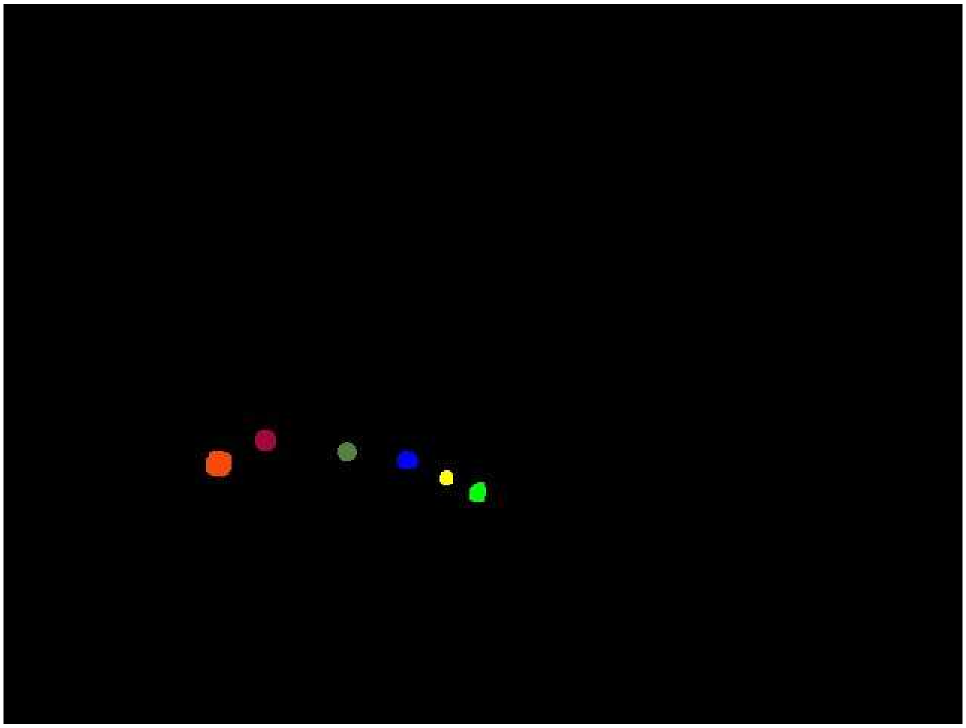


Figure 5.5: Color segmentation results for the frame shown in Figure 5.4.

5.6 Motor Babbling

All experiments described in this chapter rely on a common motor babbling procedure which allows the robot to gather self-observation data (both visual and proprioceptive) while performing random joint movements. This procedure consists of random joint movements similar to the primary circular reactions described by Piaget (see Section 2.3.1) as they are not directed at any object in the environment. Algorithm 1 shows the pseudocode for the motor babbling procedure.

During motor babbling the robot's controller randomly generates a target joint vector and then tries to move the robot to achieve this vector. The movements are performed by adjusting each joint angle in the direction of the target joint angle. If the target joint vector cannot be achieved within some tolerance (2 degrees per joint was used) then after some timeout period (8 seconds was used) the attempt is aborted and another random joint vector is chosen for the next iteration. The procedure is repeated for a specified number of iterations, i.e., random motor commands. The number of iterations was set to 500 for the experiments described below.

Algorithm 1 uses several functions which are not defined but are standard for many programming environments. RandomInt(min,max) returns an integer number in the interval $[min, max]$. For example, RandomInt$(0, 1)$ returns either a 0 or a 1 with 50% probability for each. Similarly, RandomFloat(min, max) returns a random floating point number in the interval (min, max). The function ABS(*number*) returns the absolute value of its parameter. The function GetTime() returns the system time and the function Sleep(*time*) waits for a given amount of time.

In addition to that, the robot's interface which is represented with the *robot* object has several functions (GetNumJoints, GetLowerJointLimit, GetUpperJointLimit, and MoveToTargetJointVector) that perform the operations implied by their names. It is worth noting, however, that the last function, MoveToTargetJointVector, is asynchronous and returns immediately after the move command is issued to the hardware, instead of waiting for the move to finish. This is necessary due to the fact that some of the randomly generated joint vectors correspond to invalid joint configurations (e.g., those that result in self collisions).

The *motor babbling* algorithm is not affected by self-collisions because a collision only prevents the robot from reaching the randomly generated target joint vector. After a timeout period the algorithm abandons that goal and selects another target joint vector for the next iteration. Nevertheless, self-collisions can damage the robot and should be avoided if possible. For the CRS arm, a side effect of self collisions is that the power breaker for at least one of the joints is triggered. This renders the robot unusable until the power is restored. Therefore, the motor babbling routine was modified to generate only body poses that do not result in self-collisions (this change is not shown in Algorithm 1 as it is specific for this robot in this mounting configuration only). Because the restricted poses represent only a small fraction of all possible body poses this is an acceptable solution that ensures the safety of the robot and does not compromise the validity of the experimental results.

Figure 5.6 shows some joint configurations that were randomly selected by the motor babbling procedure. The corresponding color segmentation results used to track the positions of the robot's body markers are shown in Figure 5.7.

Algorithm 1 Motor Babbling

```
GetRandomJointVector($robot$)
for $j \leftarrow 0$ to $nJoints$ do
```

```
GetRandomJointVector($robot$)
$nJoints \leftarrow robot$.GetNumJoints()
for $j \leftarrow 0$ to $nJoints$ do
    $moveThisJoint \leftarrow$ RandomInt(0,1)
    if $moveThisJoint = 1$ then
        $lowerLimit \leftarrow robot$.GetLowerJointLimit($j$)
        $upperLimit \leftarrow robot$.GetUpperJointLimit($j$)
        $JV[j] \leftarrow$ RandomFloat($lowerLimit$, $upperLimit$)
    else
        // Keep the the current joint angle for this joint.
        $JV[j] \leftarrow robot$.GetCurrentJointAngle($j$)
    end if
end for
return $JV$

IsRobotAtTargetJointVector($robot, targetJV, tolorance$)
$nJoints \leftarrow robot$.GetNumJoints()
for $j \leftarrow 0$ to $nJoints$ do
    $dist \leftarrow$ ABS($targetJV[j]$ - $robot$.GetCurrentJointAngle($j$))
    if $dist > tolerance$ then
        return $false$
    end if
end for
return $true$

MotorBabbling($robot, nIterations, timeout, tolerance, sleepTime$)
for $i \leftarrow 0$ to $nIterations$ do
    $motor[i].targetJV \leftarrow$ GetRandomJointVector($robot$)
    $motor[i].timestamp \leftarrow$ GetTime()
    repeat
        $robot$.MoveToTargetJointVector($motor[i].targetJV$)
        Sleep($sleepTime$)
        if (GetTime() - $motor[i].timestamp$) $> timeout$ then
            // Can't reach that joint vector. Try another one on the next iteration.
            break
        end if
        $done \leftarrow$ IsRobotAtTargetJointVector($robot$, $motor[i].targetJV$, $tolerance$)
    until $done = true$
end for
return $motor$
```

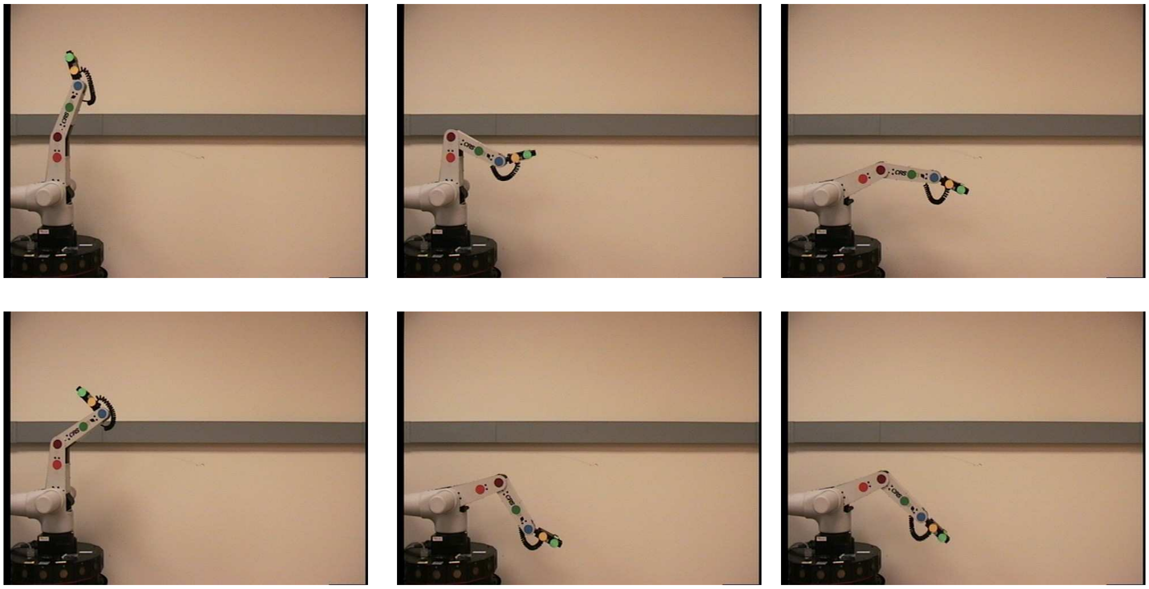


Figure 5.6: Several of the robot poses selected by the motor babbling procedure.

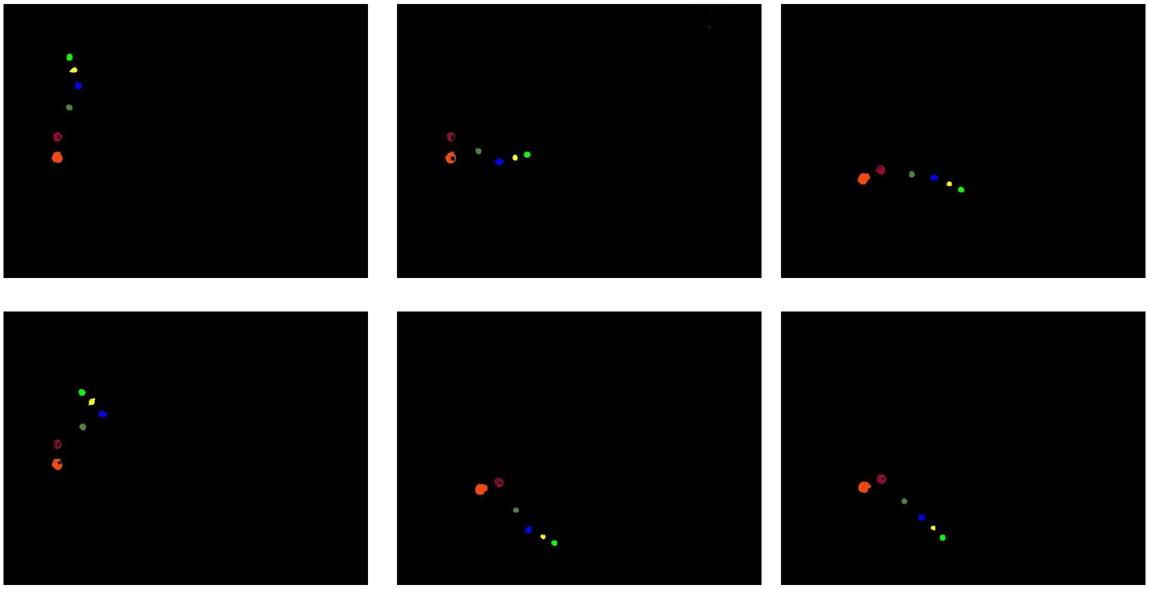


Figure 5.7: Color segmentation results for the robot poses shown in Figure 5.6.

5.7 Visual Movement Detection

Two methods for detecting the movements of the visual features (color markers) were tried. The first method compared the centroid position of each marker from one frame to the next. If the distance between these two positions was more than some empirically established threshold (1.5 pixels) the marker was declared as moving. In the course of this research, however, it became clear that this method is not very reliable. The reason for this is that the body markers move with different speeds depending on their positions on the robot's body. Markers that are placed on the robot's wrist, for example, move faster than markers that are placed on the robot's shoulder (see Figure 5.8). Thus, the markers located on the wrist appear to start moving sooner than the markers located on the shoulder. The opposite result is observed at the end of a movement: markers placed on the shoulder of the robot appear to stop moving sooner than markers placed on the wrist. This effect is observed even though the robot's movements are executed such that all actively controlled joints start and stop moving at the same time.

While it seems that this problem can be resolved if the movement threshold is reduced (e.g., from 1.5 pixels per frame to 0.5 pixels per frame) this actually results in total degradation of the quality of the tracking results. The reason for this is that the smaller movement threshold has approximately the same magnitude as the position detection noise for each marker even when the the robot is not moving and the markers are static (see Figure 5.9). Thus, if the movement threshold is reduced to less than 1.5 pixels per frame then marker movements will be detected even though the robot is not moving.

The position detection results shown in Figure 5.9 are quite decent as the marker positions are detected with sub-pixel accuracy (approximately 0.3 pixels per frame). Nevertheless, the figure also shows why the first movement detection approach is not feasible. If the movement threshold is reduced too much the position detection noise becomes almost as large as the movement threshold. This makes it impossible to distinguish between a marker movement resulting from a corresponding robot movement and a marker movement resulting from detection noise. Therefore, this first method for movement detection was abandoned and not used in any of the experiments. Instead, an alternative method described below was used.

The second movement detection method overcomes the noise problems mentioned above by performing the movement detection over a fixed interval of time which is longer than the interval between two consecutive frames. Therefore, this method can compensate for the frame-to-frame tracking noise as it looks for movements over a longer time window. In this way, the movement threshold can be larger than the tracking noise. Another advantage of this method is that it is less sensitive to the small variations in the frame rate due to system load.

In the final implementation, for each image frame a color marker was declared to be moving if its position changed by more than 1.5 pixels during the 0.1 second interval immediately preceding the current frame. The timing intervals were calculated from the timestamps of the frames stored in the standard UNIX format.

The result of this tracking technique is a binary 0/1 signal for each of the currently visible markers, similar to the graphs shown in Figure 5.2. These signals are still slightly noisy and therefore they were filtered with a box filter (also called averaging filter) of width 5 which corresponds to smoothing each tracking signal over 5 consecutive frames. The filter changes the values of the movement detection signal to the average for the local neighborhood. For example, if the movement detection signal is 001011100 then the filter will output 000111100. On the other hand, if the sequence is 001000 or 001100 then the filter will output 000000.

Algorithm 2 shows the pseudocode for the movement detector and the box filter.

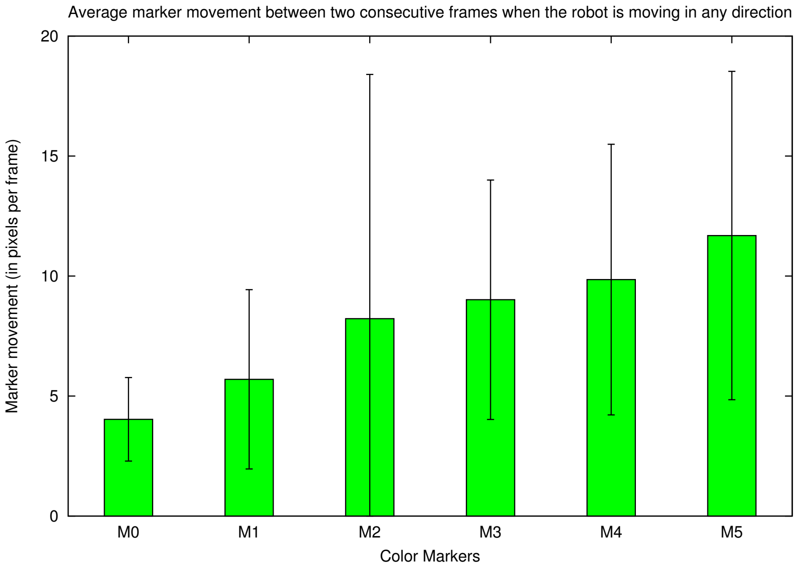


Figure 5.8: Average marker movement between two consecutive frames when the robot *is* moving. The results are in pixels per frame for each of the six body markers.

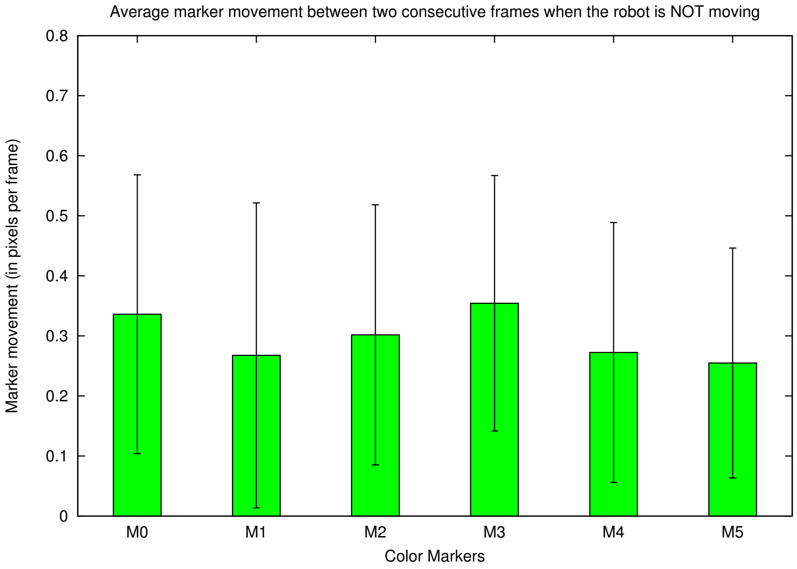


Figure 5.9: Average marker movement between two consecutive frames when the robot *is not* moving. In other words, the figure shows the position detection noise when the six body markers are static. The results are in pixels per frame.

Algorithm 2 Movement Detection

```
IsMoving($markerID, treshold, imageA, imageB$)
$posA \leftarrow$ FindMarkerPosition($markerID, imageA$)
$posB \leftarrow$ FindMarkerPosition($markerID, imageB$)
$\Delta x \leftarrow posA.x - posB.x$
$\Delta y \leftarrow posA.y - posB.y$
$dist \leftarrow \sqrt{(\Delta x)^2 + (\Delta y)^2}$
if $dist > threshold$ then
    return 1
else
    return 0
end if

BoxFilter($sequence[\,][\,], index, m$)
$sum \leftarrow 0$
for $i \leftarrow index - 2$ to $index + 2$ do
    $sum \leftarrow sum + sequence[i][m]$
end for
if $sum \geq 3$ then
    return 1
else
    return 0
end if

MovementDetector($nFrames, \Delta t, treshold$)
// Buffer some frames in advance so the BoxFilter can work OK
for $i \leftarrow 0$ to 3 do
    $frame[i].image \leftarrow$ GetNextFrame()
    $frame[i].timestamp \leftarrow$ GetTime()
end for
for $i \leftarrow 4$ to $nFrames$ do
    $frame[i].image \leftarrow$ GetNextFrame()
    $frame[i].timestamp \leftarrow$ GetTime()
    // Find the index, k, of the frame captured $\Delta t$ seconds ago
    $startTS \leftarrow frame[i].timestamp$ - $\Delta t$
    $k \leftarrow index$
    while $((frame[k].timestamp < startTS)$ and $(k > 0))$ do
        $k \leftarrow k - 1$
    end while

    // Detect marker movements and filter the data
    for $m \leftarrow 0$ to $nMarkers$ do
        $MOVE[i][m]$       $\leftarrow$ IsMoving($m, treshold, frame[i].image, frame[k].image$)
        $MOVE[i-2][m] \leftarrow$ BoxFilter($MOVE, i-2, m$)
    end for
end for
return $MOVE$
```

5.8 Learning the Efferent-Afferent Delay

This section describes a procedure that can be used by a robot to estimate its efferent-afferent delay from self-observation data (i.e., a procedure which solves sub-problem 1 described in Section 5.4). The methodology for solving this problem was already described in Section 5.4 and its mathematical formulation was given in Section 5.3. The different experimental conditions used to test the delay estimation procedure and the obtained experimental results are described in Sections 5.8.1 - 5.8.4. Section 5.9 describes how the results from this section can be used to solve the problem of "self" versus "other" discrimination (i.e., sub-problem 2 described in Section 5.4).

The pseudocode for the procedure for estimating the efferent-afferent delay of a robot is shown in Algorithm 3. The algorithm uses the results from the motor babbling procedure described in Section 5.6, i.e., it uses the array of motor commands and their timestamps. It also uses the results from the movement detection method described in Section 5.7, i.e., it uses the number of captured frames and the *move* array which holds information about what feature was moving during which frame. The algorithm is presented in batch form but it is straightforward to rewrite it in incremental form.

The algorithm maintains a histogram of the measured delays over the interval $[0, 6)$ seconds. Delays longer than 6 seconds are ignored. Each bin of the histogram corresponds to $1/30^{-th}$ of a second, which is equal to the time interval between two consecutive frames. For each frame the algorithm checks which markers, if any, are starting to move during that frame. This information is already stored in the *move* array which is returned by the MovementDetector function in Algorithm 2. If the start of a movement is detected, the algorithm finds the last motor command that was executed prior to the current frame. The timestamp of the last motor command is subtracted from the timestamp of the current frame and the resulting delay is used to update the histogram. Only one histogram update per frame is allowed, i.e., the bin count for only one bin is incremented by one. This restriction ensures that if there is a large object with many moving parts in the robot's field of view the object's movements will not bias the histogram and confuse the detection process. The code for the histogram routines is shown in Algorithm 4 and Algorithm 5.

Algorithm 3 Learning the efferent-afferent delay

```
CalculateEfferentAfferentDelay($nFrames, frame[\,], MOVE[\,][\,], motor[\,]$)

// Skip the frames that were captured prior to the first motor command.
$start \leftarrow 1$
while $frame[start].timestamp < motor[0].timestamp$ do
    $start \leftarrow start + 1$
end while

// Create a histogram with bin size=$1/30^{-th}$ of a second
// for the time interval $[0, 6)$ seconds.
$hist \leftarrow$ InitHistogram(0.0, 6.0, 180)

$idx \leftarrow 0$ // Index into the array of motor commands
for $k \leftarrow start$ to $nFrames - 1$ do
    // Check if a new motor command has been issued.
    if $frame[k].timestamp > motor[idx + 1].timestamp$ then
        $idx \leftarrow idx + 1$
    end if

    for $i \leftarrow 0$ to $nMarkers - 1$ do
        // Is this a $0 \rightarrow 1$ transition, i.e., start of movement?
        if $((MOVE[k-1][i] = 0)$ and $(MOVE[k][i] = 1))$ then
            $delay \leftarrow frame[k].timestamp - motor[idx].timestamp$
            $hist$.AddValue($delay$)
            break // only one histogram update per frame is allowed
        end if
    end for
end for

// Threshold the histogram at 50% of the peak value.
$maxCount \leftarrow hist$.GetMaxBinCount()
$threshold \leftarrow maxCount/2.0$
$hist$.Threshold($threshold$)

efferent-afferent-delay $\leftarrow hist$.GetMean()
return efferent-afferent-delay
```

Algorithm 4 Histogram Code

```
InitHistogram($min, max, nBins$)
// To simplify the code two additional bins are reserved: bin[0] and bin[nBins+1].
// These bins hold values (if any) that are less than min or greater than max.
$bin \leftarrow$ new $int[nBins + 2]$ // indexes are: 0,1. ..., nBins, (nBins+1)
$limit \leftarrow$ new $double[nBins + 2]$ // indexes are: 0,1. ..., nBins, (nBins+1)

// Initialize the bin boundaries
$step \leftarrow (max - min)/nBins$
$limit[0] \leftarrow min$
for $i \leftarrow 1$ to $nBins - 1$ do
    $limit[i] \leftarrow limit[i - 1] + step$
end for
$limit[nBins] \leftarrow max$
$limit[nBins + 1] \leftarrow MAXDOUBLE$

for $i \leftarrow 0$ to $nBins + 1$ do
    $bin[i] \leftarrow 0$
end for
```

```
AddValue($value$)
for $b \leftarrow 0$ to $nBins + 1$ do
    if $value < limit[b]$ then
        $bin[b] \leftarrow bin[b] + 1$
        break
    end if
end for

```

```
GetMinBinCount()
$min \leftarrow MAXINT$
for $b \leftarrow 1$ to $nBins$ do
    if $bin[b] < min$ then
        $min \leftarrow bin[b]$
    end if
end for
return $min$
```

```
GetMaxBinCount()
$max \leftarrow MININT$
for $b \leftarrow 1$ to $nBins$ do
    if $bin[b] > max$ then
        $max \leftarrow bin[b]$
    end if
end for
return $max$
```

Algorithm 5 Histogram Code (continued)

```
THRESHOLD(threshold)
for $b \leftarrow 0$ to $nBins + 1$ do
    if $bin[b] < threshold$ then
        $bin[b] \leftarrow 0$
    else
        $bin[b] \leftarrow bin[b] - threshold$
    end if
end for

GETMEAN()
$count \leftarrow 0$
$sum \leftarrow 0$
for $b \leftarrow 1$ to $nBins$ do
    $count \leftarrow count + bin[b]$
    $sum \leftarrow sum + bin[b] * (limit[b-1] + limit[b])/2.0$
end for
return $sum/count$;

GETVARIANCE()
// This function is not used by Algorithm 3.
$count \leftarrow 0$
$sum \leftarrow 0$
$sum2 \leftarrow 0$
for $b \leftarrow 1$ to $nBins$ do
    $count \leftarrow count + bin[b]$
    $middle \leftarrow (limit[b-1] + limit[b])/2.0$
    $sum \leftarrow sum + bin[b] * middle$
    $sum2 \leftarrow sum2 + bin[b] * middle * middle$
end for

if $count \leq 1$ then
    return $0$
else
    return $(sum2 - (sum * sum)/count)/(count - 1)$
end if

GETSTDEV()
// This function is not used by Algorithm 3.
$variance \leftarrow$ GETVARIANCE()
return $\sqrt{variance}$
```

After all delays are measured the algorithm finds the bin with the largest count; this corresponds to the peak of the histogram. To reduce the effect of noisy histogram updates, the histogram is thresholded with an empirically derived threshold equal to 50% of the peak value. For example, if the largest bin count is 200 the threshold will be set to 100. After the histogram is thresholded, the mean delay is estimated by multiplying the bin count of each bin with its corresponding delay, then adding all products and dividing the sum by the total bin count (see the pseudocode for GetMean() in Algorithm 5).

There are two reasons why a histogram-based approach was selected. The first reason is that by keeping a histogram the algorithm only uses a fixed amount of memory. The alternative is to store all measured delays and calculate the mean using the entire data sample. This alternative approach, however, will not estimate the mean delay very accurately as the calculations will be biased by the values of outliers created by noisy readings. In order to eliminate the outliers, some sort of thresholding will be required which, in a way, will be equivalent to the histogram-based method.

The second reason for using a histogram-based method is related to findings that biological brains have a large number of neuron-based delay detectors specifically dedicated to measuring timing delays (Gallistel, 2003; Gibbon et al., 1997). Supposedly, these detectors are fine tuned to detect only specific timing delays. Thus, one way to think of the bins of the histogram is as a bank of detectors each of which is responsible for detecting only a specific timing delay.

The value of the mean delay by itself is not very useful, however, as it is unlikely that other measured delays will have the exact same value. As mentioned in Section 5.4, in order to classify the visual features as either "self" or "other" the measured delay for the feature must be within some tolerance interval around the mean. This interval was shown as the brown region in Figure 5.2. One way to determine this tolerance interval is to calculate the standard deviation of the measured delays, σ, and then classify a feature as "self" if its movement delay, d, lies within one standard deviation of the mean, μ. In other words, the feature is classified as "self" if $\mu - \sigma \leq d \leq \mu + \sigma$.

The standard deviation can also be calculated from the histogram (as shows in Algorithm 5). Because the histogram is thresholded, however, this estimate will not be very reliable as some delays that are not outliers will be eliminated. In this case, the standard deviation will be too small to be useful. On the other hand, if the histogram is not thresholded the estimate for the standard deviation will be too large to be useful as it will be calculated over the entire data sample which includes the outliers as well. Thus, the correct estimation of the standard deviation is not a trivial task. This is especially true when the robot is not the only moving object in the environment (see Figure 5.19).

Fortunately, the psychophysics literature provides an elegant solution to this problem. As it turns out, the discrimination abilities for timing delays in both animals and humans obey the so-called Weber's law (Triesman, 1963, 1964; Gibbon, 1977). This law is named after the German physician Ernst Heinrich Weber (1795-1878) who was one of the first experimental psychologists. Weber observed that the sensory discrimination abilities of humans depend on the magnitude of the stimulus that they are trying to discriminate against. The following example, based on hypothetical numbers, will be used to explain Weber's law.[1] Suppose that your friend asks you to lift a 10-kilogram weight and remember the effort that is involved. You lift the weight and notice the force with which it pulls down your hand. At the next trial your friend adds an additional 0.5 kilograms without your knowledge and asks you to lift the weight again and tell him if you can feel any difference. You lift the weight again but you can't feel any difference. On the third trial your friend adds an additional 0.5 kilograms and asks you to try again. Now you can feel the difference and you can confidently say that the new weight is heavier. Thus, it takes a weight difference of 1 kilogram before you can feel the difference. If the same experiment is performed with a starting weight of 20 kilograms you will find out that the additional weight has to be equal to 2 kilograms before you can tell the difference. In other words, the magnitude of the just noticeable difference (JND) is different in each case: 1 kilogram versus 2 kilograms. In both cases, however, the value of the JND is equal to 10% of the original weight.

[1] A similar example is given at
http://www.cis.rit.edu/people/faculty/montag/vandplite/pages/chap_3/ch3p1.html

Using mathematical notation Weber's law can be stated as:

$$\left|\frac{\Delta I}{I}\right| = c$$

where I represents the magnitude of some stimulus, ΔI is the value of the just noticeable difference (JND), and c is a constant that does not depend on the value of I. The fraction $\frac{\Delta I}{I}$ is known as the Weber fraction. The law implies that the difference between two signals is not detected if that difference is less than the Weber fraction.

Therefore, Weber's law can also be used to predict if the difference between two stimuli I and I' will be detected. The stimuli will be indistinguishable if the following inequality holds:

$$\left|\frac{I - I'}{I}\right| < c$$

where c is a constant that does not depend on the values of I and I'.

The robot experiments that are described later use a similar discrimination rule:

$$\left|\frac{\mu - d}{\mu}\right| < \beta$$

where μ is the mean efferent-afferent delay, d is the currently measured delay between a motor command and perceived visual movement, and β is a constant that does not depend on μ.

While the neural mechanisms behind Weber's law are still unknown there are numerous experiments that have shown that both animals and humans obey that law. In fact, the law applies to virtually all sensory discrimination tasks: distinction between colors and brightness (Aguilar and Stiles, 1954), distances, sounds, weights, and time (Triesman, 1963, 1964; Gibbon, 1977). Furthermore, in timing discrimination tasks the just noticeable difference is approximately equal to the standard deviation of the underlying timing delay, i.e., $\frac{\sigma}{\mu} = \beta$. Distributions with this property are know as scalar distributions because the standard deviation is a scalar multiple of the mean (Gibbon, 1977). This result has been used in some of the most prominent theories of timing interval learning, e.g., (Gibbon, 1977, 1991; Gibbon and Church, 1984; Gallistel and Gibbon, 2000).

Thus, the problem of how to reliably estimate the standard deviation of the measured efferent-afferent delay becomes trivial. The standard deviation is simply equal to a constant multiplied by the mean efferent-afferent delay, i.e., $\sigma = \beta\mu$. The value of the parameter β can be determined empirically. For timing discrimination tasks in pigeons its value has been estimated at 30%, i.e., $\frac{\sigma}{\mu} = 0.3$ (Catania, 1970, p. 22). Other estimates for different animals range from 10% to 25% (Triesman, 1964, p. 328). In the robot experiments described below the value of β was set to 25%.

It is worth mentioning that the robotic implementation of self-detection does not have to rely on Weber's law. After all, the robot is controlled by a computer that has an internal clock with precision to the nanosecond. However, the problem of reliably estimating the standard deviation of timing delays remains. Therefore, it was decided to use the implications of Weber's law in order to solve/eliminate this problem. The solution proposed here is biologically plausible and also computationally efficient.

The four subsections that follow describe the four different experimental conditions used to test the procedure for estimating the efferent-afferent delay of a robot from self-observation data (i.e., Algorithm 3). The four test conditions are summarized in Table 5.1.

Table 5.1: The four experimental conditions described in the next four subsections.

Test Condition	**Description**	**Section**
Single Robot	Ideal test conditions. The robot is the only moving object in the environment and the only object that has perceptual features.	Section 5.8.1
Single Robot and Static Background Features	The robot is still the only moving object in the environment but there are also static environmental features which the robot can detect.	Section 5.8.2
Two Robots: Uncorrelated Movements	The robot is no longer the only moving object. The movements of the second robot are independent of the movements of the first robot.	Section 5.8.3
Two Robots: Mimicking Movements	As before, there are two moving robots in the environment. The second robot, however, mimics the movements of the first robot.	Section 5.8.4

5.8.1 Experiments with a Single Robot

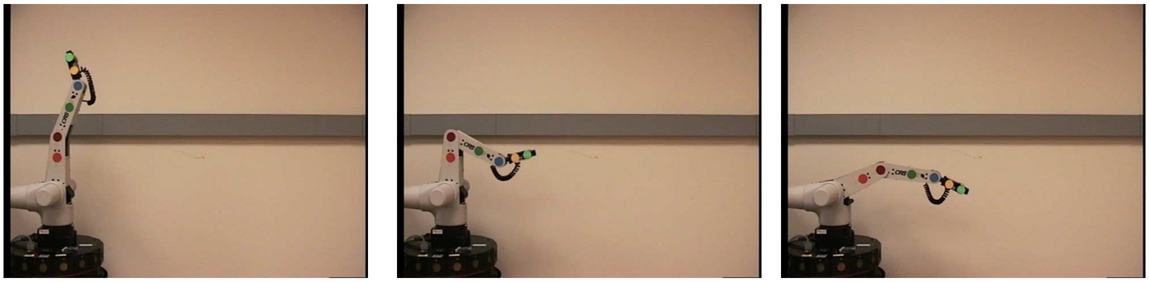


Figure 5.10: Frames from a test sequence in which the robot is the only moving object.

The first set of experiments tested the algorithm under ideal conditions when the robot is the only moving object in the environment (see Figure 5.10). Furthermore, the robot is the only object in the environment that has detectable visual features, i.e., color markers. The experimental data consists of two datasets which were collected by running the motor babbling procedure for 500 iterations. For each dataset the entire sequence of frames captured by the camera were converted to JPG files and saved to disk. The frames were recorded at 30 frames per second at a resolution of 640x480 pixels. Each dataset corresponds roughly to 45 minutes of wall clock time. This time limit was selected so that the data for one dataset can fit on a single DVD with storage capacity of 4.7GB. Each frame also has a timestamp denoting the time at which the frame was captured (more precisely, this timestamp indicates the time at which the frame was stored in the computer's memory after it was transfered from the capture card). The motor commands (along with their timestamps) were also saved as a part of the dataset. The results reported in this section are based on these two datasets.

Before the results are presented it is useful to take a look at the raw data gathered for each of the two datasets. Figure 5.11 shows a histogram for the measured efferent-afferent delays in dataset 1. Figure 5.12 shows the same representation for dataset 2. Each bin of these two histograms corresponds to $1/30^{-th}$ of a second which is equal to the time between two consecutive frames. As can be seen from these histograms, the average measured delay is approximately 1 second. This delay may seem relatively large but seems unavoidable since it is due to the slowness of the robot's controller. A robot with a faster controller may have a shorter delay. For comparison, the average efferent-afferent delay reported for a more advanced robot by Michel et al. (2004) was 0.5 seconds.

In Figures 5.11 and 5.12 the sum of the bin values is greater than 500 for both histograms, i.e., it is greater than the number of motor commands in each dataset. This is possible because for each motor command Algorithm 3 may increment the bin count for more than one bin. While the algorithm does not allow multiple bin updates for one and the same frame it is possible that two body markers may start to move one after another during two consecutive frames which will result in two histogram updates. In addition to that, there are a few false positive movements detected due to sensory noise which also contribute to this effect. Nevertheless, the histograms preserve their shape even if only one histogram update per motor command is allowed (see Figures 5.13 and 5.14).

The measured delays are also very consistent across different body markers. Figures 5.15 and 5.16 show the average measured delays for each of the six body markers as well as their corresponding standard deviations. As expected, all markers have similar delays and the small variations between them are not statistically significant.

Algorithm 3 estimated the following efferent-afferent delays for each of the two datasets: 1.02945 seconds (for dataset 1) and 1.04474 seconds (for dataset 2). The two estimates are very close to each other. The difference is less than $1/60^{-th}$ of a second, which is equivalent to half a frame (see the first column in Table 5.2). For comparison, Table 5.2 also shows the average efferent-afferent delays calculated using the raw delay data. As can be seen from the table, the second method slightly overestimates the peak of the histogram because its calculations are affected by noisy readings. Thus, Algorithm 3 offers a more reliable estimate of the efferent-afferent delay (see Figures 5.11 and 5.12).

Table 5.2: The mean efferent-afferent delay for dataset 1 and dataset 2 estimated using two different methods.

	Algorithm 3	**Raw Data**
Mean delay for dataset 1	1.02945	1.12823
Mean delay for dataset 2	1.04474	1.16045

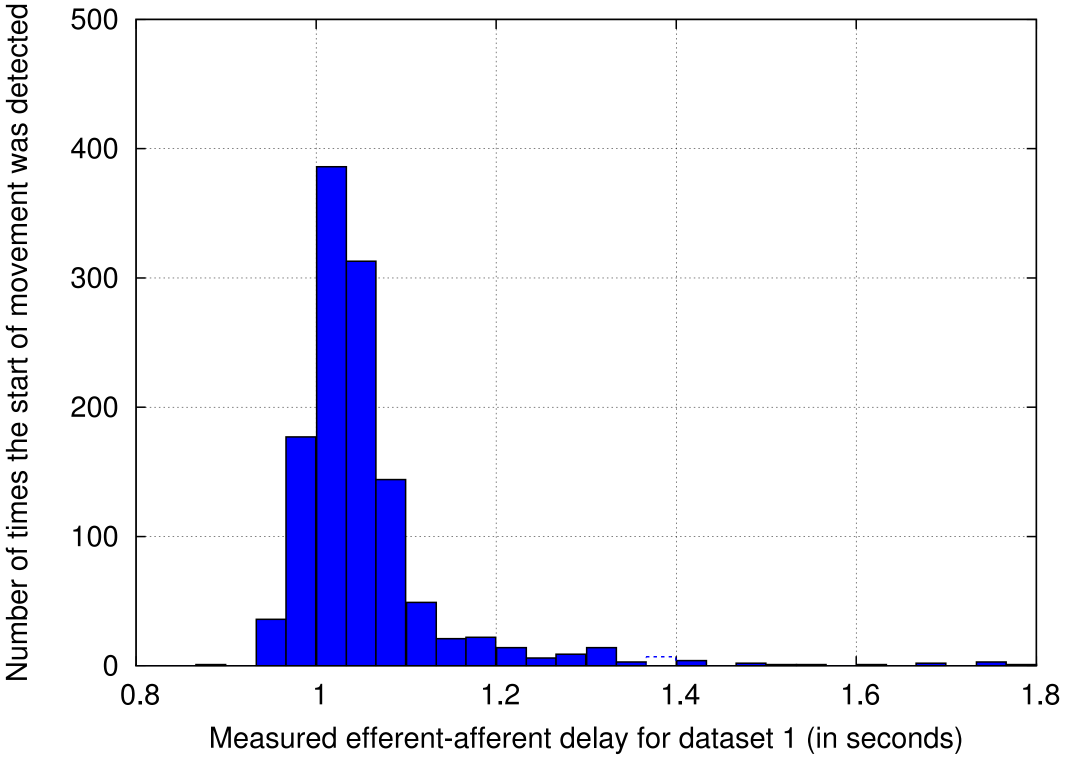


Figure 5.11: Histogram for the measured efferent-afferent delays in dataset 1.

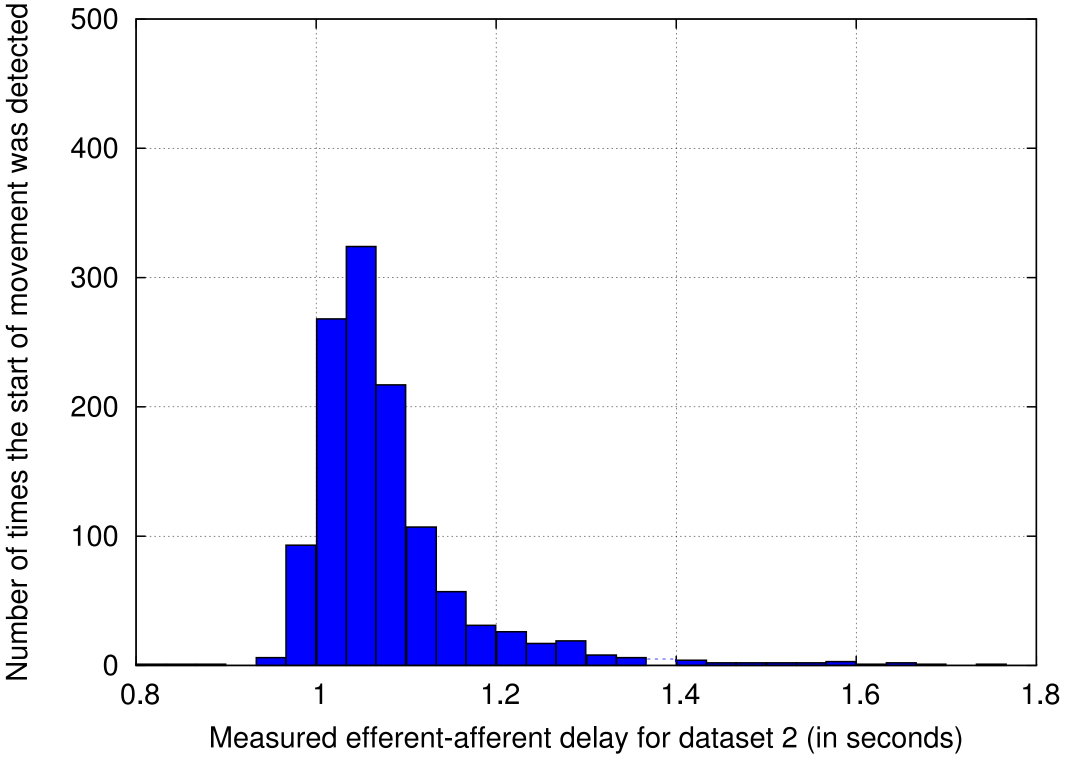


Figure 5.12: Histogram for the measured efferent-afferent delays in dataset 2.

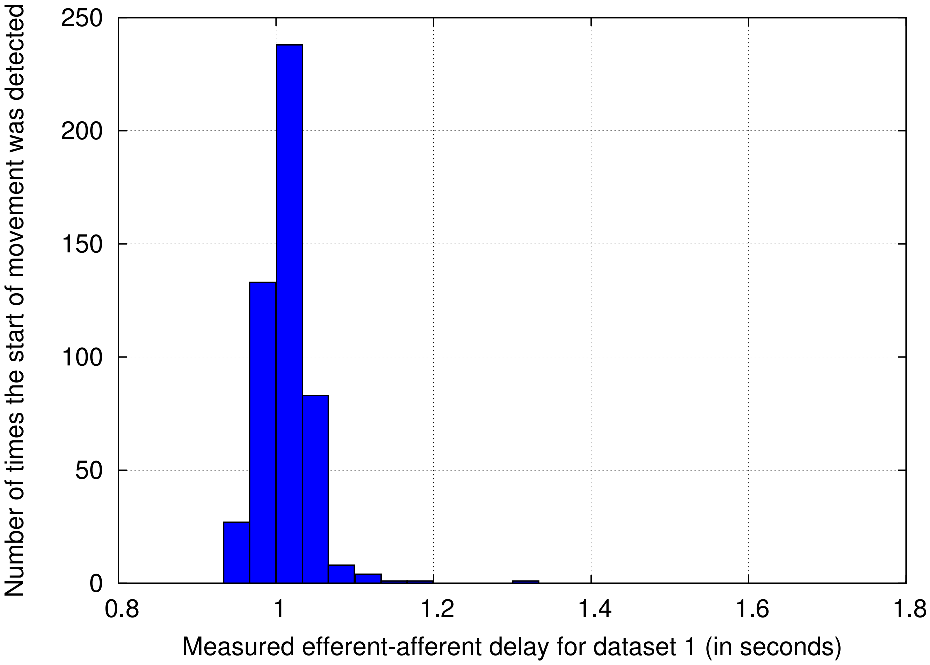


Figure 5.13: Histogram for the measured efferent-afferent delays in dataset 1. Unlike the histogram shown in Figure 5.11, the bins of this histogram were updated only once per motor command. Only the earliest detected movement after a motor command was used.

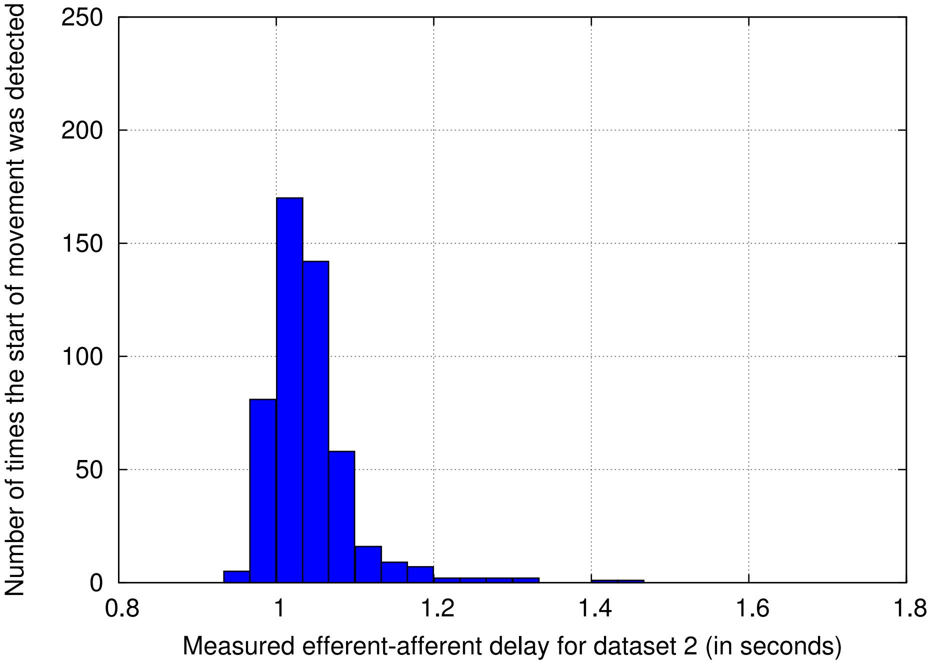


Figure 5.14: Histogram for the measured efferent-afferent delays in dataset 2. Unlike the histogram shown in Figure 5.12, the bins of this histogram were updated only once per motor command. Only the earliest detected movement after a motor command was used.

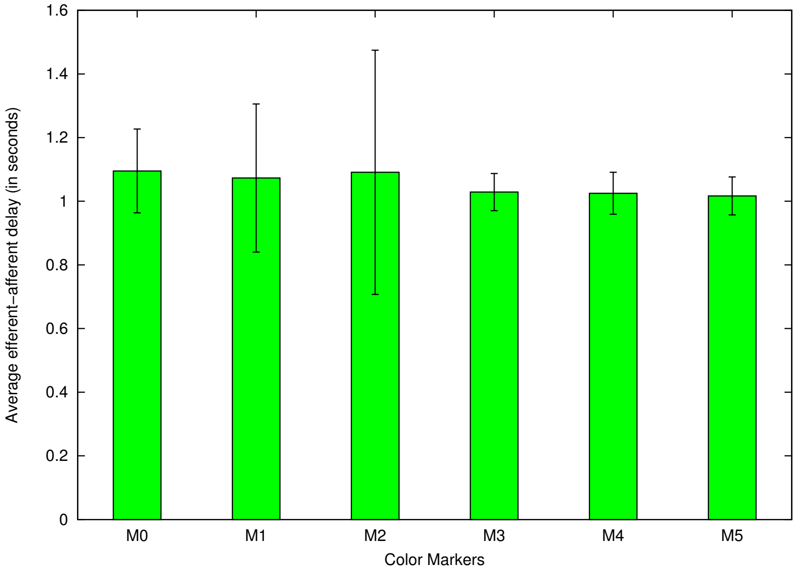


Figure 5.15: The average efferent-afferent delay and its corresponding standard deviation for each of the six body markers calculated using dataset 1.

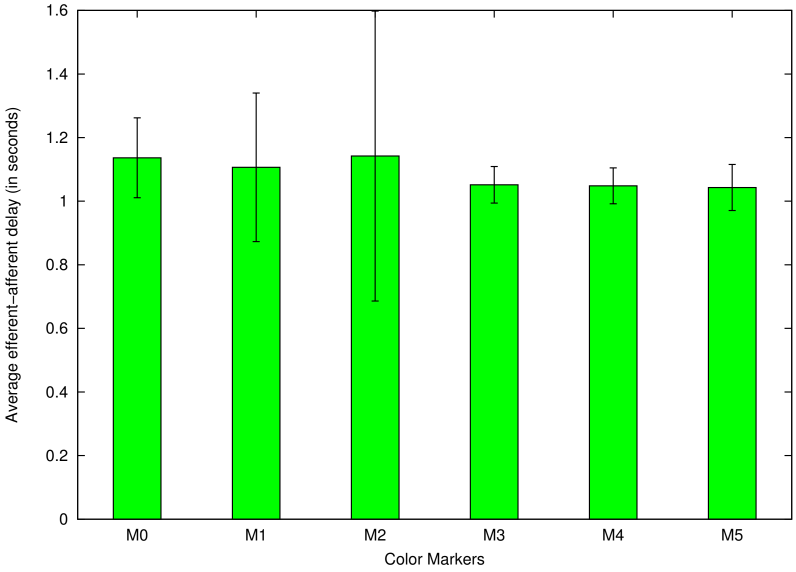


Figure 5.16: The average efferent-afferent delay and its corresponding standard deviation for each of the six body markers calculated using dataset 2.

5.8.2 Experiments with a Single Robot and Static Background Features

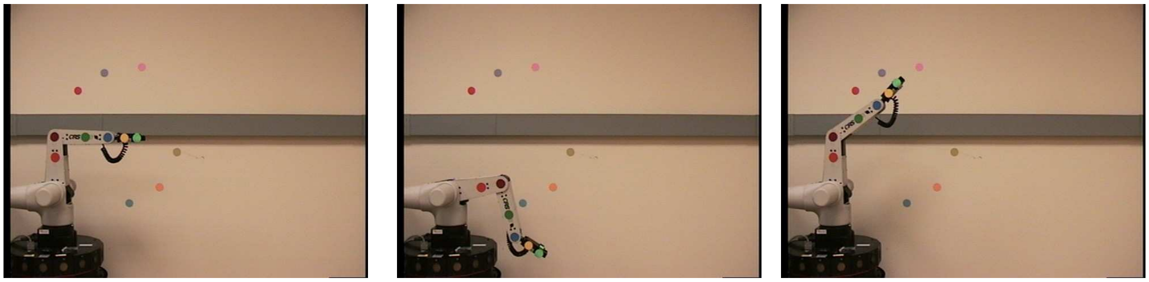


Figure 5.17: Frames form a test sequence with six static background markers.

The second experimental setup tested Algorithm 3 in the presence of static visual features placed in the environment. In addition to the robot's body markers, six other markers were placed on the background wall (see Figure 5.17). All background markers remained static during the experiment, but it was possible for them to be occluded temporarily by the robot's arm. Once again, the robot was controlled using the motor babbling procedure. A new dataset with 500 motor commands was collected using the procedure described in Section 5.8.1.

Figure 5.18 shows the delay histogram for this dataset. This histogram is similar to the histograms shown in the previous subsection. One noticeable difference, however, is that almost all bins in this case have some values. This is due to the detection of false positive movements for the background markers that could not be filtered out by the box filter. This is one of the main reasons why it is necessary to threshold the histogram. A similar but even more pronounced effect will be observed in the next section in which the background markers are allowed to move.

Figure 5.18 shows that, these false positive movements exhibit an almost uniform distribution over the interval from 0 to 5 seconds. This is to be expected as they are not correlated with the motor commands of the robot. The drop off after 5 seconds is due to the fact that the robot executes a new motor command approximately every 5 seconds. Therefore, any false positive movements of the background markers that are detected after the 5-second interval will be associated with the next motor command.

Table 5.3 shows the mean efferent-afferent delay as well as the standard deviation measured in two different ways. As shown in the table, Algorithm 3 performs better than the method that uses the raw data which overestimates the value of the delay by 0.8 seconds. The second method also overestimates the value of the standard deviation, estimating it at 1.33165 seconds. As explained in Section 5.8, Algorithm 3 underestimates the value of the standard deviation. This is not a problem, however, as the decision curve for "self" versus "other" discrimination is constructed using Weber's law which requires only the mean to be estimated correctly. The discrimination threshold, β, was set to 25% of the mean.

Table 5.3: Two estimates for the mean and the standard deviation of the efferent-afferent delay in the dataset with background markers.

	Algorithm 3	**Raw Data**
Mean	1.03559	1.86150
Stdev	0.01891	1.33165

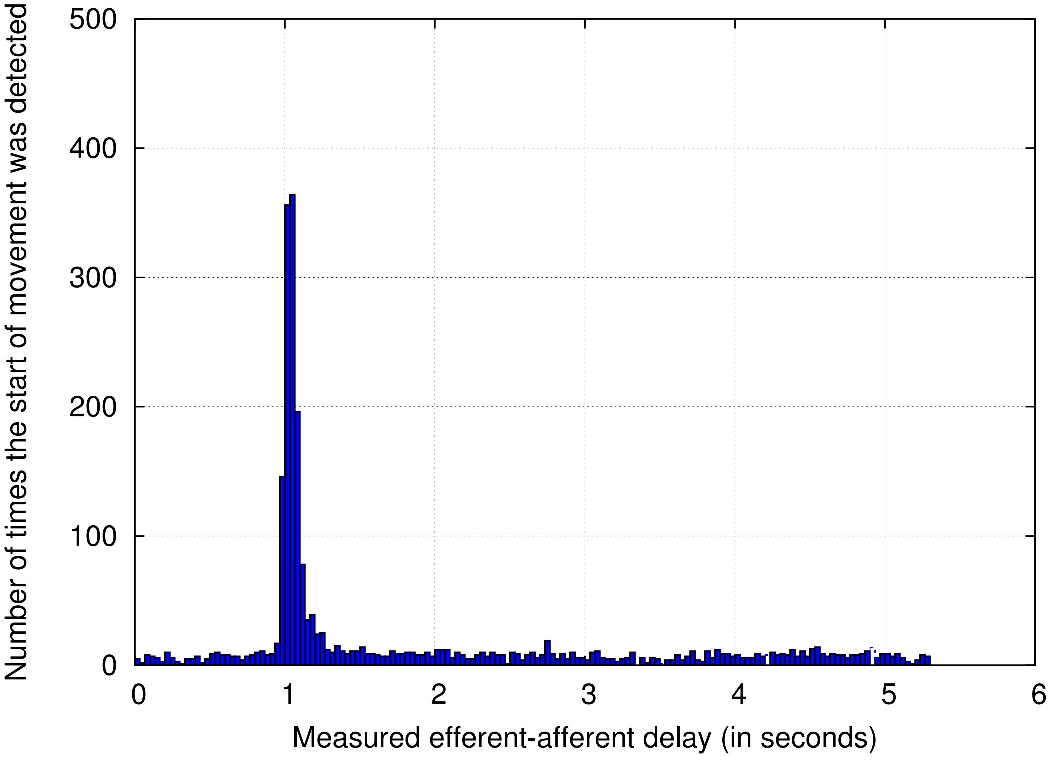


Figure 5.18: Histogram for the measured efferent-afferent delays for six robot and six static background markers (see Figure 5.17). Each bin corresponds to $1/30^{-th}$ of a second. Due to false positive movements detected for the background markers almost all bins of the histogram have some values. See text for more details.

5.8.3 Experiments with Two Robots: Uncorrelated Movements

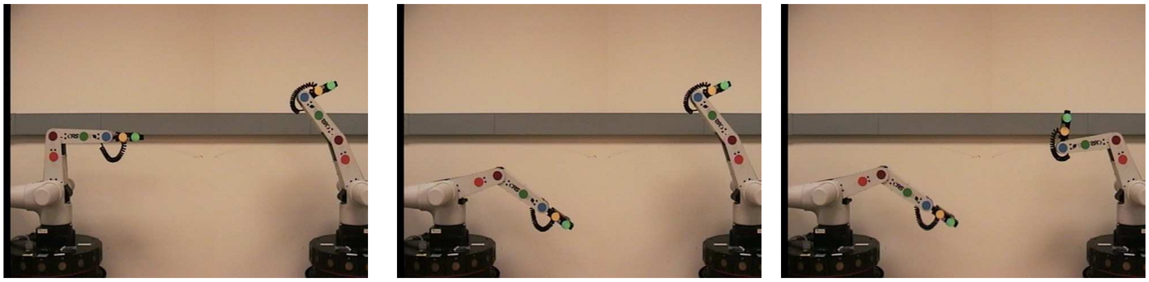


Figure 5.19: Frames from a test sequence with two robots in which the movements of the robots are uncorrelated. Each robot is controlled by a separate motor babbling routine. The robot on the left is the one trying to estimate its efferent-afferent delay.

Two sets of experiments were designed to test whether the robot can learn its efferent-afferent delay in situations in which the robot is not the only moving object in the environment. In this case, another moving object was introduced: another robot arm which was placed in the field of view of the first robot. The two test conditions vary in the degree of correlation between the movements of the two robots as follows: 1) uncorrelated movements; and 2) mimicking movements. For each of these test conditions a new dataset with 500 motor commands was generated. Figure 5.19 shows three frames from the dataset corresponding to the first test condition. The second test condition is described in the next subsection.

Before the experiments are described, however, a small technical clarification must made. Because there was only one robot available to perform these sets of experiments the second robot was generated using a digital video special effect. Each frame in which there are two robots is a composite of two other frames with only one robot in each (these frames were taken from the two datasets described in Section 5.8.1). The robot on the left is in the same position as in the previous datasets. In order to get the robot on the right, the left part of the second frame was cropped, flipped horizontally, translated and pasted on top of the right part of the first frame.

Similar experimental designs are quite common in self-detection experiments with infants (e.g., Watson (1994); Bahrick and Watson (1985)). In these studies the infants are placed in front of two TV screens. On the first screen the infants can see their own leg movements

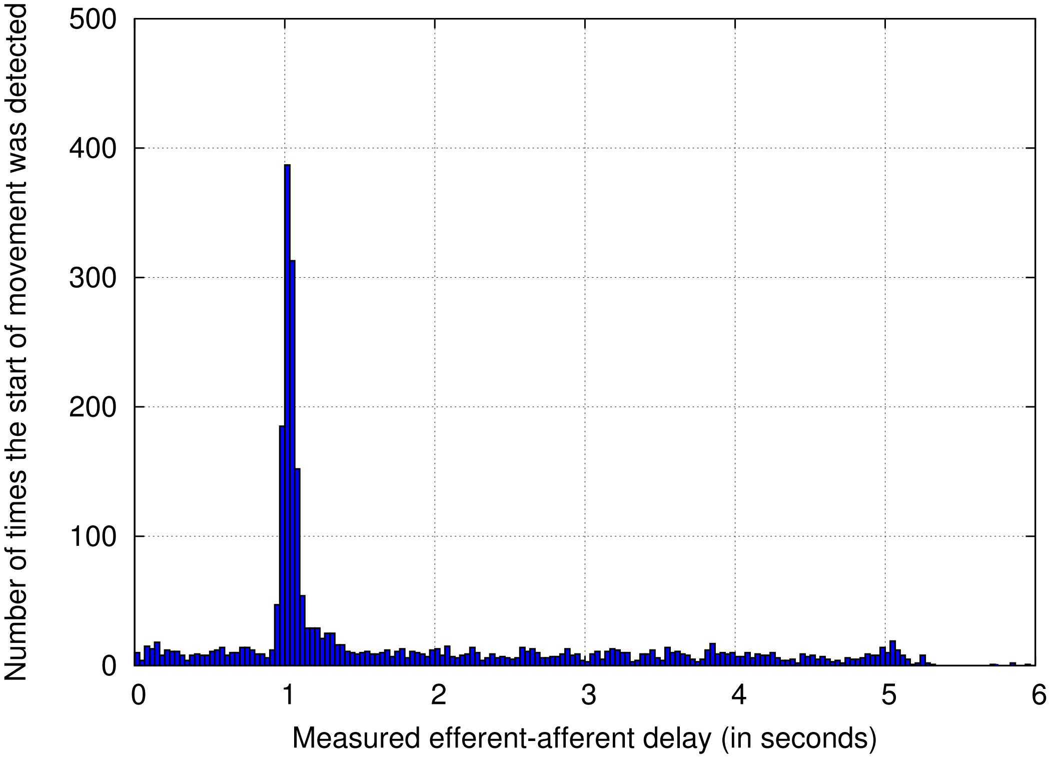


Figure 5.20: Histogram for the measured delays between motor commands and observed visual movements in the test sequence with two robots whose movements are uncorrelated (see Figure 5.19).

captured by a camera. On the second screen they can see the movements of another infant recorded during a previous experiment.

Figure 5.19 shows the first experimental setup with two robots. Under this test condition the movements of the two robots are uncorrelated. The frames for this test sequence were generated by combining the frames from dataset 1 and dataset 2 (described in Section 5.8.1). The motor commands and the robot on the left come from dataset 1; the robot on the right comes from dataset 2. Because the two motor babbling sequences have different random seed values the movements of the two robots are uncorrelated. For all frames the robot on the left is the one that is trying to estimate its efferent-afferent delay.

Figure 5.20 shows a histogram for the measured delays in this sequence. As can be seen from the figure, the histogram has some values for almost all of its bins. Nevertheless, there is still a clearly defined peak which has the same shape and position as in the previous test cases taken under ideal conditions. Algorithm 3 estimated the efferent-afferent delay

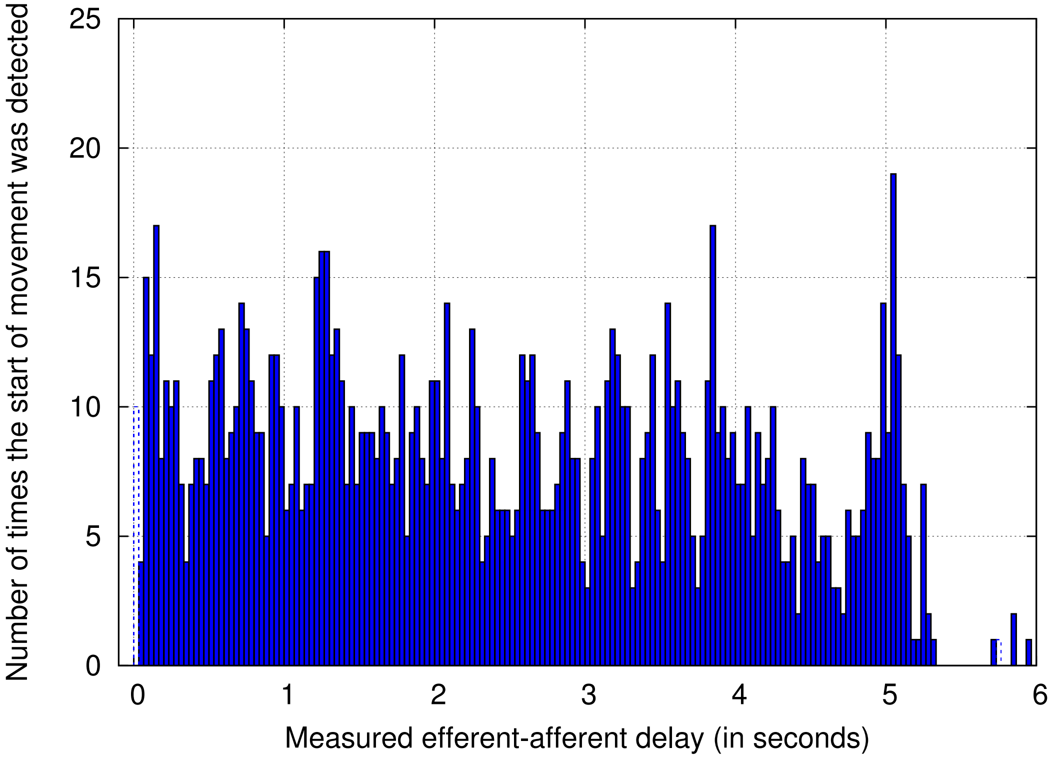


Figure 5.21: Contributions to the bins of the histogram shown in Figure 5.20 by the movements of the second robot only. This histogram shows that the movements of the second robot occur at all possible times after a motor command of the first robot. The drop off after 5 seconds is due to the fact that the first robot performs one motor command approximately every 5 seconds. Thus, any subsequent movements of the second robot after the 5-second interval are matched to the next motor command of the first robot.

at 1.02941 after the histogram was thresholded with a threshold equal to 50% of the peak value. For comparison, Table 5.4 shows that the mean delay estimated from the raw data is 1.82032, which is significantly overestimated (see Figure 5.20). The table also shows the value of the standard deviation calculated using the two methods. Once again, Algorithm 3 underestimates its value while the other method overestimates it.

Because the movements of the second robot are uncorrelated with the motor commands of the first robot the detected movements for the body markers of the second robot are scattered over all bins of the histogram. Thus, the movements of the second robot could not confuse the algorithm into picking a wrong value for the mean efferent-afferent delay. Figure 5.21 presents a closer look at the measured delays between the motor commands of the first robot and the movements of the body markers of the second robot. The histogram

Table 5.4: Two estimates for the mean and the standard deviation of the efferent-afferent delay in the dataset with two robots.

	Algorithm 3	**Raw Data**
Mean	1.02941	1.82032
Stdev	0.01622	1.34058

shows that, these movements exhibit almost an uniform distribution over the interval from 0 to 5 seconds. The drop off after 5 seconds is due to the fact that the the first robot performs a new movement approximately every 5 seconds. Therefore, any movements performed by the second robot after the 5-second interval will be associated with the next motor command of the first robot.

These results, once again, show that even when there are other moving objects in the environment it is possible for a robot to learn its own efferent-afferent delay. Even though there are instances in which the body markers of the second robot are detected to move with the perfect contingency, there are significantly more other instances in which they start to move either too early or too late. Because of this timing difference, the movements of any background object are represented as noise in the overall histogram.

To minimize the chance of imprinting on the wrong efferent-afferent delay, however, the developmental period during which the characteristic delay is learned can be increased. In fact, according to Watson (1994), this period lasts approximately three months in humans. For the current set of robot experiments, however, it was shown that 500 motor commands (or about 45 minutes of real time data) were sufficient to estimate this delay reliably for the test conditions described in this section.

The next subsection explores what happens if the independence condition between the two robots is violated and the second robot mimics the first one.

5.8.4 Experiments with Two Robots: Mimicking Movements

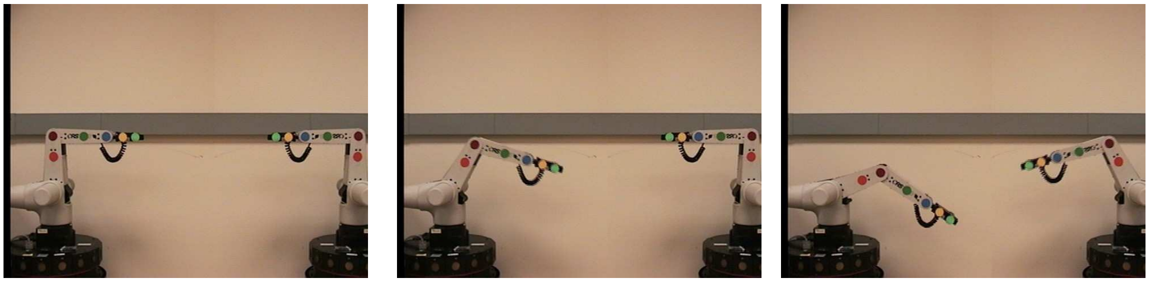


Figure 5.22: Frames from a test sequence with two robots in which the robot on the right mimics the robot on the left. The mimicking delay is 20 frames (0.66 seconds).

Under this test condition the second robot (the one on the right) is mimicking the first robot (the one on the left). The mimicking robot starts to move 20 frames (0.66 seconds) after the first robot. Another dataset of 500 motor commands was constructed using the frames of dataset 1 (described in Section 5.8.1) and offsetting the left and right parts of the image by 20 frames.

Because the mimicking delay is always the same, the resulting histogram (see Figure 5.23) is bimodal. The left peak, centered around 1 second, is produced by the body markers of the first robot. The right peak, centered around 1.7 seconds, is produced by the body markers of the second robot. Algorithm 3 cannot deal with situations like this and therefore it selects a delay which is between the two peaks (see Table 5.5). Calculating the mean delay from the raw data produces a mean estimate that is between the two peak values as well.

Table 5.5: Two estimates for the mean and the standard deviation of the efferent-afferent delay in the mimicking dataset with two robots.

	Algorithm 3	**Raw Data**
Mean	1.36363	1.44883
Stdev	0.334185	0.52535

It is possible to modify Algorithm 3 to avoid this problem by choosing the peak that corresponds to the shorter delay, for example. Evidence from animal studies, however, shows that when multiple time delays (associated with food rewards) are reinforced the animals learn “the mean of the reinforced distribution, not its lower limit” (Gibbon, 1977, p. 293), i.e., if the reinforced delays are generated from different underlying distributions the animals learn the mean associated with the mixture model of these distributions. Therefore, the algorithm was left unmodified.

Another reason to leave the algorithm intact exists: the mimicking test condition is a degenerate case which is highly unlikely to occur in any real situation in which the two robots are independent. Therefore, this negative result should not undermine the usefulness of Algorithm 3 for learning the efferent-afferent delay. The probability that two independent robots will perform the same sequence of movements over *an extended period of time* is effectively zero. Continuous mimicking for extended periods of time is certainly a situation which humans and animals never encounter in the real world.

The results of the mimicking robot experiments suggest an interesting study that can be conducted with monkeys provided that a brain for detecting and interpreting the signals from the motor neurons of an infant monkey were available. The decoded signals could then be used to send movement commands to a robot arm which would begin to move shortly after the monkey’s arm. If there is indeed an imprinting period, as Watson (1994) suggests, during which the efferent-afferent delay must be learned then the monkey should not be able to function properly after the imprinting occurs and the implant is removed.

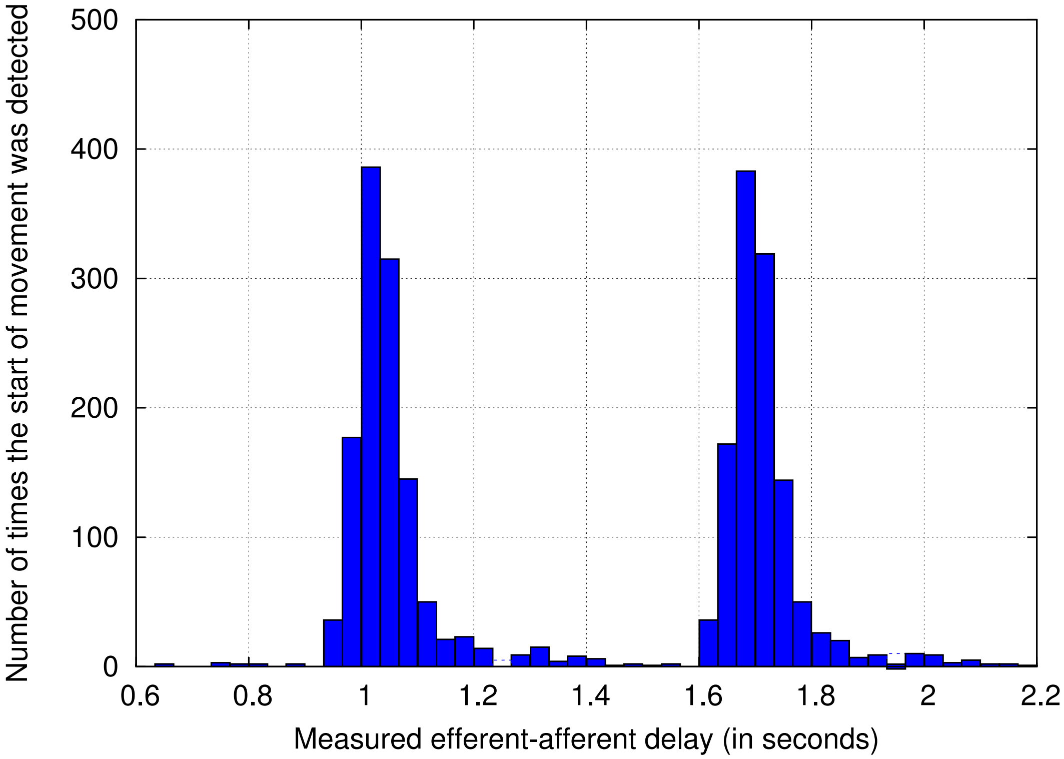


Figure 5.23: Histogram for the measured delays between motor commands and observed visual movements in the mimicking test sequence with two robots (see Figure 5.22). The left peak is produced by the movements of the body markers of the first robot. The right peak is produced by the movements of the body markers of the second/mimicking robot.

5.9 "Self" versus "Other" Discrimination

The previous section examined whether it is possible for a robot to learn its efferent-afferent delay from self-observation data (i.e., sub-problem 1 described in Section 5.4). This section examines whether the robot can use this delay to label the visual features that it detects as either "self" (i.e., belonging to the robot's body) or "other" (i.e., belonging to the external world). In other words, this section formulates and tests an approach to sub-problem 2 described in Section 5.4. The methodology for solving this problem was already described in Section 5.4 and its mathematical formulation was given in Section 5.3. The different experimental conditions used to test this approach and the obtained experimental results are described in Sections 5.9.1 - 5.9.4.

The basic methodology for performing this discrimination was already shown in Figure 5.2. In the concrete implementation, the visual field of view of the robot is first segmented into features and then their movements are detected using the method described in Section 5.7. For each feature the robot maintains two independent probabilistic estimates which jointly determine how likely it is for the feature to belong to the robot's own body.

The two probabilistic estimates are the *necessity index* and the *sufficiency index* as described in (Watson, 1985, 1994). The necessity index measures whether the feature moves consistently after every motor command. The sufficiency index measures whether for every movement of the feature there is a corresponding motor command that precedes it. The formulas for these two probabilities are given below.

$$\text{Necessity index} = \frac{\text{Number of temporally contingent movements}}{\text{Number of motor commands}}$$

$$\text{Sufficiency index} = \frac{\text{Number of temporally contingent movements}}{\text{Number of observed movements for this feature}}$$

Figure 5.24 shows an example with three visual features and their calculated necessity and sufficiency indexes. After two motor commands feature 1 (red) has a necessity index of 0.5 (1 contingent movement/2 motor commands) and a sufficiency index of 0.5 (1 contingent movement/2 observed movements). Feature 2 (green) has a necessity index of 1.0 (2 contingent movements/2 motor commands) but its sufficiency index is only 0.5 (2 contingent

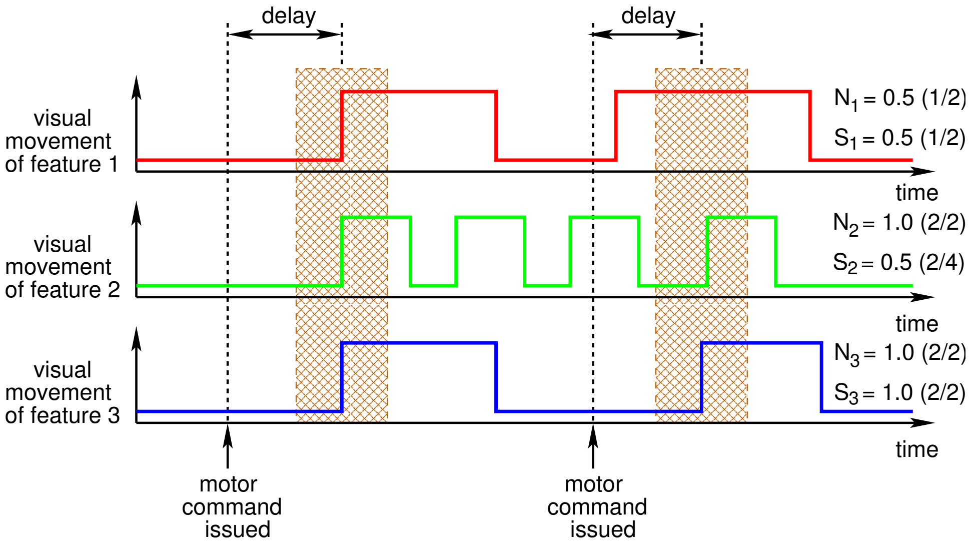


Figure 5.24: The figure shows the calculated values of the necessity (N_i) and sufficiency (S_i) indexes for three visual features. After two motor commands, feature 1 is observed to move twice but only one of these movements is contingent upon the robot's motor commands. Thus, feature 1 has a necessity $N_1 = 0.5$ and a sufficiency index $S_1 = 0.5$. The movements of feature 2 are contingent upon both motor commands (thus N_2=1.0) but only two out of four movements are temporally contingent (thus S_2=0.5). Finally, feature 3 has both N_3 and S_3 equal to 1.0 as all of its movements are contingent upon the robot's motor commands.

movements/4 observed movements) as only half of its movements are contingent. Finally, feature 3 (blue) has a necessity index of 1.0 (2 contingent movements/2 motor commands) and a sufficiency index of 1.0 (2 contingent movements/2 observed movements). Based on these results the robot can classify feature 3 as "self" because both its necessity and sufficiency indexes are equal to 1. Features 1 and 2 can be classified as "other."

For each feature, f_i, the robot maintains a necessity index, N_i, and a sufficiency index, S_i. The values of these indexes at time t are given by $N_i(t)$ and $S_i(t)$. Following Figure 5.24, the values of these indexes can be calculated by maintaining 3 counters: $C_i(t)$, $M_i(t)$, and $T_i(t)$. Their definitions are as follows: $C_i(t)$ represents the number of motor commands executed by the robot from some start time t_0 up to the current time t. $M_i(t)$ is the number of observed movements for feature f_i from time t_0 to time t; and $T_i(t)$ is the number of temporally contingent movements observed for feature f_i up to time t. The first two counters are trivial to calculate. The third counter, $T_i(t)$, is incremented every time the feature f_i is detected to move (i.e., when $M_i(t) = 1$ and $M_i(t-1) = 0$) and the movement delay relative to the last motor command is approximately equal to the mean efferent-afferent delay plus or minus some tolerance interval. In other words,

$$T_i(t) = \begin{cases} T_i(t-1)+1 & : \text{ if } M_i(t)=1 \textbf{ and } M_i(t-1)=0 \textbf{ and } \left|\dfrac{\mu - d_i}{\mu}\right| < \beta \\ T_i(t-1) & : \text{ otherwise} \end{cases}$$

where μ is the estimate for the mean efferent-afferent delay; d_i is the delay between the currently detected movement of feature f_i and the last motor command; and β is a constant. The value of β is independent from both μ and d_i and is equal to Weber's fraction (see Section 5.8). The inequality in this formula essentially defines the width of the decision region (see the brown regions in Figure 5.24).

Using this notation the values of the necessity and sufficiency indexes at time t can be calculated as follows:

$$\begin{aligned} N_i(t) &= \frac{T_i(t)}{C_i(t)} \\ S_i(t) &= \frac{T_i(t)}{M_i(t)} \end{aligned}$$

Both of these indexes are updated over time as new evidence becomes available, i.e., after a new motor command is issued or after the feature is observed to move. The belief of the robot that f_i is part of its body at time t is given jointly by $N_i(t)$ and $S_i(t)$. If the robot has to classify feature f_i it can threshold these values; if both are greater than the threshold value, α, the feature f_i is classified as "self". In other words,

$$f_i \in \begin{cases} F_{self} & : \text{ if and only if } N_i(t) > \alpha \text{ and } S_i(t) > \alpha \\ F_{other} & : \text{ otherwise} \end{cases}$$

Ideally, both $N_i(t)$ and $S_i(t)$ should be 1. In practice, however, this is rarely the case as there is always some sensory noise that cannot be filtered out. Therefore, for all robot experiments the threshold value, α, was set to an empirically derived value of 0.75.

The subsections that follow test this approach for "self" versus "other" discrimination in a number of experimental situations. In this set of experiments, however, it is assumed that the robot has already estimated its efferent-afferent delay and is only required to classify the features as either “self” or “other” using this delay.

These test situations are the same as the ones described in the previous section. For all experiments that follow the value of the mean efferent-afferent delay was set to 1.035. This value is equal to the average of the four means (1.02945, 1.04474, 1.03559, 1.02941) calculated for the four datasets described in Sections 5.8.1-5.8.3 (rounded up to three decimal points). The value of β was set to 0.25. Thus, a visual movement will be classified as temporally contingent to the last motor command if the measured delay is between 0.776 and 1.294 seconds.

The four subsections that follow describe the four different experimental conditions used to test the procedure for self/other discrimination described above. The four test conditions are summarized in Table 5.6. These are the same test conditions as the one described in Section 5.8. In this case, however, the goal of the robot is not to learn its efferent-afferent delay. Instead, the goal is to classify different visual features as either “self” or “other.”

Table 5.6: The four experimental conditions described in the next four subsections.

Test Condition	**Description**	**Section**
Single Robot	Ideal test conditions. The robot is the only moving object in the environment and the only object that has perceptual features.	Section 5.9.1
Single Robot and Static Background Features	The robot is still the only moving object in the environment but there are also static environmental features which the robot can detect.	Section 5.9.2
Two Robots: Uncorrelated Movements	The robot is no longer the only moving object. The movements of the second robot are independent of the movements of the first robot.	Section 5.9.3
Two Robots: Mimicking Movements	As before, there are two moving robots in the environment. The second robot, however, mimics the movements of the first robot.	Section 5.9.4

5.9.1 Experiments with a Single Robot

The test condition here is the same as the one described in Section 5.8.1 and uses the same two datasets derived from 500 motor babbling commands each. In this case, however, the robot already has an estimate for its efferent-afferent delay (1.035 seconds) and is only required to classify the markers as either "self" or "other". Because the two datasets don't contain any background markers the robot should classify all markers as "self." The experiments show that this was indeed the case.

Figure 5.25 shows the value of the sufficiency index calculated over time for each of the six body markers in dataset 1. Figure 5.26 shows the same thing for dataset 2. As mentioned above, these values can never be equal to 1.0 for a long period of time due to sensory noise. Figures 5.25 and 5.26 demonstrate that the sufficiency indexes for all six markers in both datasets are greater than 0.75 (which is the value of the threshold α).

An interesting observation about these plots is that after the initial adaptation period (approximately 5 minutes) the values for the indexes stabilize and don't change much (see Figure 5.25 and Figure 5.26). This suggests that these indexes can be calculated over a running window instead of over the entire dataset with very similar results. The oscillations in the first 5 minutes of each trial (not shown) are due to the fact that all counters and index values initially start from zero. Also, when the values of the counters are relatively small (e.g., 1 to 10) a single noisy update for any counter results in large changes for the value of the fraction which is used to calculate a specific index (e.g., the difference between 1/2 and 1/3 is large but the difference between 1/49 and 1/50 is not).

Figure 5.27 shows the value of the necessity index calculated over time for each of the six markers in dataset 1. Figure 5.28 shows the same thing for dataset 2. The figures show that the necessity indexes are consistently above the 0.75 threshold only for body markers 4 and 5 (yellow and green). At first this may seem surprising; after all, the six markers are part of the robot's body and, therefore, should have similar values for their necessity indexes. The reason for this result is that the robot has three different joints which can be affected by the motor babbling routine (see Algorithm 1). Each motor command moves one of the three joints independently of the other joints. Furthermore, one or more of these motor commands can be executed simultaneously.

Thus, the robot has a total of 7 different types of motor commands. Using binary notation these commands can be labeled as: 001, 010, 011, 100, 101, 110, and 111. In this notation, 001 corresponds to a motor command that moves only the wrist joint; 010 moves only the elbow joint; and 111 moves all three joints at the same time. Note that 000 is not a valid command since it does not move any of the joints. Because markers 4 and 5 are located on the wrist they move for every motor command. Markers 0 and 1, however, are located on the shoulder and thus they can be observed to move only for four out of seven motor commands: 100, 101, 110, and 111. Markers 2 and 3 can be observed to move for 6 out of 7 motor commands (all except 001), i.e., they will have a necessity index close to of 6/7 which is approximately 0.85 (see Figure 5.27).

This example shows that the probability of necessity may not always be computed correctly as there may be several competing causes. In fact, this observation is well supported fact in the statistical inference literature (Pearl, 2000, p. 285). "Necessity causation is a concept tailored to a specific event under consideration (singular causation), whereas sufficient causation is based on the general tendency of certain event *types* to produce other event types."(Pearl, 2000, p. 285). This distinction was not made by Watson (1985, 1994) as he was only concerned with discrete motor actions (e.g., kicking or no kicking) and it was tacitly assumed that the infants always kick with both legs simultaneously.

While the probability of necessity may not be identifiable in the general case, it is possible to calculate it for each of the possible motor commands. To accommodate for the fact that the necessity indexes, $N_i(t)$, are conditioned upon the motor commands the notation is augmented with a superscript, m, which stands for one of the possible types of motor commands. Thus, $N_i^m(t)$ is the necessity index associated with feature f_i and calculated only for the m^{th} motor command at time t. The values of the necessity index for each feature, f_i can now be calculated for each of the m possible motor commands:

$$N_i^m(t) \quad = \quad \frac{T_i^m(t)}{C_i^m(t)}$$

where $C_i^m(t)$ is the total number of motor commands of type m performed up to time t;

and $T_i^m(t)$ is the number of movements for feature f_i that are temporally contingent to motor commands of type m. The calculation for the sufficiency indexes remains the same as before.

Using this change of notation, a marker can be classified as "self" at time t if the sufficiency index $S_i(t)$ is greater than α and there exists at least one type of motor command, m, such that $N_i^m(t) > \alpha$. In other words,

$$f_i \in \begin{cases} F_{self} & : \text{ if and only if } \exists m \; : \; N_i^m(t) > \alpha \text{ and } S_i(t) > \alpha \\ F_{other} & : \text{ otherwise} \end{cases}$$

Figure 5.29 shows the values of the necessity index for each of the six body markers calculated over time using dataset 1 and the new notation. Each graph in this figure shows 7 lines which correspond to one of the seven possible motor commands. As can be seen from the figure, for each marker there is at least one motor command, m, for which the necessity index $N_i^m(t)$ is greater than the threshold, $\alpha = 0.75$. Thus, all six markers are correctly classified as "self." Figure 5.30 displays similar results for dataset 2.

It is worth noting that the approach described here relies only on identifying which joints participate in any given motor command and which markers are observed to start moving shortly after this motor command. The type of robot movement (e.g., fast, slow, fixed speed, variable speed) and how long a marker is moving as a result of it does not affect the results produced by this approach. The following subsections test this approach under different experimental conditions.

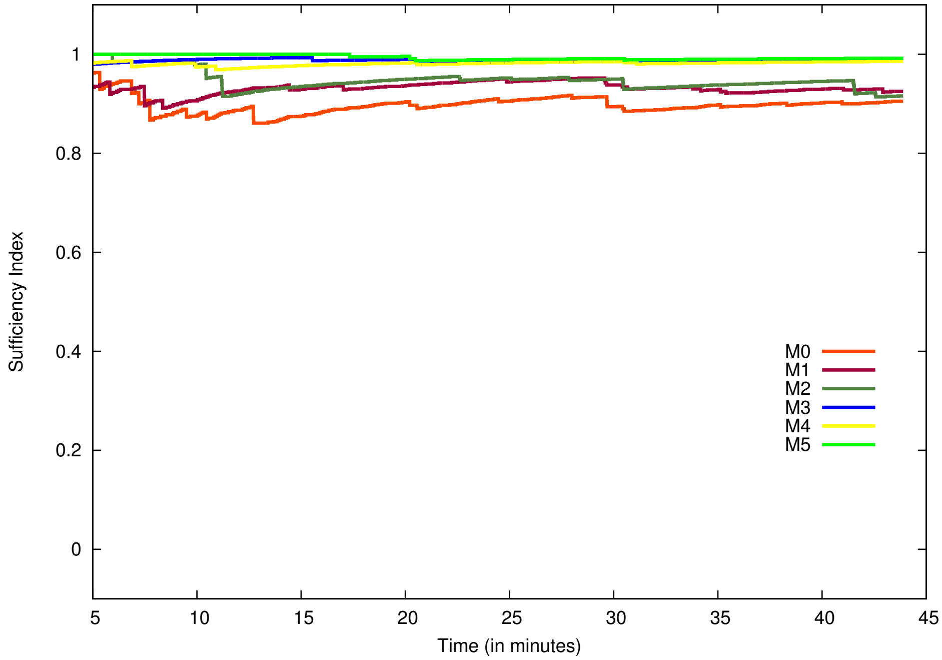


Figure 5.25: The figure shows the value of the sufficiency index calculated over time for the six body markers. The index value for all six markers is above the threshold $\alpha = 0.75$. The values were calculated using dataset 1.

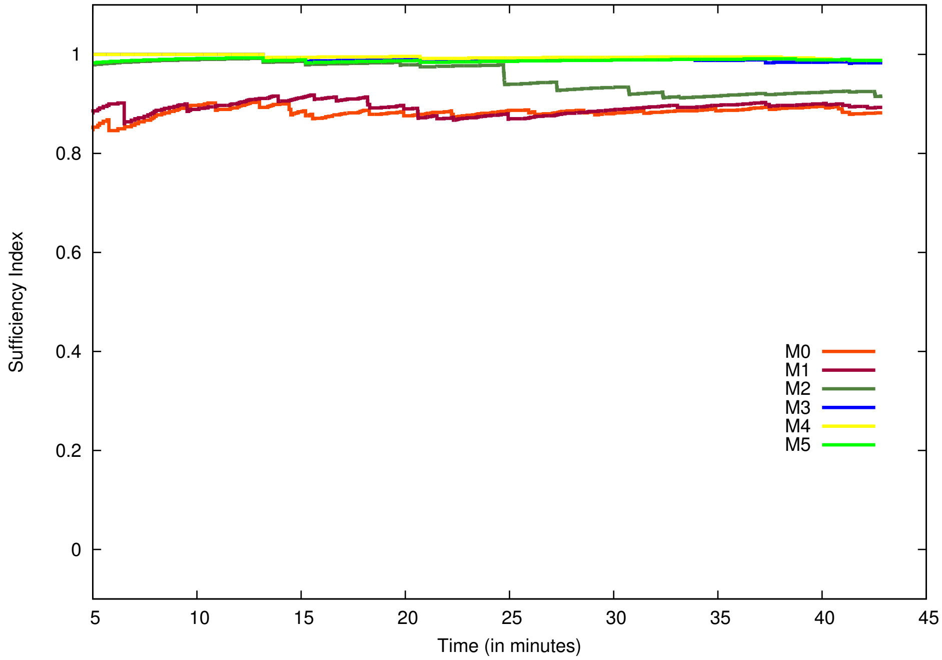


Figure 5.26: The figure shows the value of the sufficiency index calculated over time for the six body markers. The index value for all six markers is above the threshold $\alpha = 0.75$. The values were calculated using dataset 2.

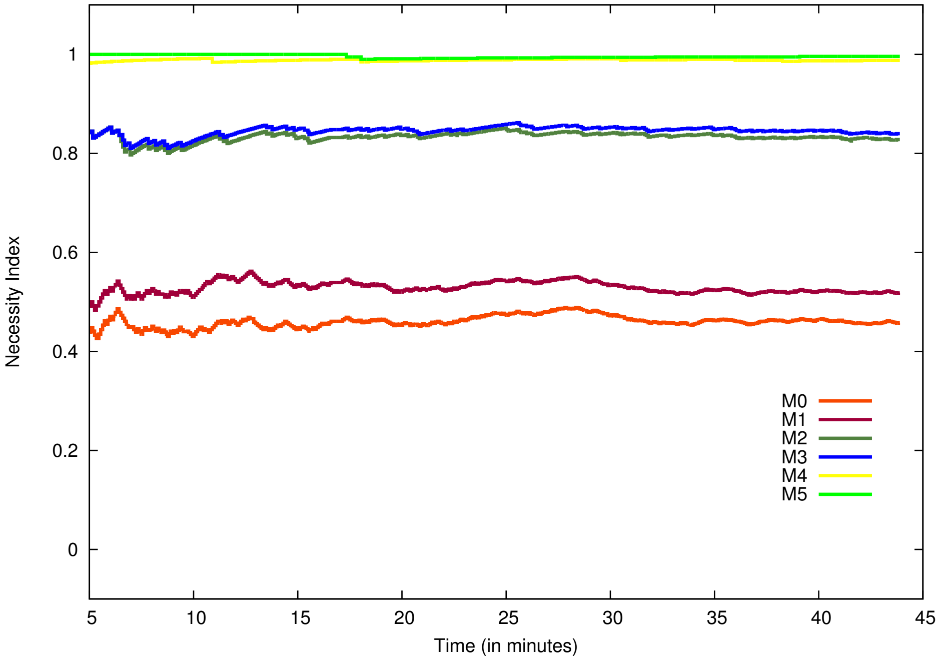


Figure 5.27: The value of the necessity index calculated over time for each of the six body markers in dataset 1. This calculation does not differentiate between the type of motor command that was performed. Therefore, not all markers can be classified as self as their index values are less than the threshold $\alpha = 0.75$ (e.g., M0 and M1). The solution to this problem is shown in Figure 5.29 (see text for more details).

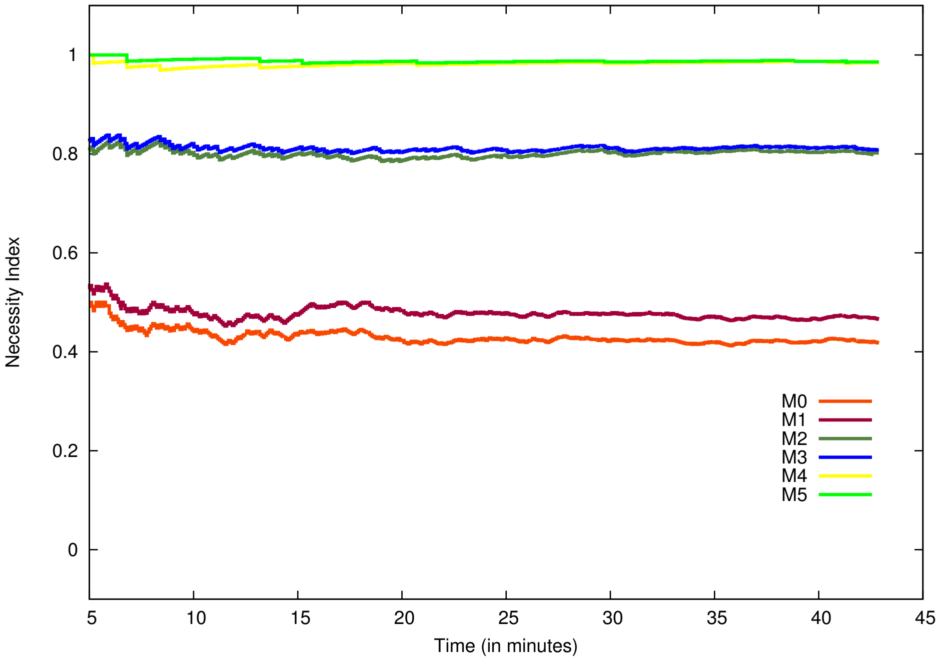
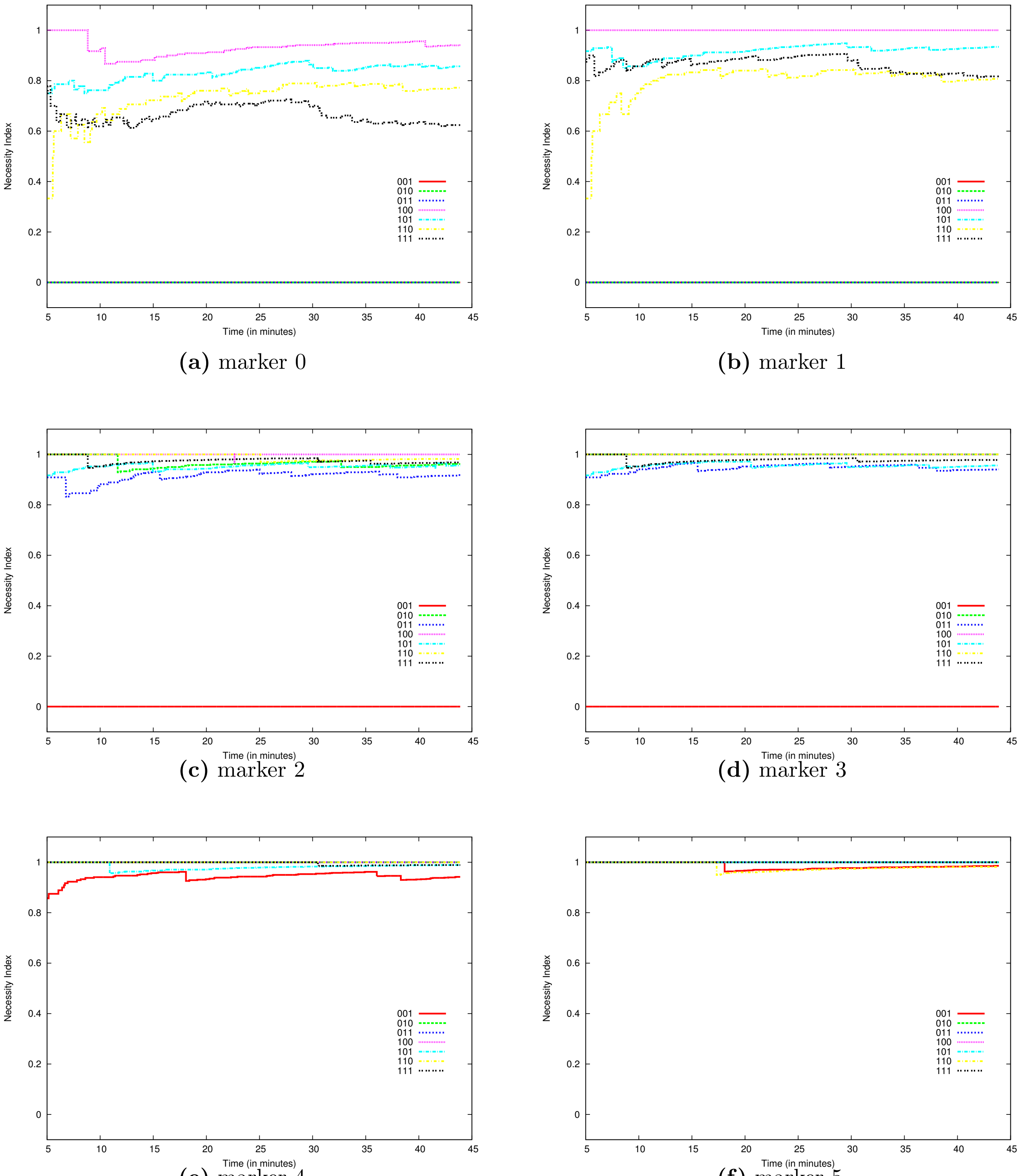


Figure 5.28: The value of the necessity index calculated for over time for each of the six body markers in dataset 2. This calculation does not differentiate between the type of motor command that was performed. Therefore, not all markers can be classified as self as their index values are less than the threshold $\alpha = 0.75$ (e.g., M0 and M1). The solution to this problem is shown in Figure 5.30 (see text for more details).



(a) marker 0

(b) marker 1

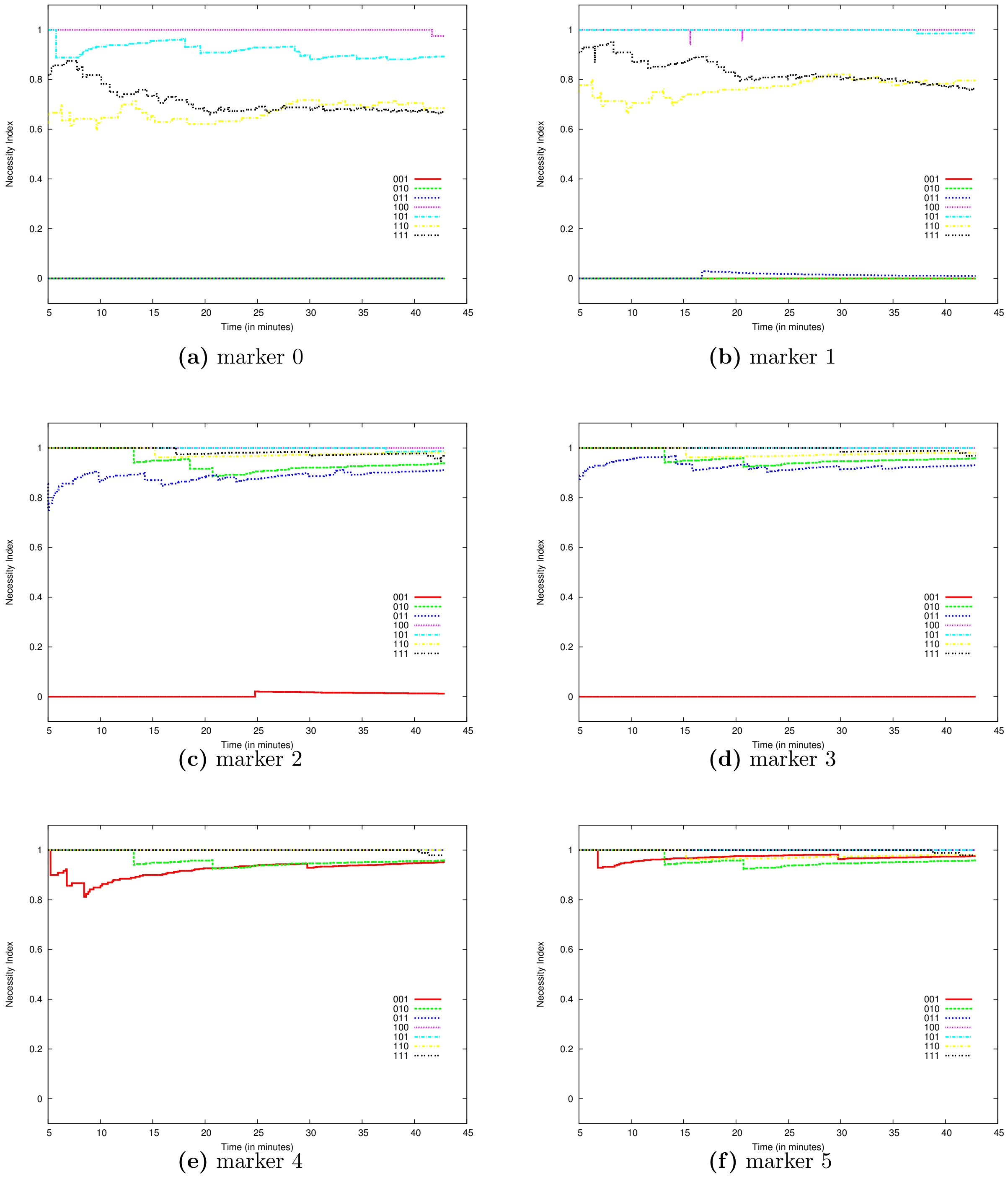
(c) marker 2

(d) marker 3

(e) marker 4

(f) marker 5

Figure 5.29: The figures shows the values of the necessity index, $N_i^m(t)$, for each of the six body markers (in dataset 1). Each figure shows 7 lines which correspond to one of the 7 possible types of motor commands: $001, \ldots, 111$. To be considered for classification as "self" each marker must have a necessity index $N_i^m(t) > 0.75$ for at least one motor command, m, at the end of the trial. All markers are classified as self in this dataset.



(a) marker 0

(b) marker 1

(c) marker 2

(d) marker 3

(e) marker 4

(f) marker 5

Figure 5.30: The figure shows the values of the necessity index, $N_i^m(t)$, for each of the six body markers (in dataset 2). Each figure shows 7 lines which correspond to one of the 7 possible types of motor commands: $001, \ldots, 111$. To be considered for classification as "self" each marker must have a necessity index $N_i^m(t) > 0.75$ for at least one motor command, m, at the end of the trial. All markers are classified as self in this dataset.

5.9.2 Experiments with a Single Robot and Static Background Features

This test condition is the same as the one described in Section 5.8.2. In addition to the robot's body markers six additional markers were placed on the background wall (see Figure 5.17). Again, the robot performed motor babbling for 500 motor commands. The dataset recorded for the purposes of Section 5.8.2 was used here as well.

Table 5.7 shows the classification results at the end of the test. The results demonstrate that there is a clear distinction between the two sets of markers: markers M0-M5 are classified correctly as "self." All background markers, M6-M11, are classified correctly as "other." The background markers are labeled clockwise starting from the upper left marker (red) in Figure 5.17. Their colors are: red (M6), violet (M7), pink (M8), tan (M9), orange (M10), light blue (M11).

Table 5.7: Values of the necessity and sufficiency indexes at the end of the trial. The classification for each marker is shown in the last column.

Marker	$\max_m(N_i^m(t))$	$S_i(t)$	**Threshold** α	**Classification**	**Actual**
M0	0.977	0.887	0.75	"self"	"self"
M1	1.000	0.943	0.75	"self"	"self"
M2	1.000	0.998	0.75	"self"	"self"
M3	1.000	0.995	0.75	"self"	"self"
M4	1.000	0.840	0.75	"self"	"self"
M5	1.000	0.996	0.75	"self"	"self"
M6	0.000	0.000	0.75	"other"	"other"
M7	0.068	0.140	0.75	"other"	"other"
M8	0.017	0.100	0.75	"other"	"other"
M9	0.057	0.184	0.75	"other"	"other"
M10	0.147	0.112	0.75	"other"	"other"
M11	0.185	0.126	0.75	"other"	"other"

The values for the sufficiency indexes calculated over time are shown in Figure 5.31 (body markers) and Figure 5.32 (background markers). The necessity indexes for each of the seven motor commands are shown in Figure 5.33 and Figure 5.34. All background markers (except marker 8) can be temporarily occluded by the robot's arm which increases their position tracking noise. This results in the detection of occasional false positive movements for these markers. Therefore, their necessity indexes are not all equal to zero (as it is the case with marker 8). Nevertheless, by the end of the trial the maximum necessity index for all background markers is less than 0.75 and, thus, they are correctly classified as "other."

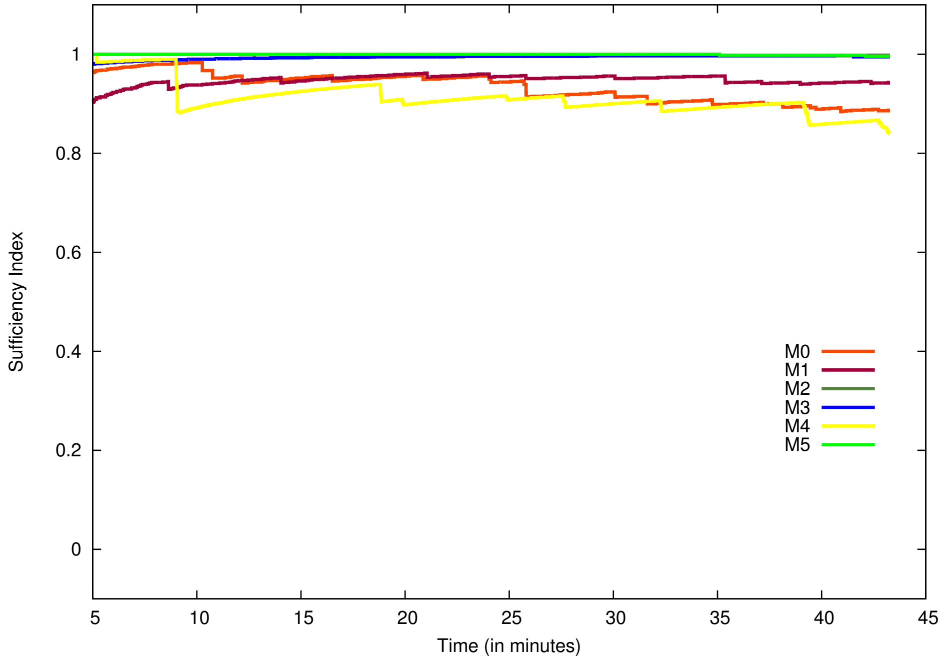


Figure 5.31: Sufficiency index for each of the six body markers. For all of these markers the index value is above the threshold $\alpha = 0.75$. The same is true for the necessity indexes as shown in Figure 5.33. Thus, all six body markers are classified as "self."

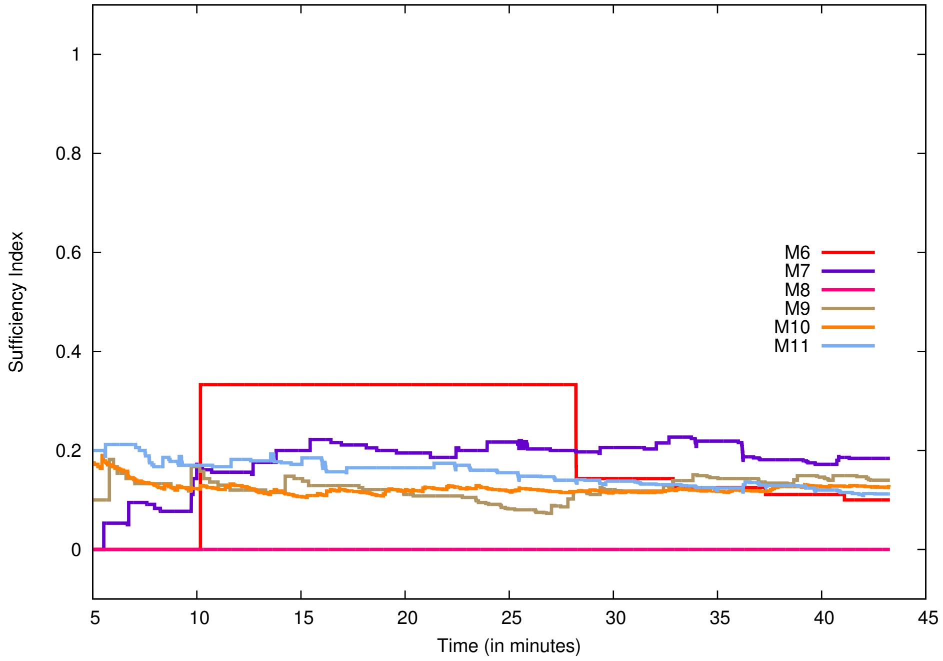
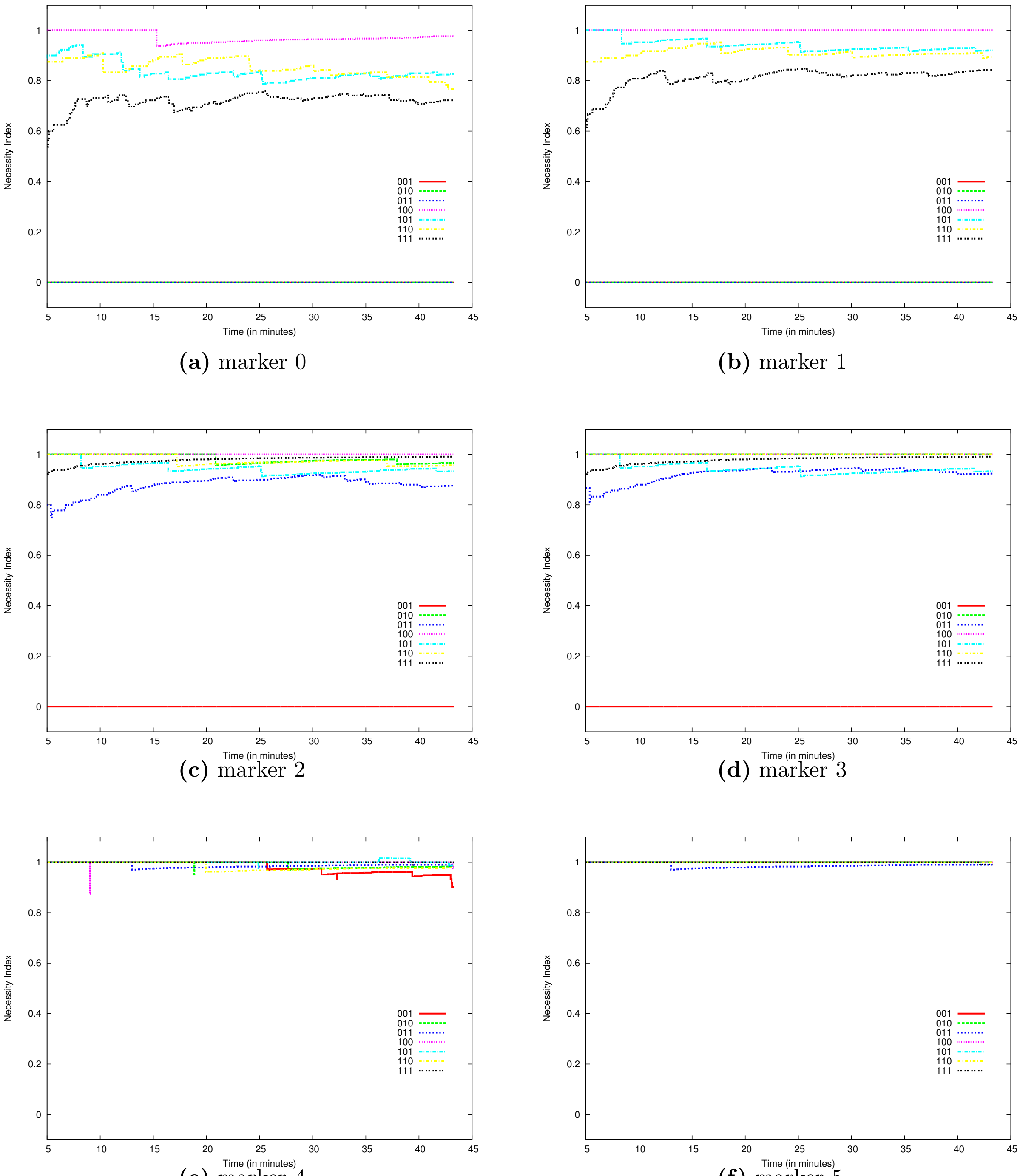


Figure 5.32: Sufficiency index for the six static background markers. For all of these markers the index value is below the threshold $\alpha = 0.75$. The same is true for the necessity indexes as shown in Figure 5.34. Thus, all six background markers are classified as "other."



(a) marker 0

(b) marker 1

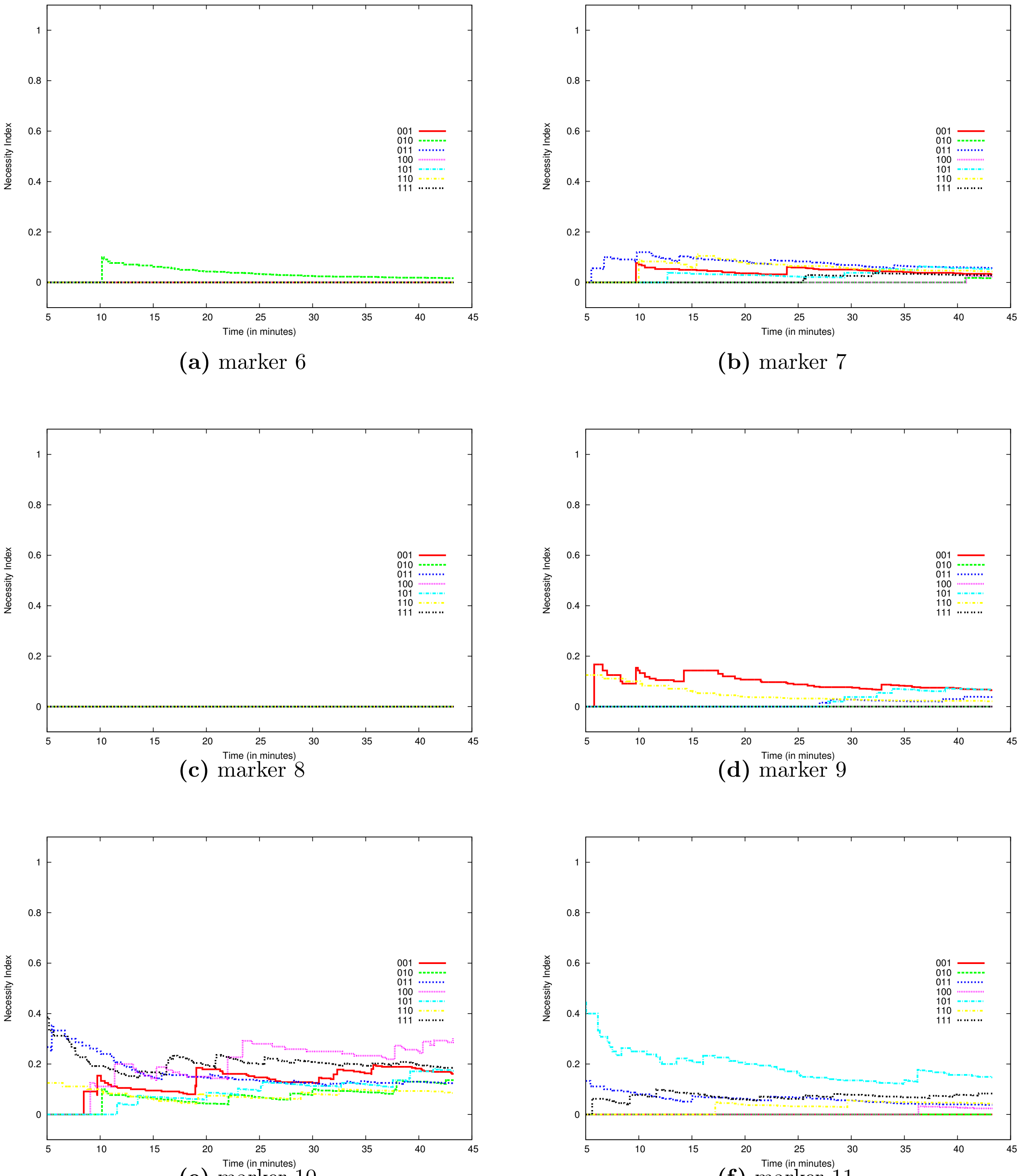
(c) marker 2

(d) marker 3

(e) marker 4

(f) marker 5

Figure 5.33: The necessity index, $N_i^m(t)$, for each of the six body markers. Each figure shows 7 lines which correspond to one of the 7 possible types of motor commands: $001, \ldots, 111$. To be considered for classification as "self" each marker must have a necessity index $N_i^m(t) > 0.75$ for at least one motor command, m, at the end of the trial. This is true for all body markers shown in this figure. Thus, they are correctly classified as "self."



(a) marker 6

(b) marker 7

(c) marker 8

(d) marker 9

(e) marker 10

(f) marker 11

Figure 5.34: The necessity index, $N_i^m(t)$, for each of the six background markers. Each figure shows 7 lines which correspond to one of the 7 possible motor commands: $001, \ldots, 111$. To be considered for classification as "self" each marker must have a necessity index $N_i^m(t) > 0.75$ for at least one motor command, m, at the end of the trial. This is not true for the background markers shown in this figure. Thus, they are all correctly classified as "other."

5.9.3 Experiments with Two Robots: Uncorrelated Movements

This experimental condition is the same as the one described in Section 5.8.3. The dataset recorded for the purposes of Section 5.8.3 was used here as well. If the self-detection algorithm works as expected only 6 of the 12 markers should be classified as "self" (markers M0-M5). The other six markers (M6-M11) should be classified as "other." Table 5.8 shows that the algorithm performs satisfactorily.

Figure 5.35 shows the sufficiency indexes for the six body markers of the first robot (i.e., the one trying to perform the self versus other discrimination – left robot in Figure 5.8.3). As expected, the index values are very close to 1. Figure 5.36 shows the sufficiency indexes for the body markers of the second robot. Since the movements of the second robot are not correlated with the motor commands of the first robot these values are close to zero.

Figure 5.37 shows the necessity indexes for each of the 6 body markers of the first robot for each of the seven motor commands. As expected, these indexes are greater than 0.75 for at least one motor command. Figure 5.38 shows the same for the markers of the second robot. In this case, the necessity indexes are close to zero. Thus, these markers are correctly classified as "other."

Table 5.8: Values of the necessity and sufficiency indexes at the end of the trial. All markers are classified correctly as "self" or "other."

Marker	$\max_m(N_i^m(t))$	$S_i(t)$	**Threshold** α	**Classification**	**Actual**
M0	0.941	0.905	0.75	"self"	"self"
M1	1.000	0.925	0.75	"self"	"self"
M2	1.000	0.912	0.75	"self"	"self"
M3	1.000	0.995	0.75	"self"	"self"
M4	1.000	0.988	0.75	"self"	"self"
M5	1.000	0.994	0.75	"self"	"self"
M6	0.066	0.102	0.75	"other"	"other"
M7	0.094	0.100	0.75	"other"	"other"
M8	0.158	0.110	0.75	"other"	"other"
M9	0.151	0.107	0.75	"other"	"other"
M10	0.189	0.119	0.75	"other"	"other"
M11	0.226	0.124	0.75	"other"	"other"

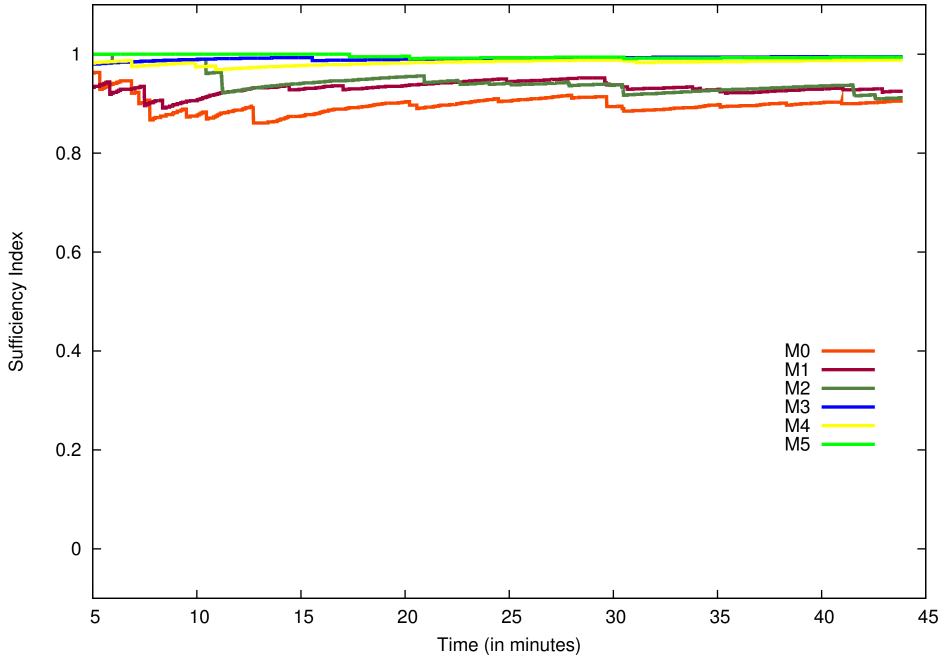


Figure 5.35: The figure shows the sufficiency indexes for each of the six body markers of the first robot (left robot in Figure 5.8.3). As expected these values are close to 1, and thus, above the threshold $\alpha = 0.75$. The same is true for the necessity indexes as shown in Figure 5.37. Thus, all markers of the first robot are classified as "self."

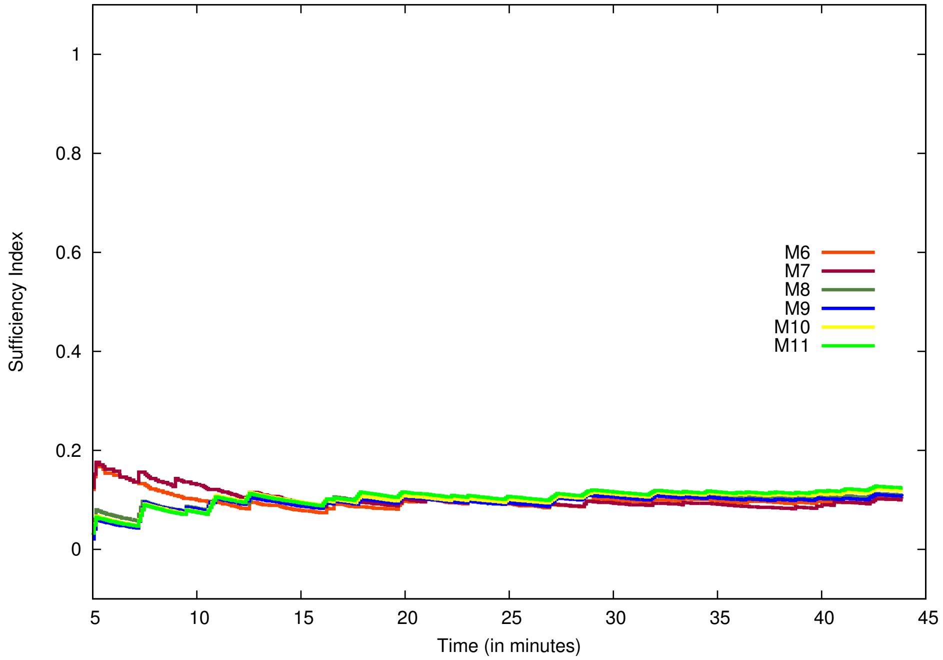
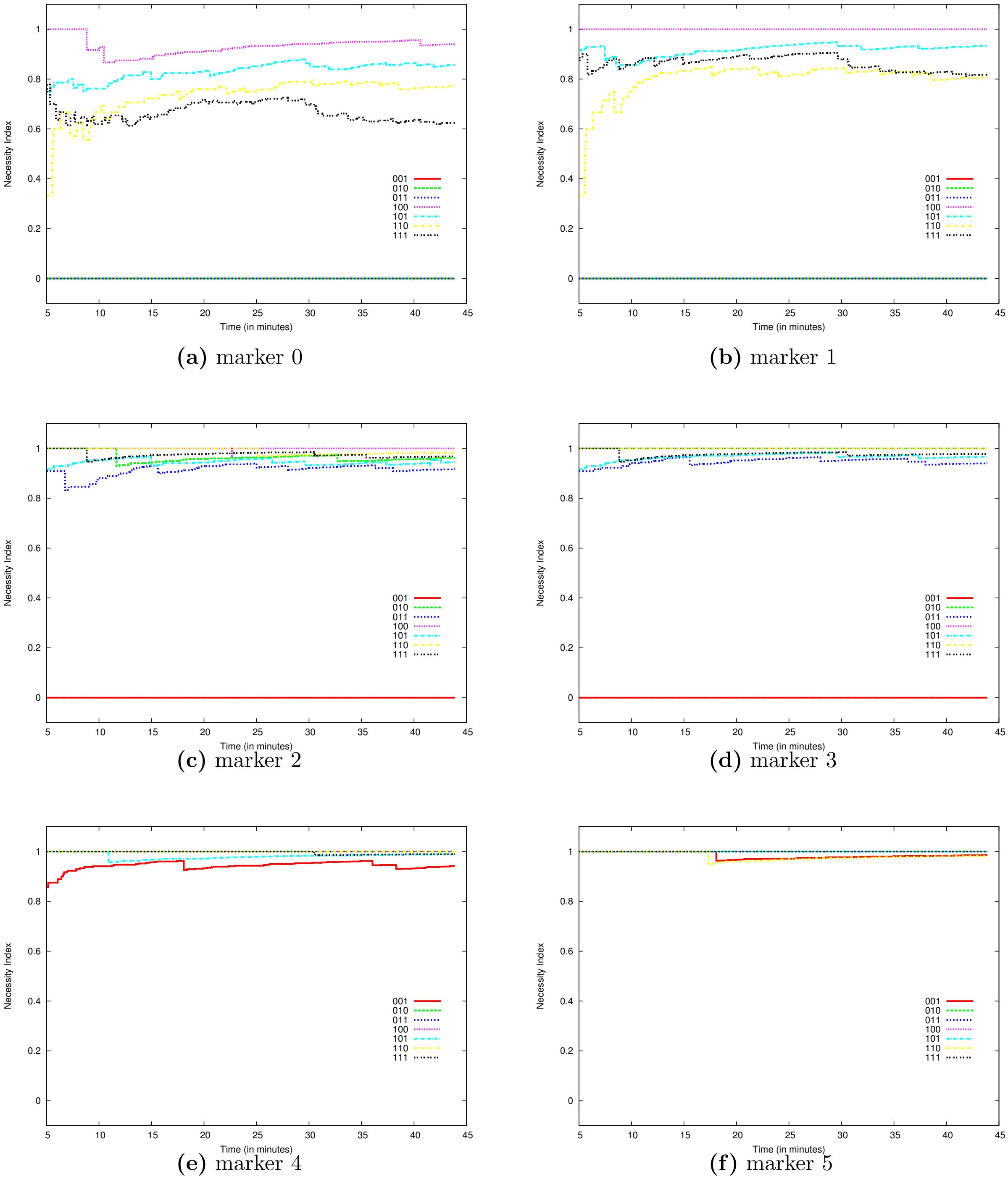


Figure 5.36: The figure shows the sufficiency indexes for each of the six body markers of the second robot (right robot in Figure 5.8.3). As expected these values are close to 0, and thus, below the threshold $\alpha = 0.75$. The same is true for the necessity indexes as shown in Figure 5.38. Thus, the markers of the second robot are classified as "other."



(a) marker 0

(b) marker 1

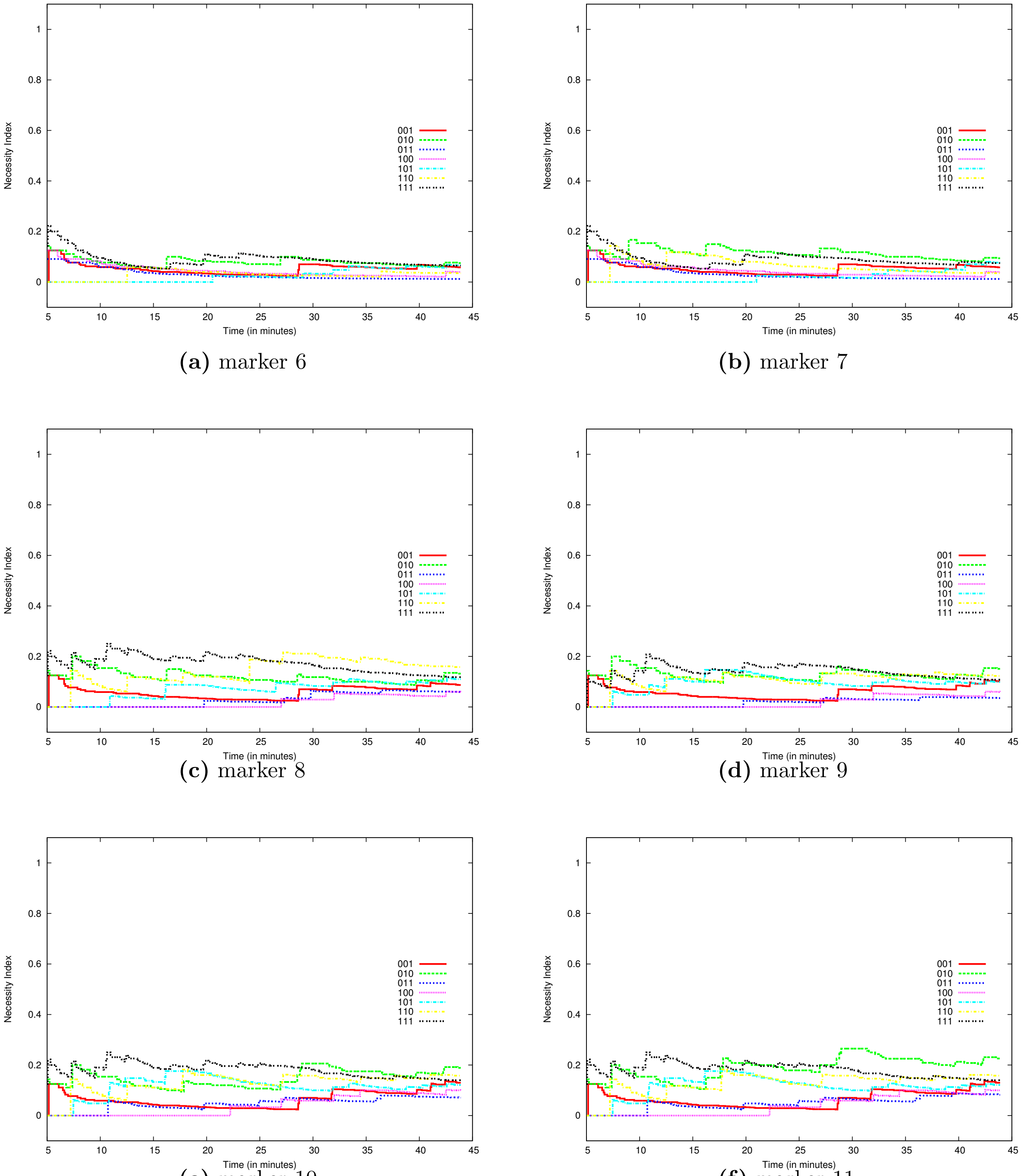
(c) marker 2

(d) marker 3

(e) marker 4

(f) marker 5

Figure 5.37: The necessity index, $N_i^m(t)$, for each of the six body markers of the first robot. Each figure shows 7 lines which correspond to one of the 7 possible types of motor commands: $001, \ldots, 111$. To be considered for classification as "self" each marker must have a necessity index $N_i^m(t) > 0.75$ for at least one motor command, m, at the end of the trial. This is true for all body markers shown in this figure. Thus, they are correctly classified as "self" in this case.



(a) marker 6 **(b)** marker 7

(c) marker 8 **(d)** marker 9

(e) marker 10 **(f)** marker 11

Figure 5.38: The necessity index, $N_i^m(t)$, for each of the six body markers of the second robot. Each figure shows 7 lines which correspond to one of the 7 possible types of motor commands: $001, \ldots, 111$. To be considered for classification as "self" each marker must have a necessity index $N_i^m(t) > 0.75$ for at least one motor command, m, at the end of the trial. This is not true for the body markers of the second robot shown in this figure. Thus, they are correctly classified as "other" in this case.

5.9.4 Experiments with Two Robots: Mimicking Movements

This test condition is the same as the one described in Section 5.8.4. The mean efferent-afferent delay for this experiment was also set to 1.035 seconds. Note that this value is different from the wrong value (1.36363 seconds) estimated for this degenerate case in Section 5.8.4.

Table 5.9 shows the values for the necessity and sufficiency indexes at the end of the 45 minute interval. As expected, the sufficiency indexes for all body markers of the first robot are close to 1 (see Figure 5.39). Similarly, the necessity indexes are close to 1 for at least one motor command (see Figure 5.41). For the body markers of the second robot the situation is just the opposite. Figure 5.40 shows their sufficiency indexes calculated over time. Figure 5.42 shows their sufficiency indexes calculated for each of the seven possible motor commands.

Somewhat surprisingly, the mimicking test condition turned out to be the easiest one to classify. Because the second robot always start to move a fixed interval of time after the first robot, almost no temporally contingent movements are detected for its body markers. Thus, both the necessity and sufficiency indexes for most markers of the second robot are equal to zero. Marker 8 is an exception because it is the counterpart to marker 2 which has the noisiest position detection (see Figure 5.9).

Table 5.9: Values of the necessity and sufficiency indexes at the end of the trial. All markers are classified correctly as "self" or "other" in this case.

Marker	$\max_m(N_i^m(t))$	$S_i(t)$	**Threshold** α	**Classification**	**Actual**
M0	0.941	0.905	0.75	"self"	"self"
M1	1.000	0.925	0.75	"self"	"self"
M2	1.000	0.918	0.75	"self"	"self"
M3	1.000	0.995	0.75	"self"	"self"
M4	1.000	0.988	0.75	"self"	"self"
M5	1.000	0.994	0.75	"self"	"self"
M6	0.000	0.000	0.75	"other"	"other"
M7	0.022	0.007	0.75	"other"	"other"
M8	0.059	0.011	0.75	"other"	"other"
M9	0.000	0.000	0.75	"other"	"other"
M10	0.000	0.000	0.75	"other"	"other"
M11	0.000	0.000	0.75	"other"	"other"

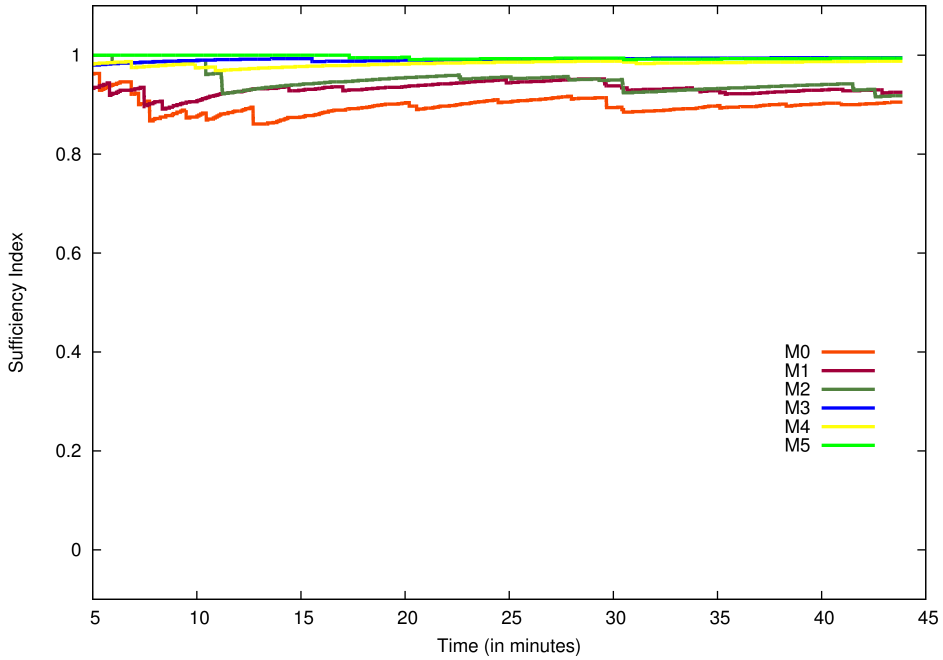


Figure 5.39: The figure shows the sufficiency indexes calculated over time for the six body markers of the first robot in the mimicking dataset. As expected these values are close to 1, and thus, above the threshold $\alpha = 0.75$. The same is true for the necessity indexes as shown in Figure 5.41. Thus, all markers of the first robot are classified as "self."

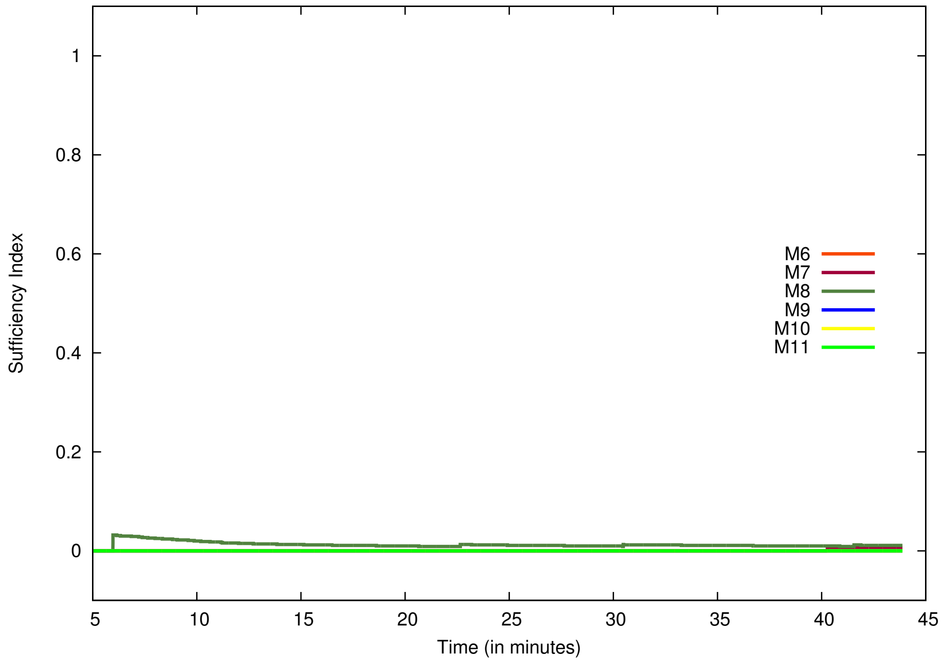
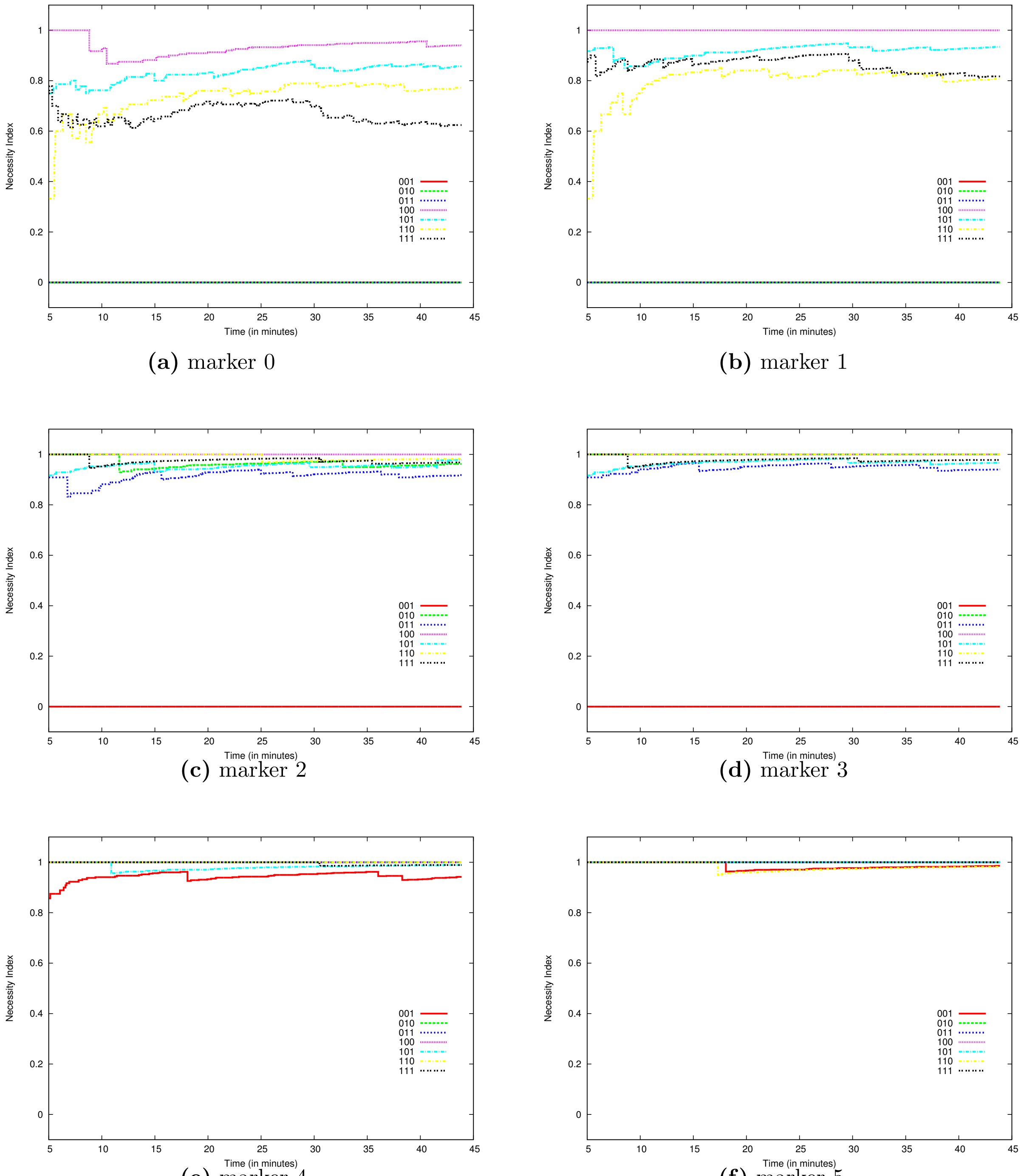


Figure 5.40: The figure shows the sufficiency indexes calculated over time for the six body markers of the second robot in the mimicking dataset. As expected these values are close to 0, and thus, below the threshold $\alpha = 0.75$. The same is true for the necessity indexes as shown in Figure 5.42. Thus, all markers of the first robot are classified as "other."



(a) marker 0

(b) marker 1

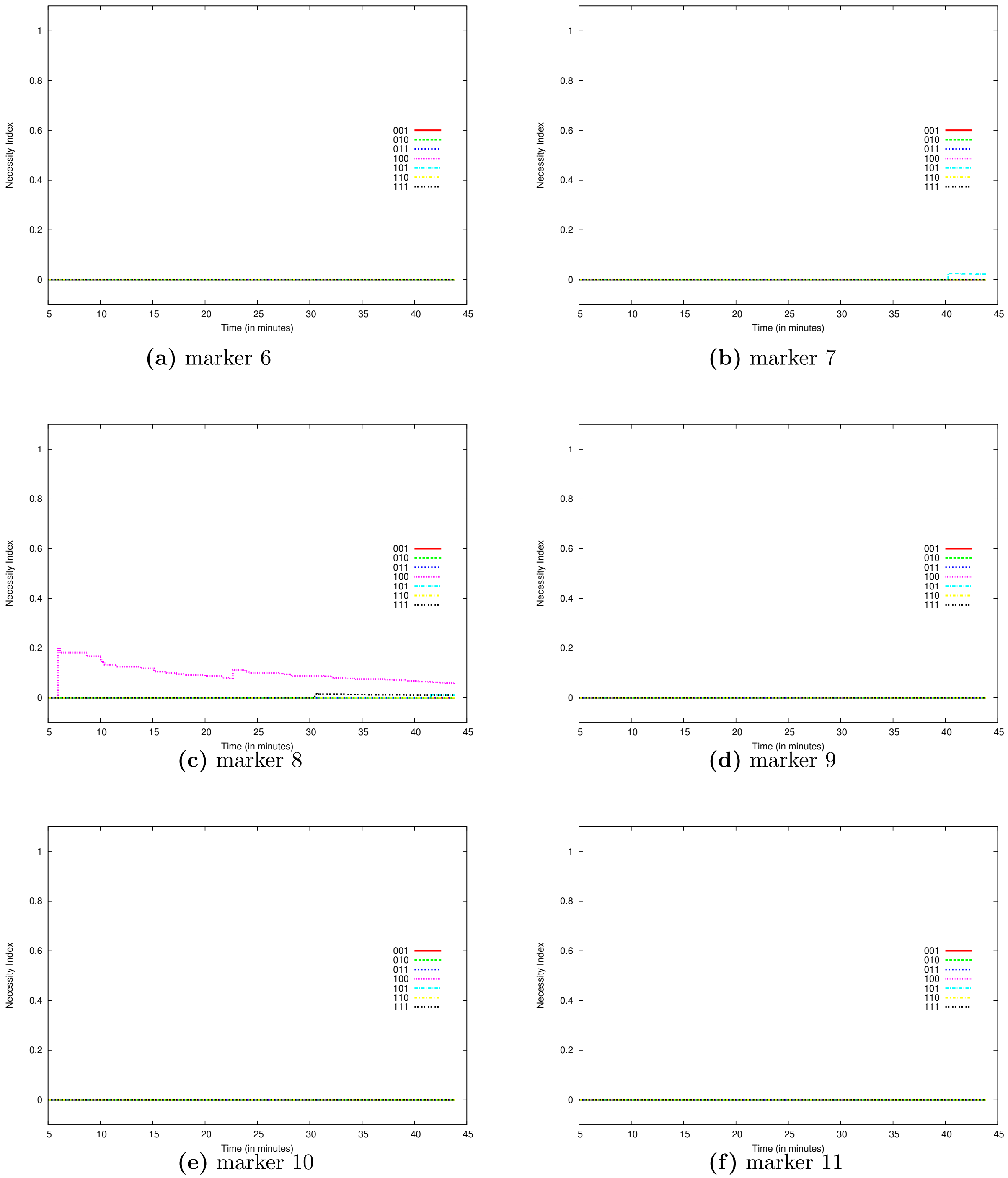
(c) marker 2

(d) marker 3

(e) marker 4

(f) marker 5

Figure 5.41: The necessity index, $N_i^m(t)$, for each of the six body markers. Each figure shows 7 lines which correspond to one of the 7 possible types of motor commands: $001, \ldots, 111$. To be considered for classification as "self" each marker must have a necessity index $N_i^m(t) > 0.75$ for at least one motor command, m, at the end of the trial. This is true for all body markers shown in this figure. Thus, they are correctly classified as "self."



(a) marker 6

(b) marker 7

(c) marker 8

(d) marker 9

(e) marker 10

(f) marker 11

Figure 5.42: The necessity index, $N_i^m(t)$, for each of the six body markers. Each figure shows 7 lines which correspond to one of the 7 possible types of motor commands: $001, \ldots, 111$. To be considered for classification as "self" each marker must have a necessity index $N_i^m(t) > 0.75$ for at least one motor command, m, at the end of the trial. This is not true for the body markers of the second robot shown in this figure. Thus, they are correctly classified as "other" in this case.

5.10 Self-Detection in a TV monitor

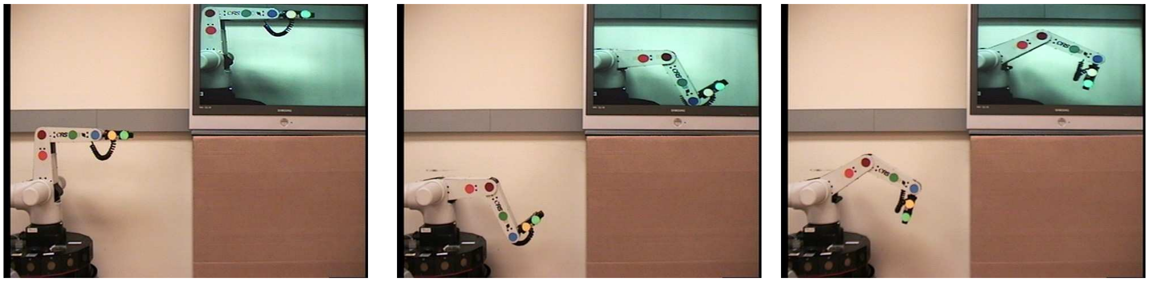


Figure 5.43: Frames from the TV sequence. The TV image shows in real time the movements of the robot captured from a camera which is different from the robot's camera.

This section describes and experiment which tests whether a robot can use its estimated efferent-afferent delay and the methodology described in Section 5.9 to detect that an image shown in a TV monitor is an image of its own body. This experiment was inspired by similar setups used by Watson (1994) in his self-detection experiments with infants.

The experiment described in this section adds a TV monitor to the existing setup as shown in Figure 5.43. The TV image displays the movements of the robot in real time as they are captured by a camera different from the robot's camera. A more detailed description of this experimental setup is given in Section 6.6 which uses the self-detection results described here to achieve video-guided robot behaviors.

A new data set of 500 movement commands was gathered for this experiment. Similarly to previous experiments, the robot was under the control of the motor babbling procedure. The dataset was analyzed in the same way as described in the previous sections. The only difference is that the position detection for the TV markers is slightly more noisy than in previous datasets. Therefore, the raw marker position data was averaged over three consecutive frames (the smallest number required for proper averaging). Also, marker movements shorter than six frames in duration were ignored.

The results for the sufficiency and necessity indexes for the robot's six body markers are similar to those described in the previous sections and thus will not be discussed any further. This section will only describe the results for the images of the six body markers in the TV monitor which will be refereed to as TV markers (or TV0, TV1, . . . ,TV5).

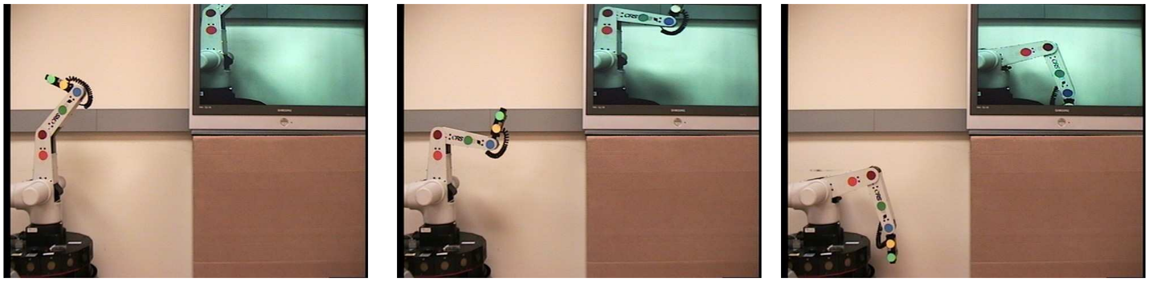


Figure 5.44: Frames from the TV sequence in which some body markers are not visible in the TV image due to the limited size of the TV screen.

Figure 5.45 shows the sufficiency indices calculated for the six TV markers. Somewhat surprisingly, the sufficiency indexes for half of the markers do not exceed the threshold value of 0.75 even though these markers belong to the robot's body and they are projected in real time on the TV monitor. The reason for this, however, is simple and it has to do with the size of the TV image. Unlike the real body markers which can be seen by the robot's camera for all body poses, the projections of the body markers in the TV image can only be seen when the robot is in specific body poses. For some body poses the robot's arm is either too high or too low and thus the markers cannot be observed in the TV monitor. Figure 5.44 shows several frames from the TV sequence to demonstrate this more clearly. The actual visibility values for the six TV markers are as follows: 99.9% for TV0, 99.9% for TV1, 86.6% for TV2, 72.1% for TV3, 68.5% for TV4, and 61.7% for TV5. In contrast, the robot's markers (M0-M5) are visible 99.9% of the time.

This result prompted a modification of the formulas for calculating the necessity and sufficiency indexes. In addition to taking into account the specific motor command, the self-detection algorithm must also take into account the visibility of the markers. In all previous test cases all body markers were visible for all body configurations (subject to the occasional transient sensory noise). Because of that, visibility was never considered even though it was implicitly included in the detection of marker movements. For more complicated robots (e.g., humanoids) the visibility of the markers should be taken into account as well. These robots have many body poses for which they may not be able to see some of their body parts (e.g., hand behind the back).

To address the visibility issue the following changes were made to the way the necessity and sufficiency indexes are calculated. For each marker a new variable $V_i(t)$ is introduced, which has a value of 1 if the i-th marker is visible at time t and 0 otherwise. The robot checks the visibility of each marker in the time interval immediately following a motor command. Let the k-th motor command be issued at time T_k and the (k+1)-st command be issued at time T_{k+1}. Let $\hat{T}_k \in [T_k, T_{k+1})$ be the time at which the k-th motor command is no longer considered contingent upon any visual movements. In other words, $\hat{T}_k = T_k + \mu + \beta\mu$, where μ is the average efferent-afferent delay and $\beta\mu$ is the estimate for the standard deviation calculated using Weber's law (see Section 5.8). If the i-th marker was visible for less than 80% of the time in the interval $\left[T_k, \hat{T}_k\right)$ (i.e., if $\sum_{t=T_k}^{\hat{T}_k} V_i(t)/(T_{k+1} - T_k) \leq 0.80$) then the movements of this marker (if any) are ignored for the time interval $[T_k, T_{k+1})$ between the two motor commands. In other words, none of the three counters ($T_i(t)$, $C_i(t)$, and $M_i(t)$) associated with this marker and used to calculate its necessity and sufficiency indexes are updated until the next motor command.

After the visibility correction, the sufficiency indexes for the images of the six TV markers are all above the 0.75 threshold as shown in Figure 5.46. The only exception is the yellow marker (TV4) which has a sufficiency index of 0.64 even after correcting for visibility. The reason for this is the marker's color, which appears very similar to the background wall in the TV image. As a result, its position tracking is noisier than before.

Figure 5.47 shows the values of the necessity indexes for the six TV markers before the visibility correction. Their values after the visibility correction are shown in Figure 5.48. As in the previous sections, the necessity values do not exceed the threshold of 0.75 because the robot must also correct for the type of motor command that is being issued. This is necessary because some markers (e.g., those on the shoulder) are not affected by all motor commands and remain static. They move only under specific motor commands (e.g., those that move the shoulder joint). Figure 5.49 shows the necessity indexes calculated for each type of motor command for each TV marker (after taking the visibility of the markers into account).

As can be seen from Figure 5.46 and Figure 5.49 five out of six TV markers are correctly classified as self because at the end of the trial they all have a sufficiency index greater than

0.75 and a necessity index greater than 0.75 for at least one motor command. The only marker that was not classified as self was the yellow marker for reasons explained above.

The results of this section offer at least one insight that can be used in future studies of self-detection with both animals and humans: visibility of features must be taken into account when calculating the probabilities of necessity and sufficiency. Previous studies with infants (e.g., Watson (1994); Bahrick and Watson (1985)) have tacitly assumed that the required features are visible at all times.

The results of this section demonstrate, to the best of my knowledge, the first-ever experiment of self-detection by a robot in a TV monitor. Section 6.6 builds upon the results from this section and shows how they can be used to achieve video-guided robot behaviors (another first).

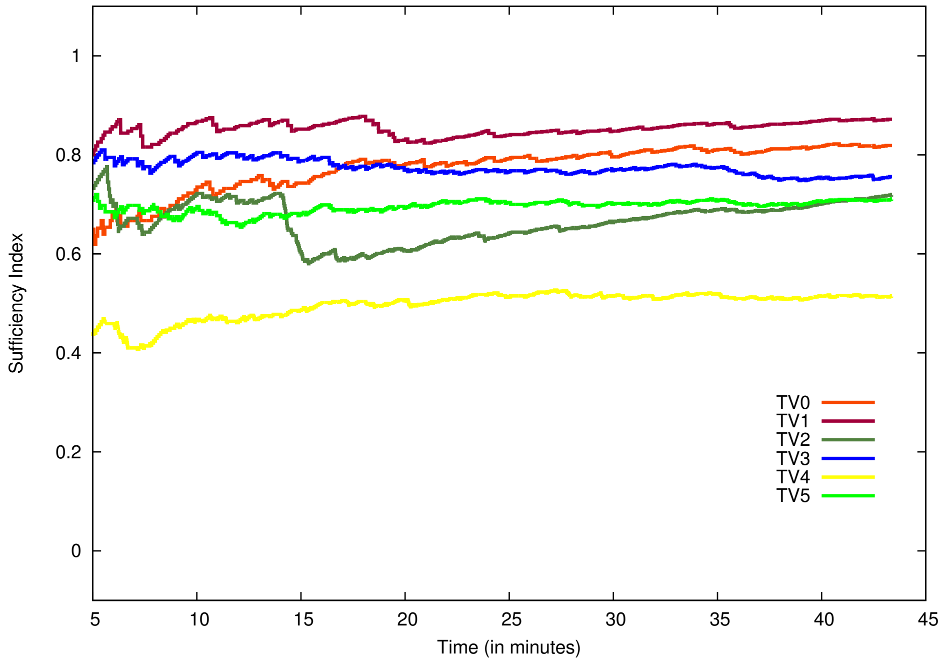


Figure 5.45: The sufficiency indexes calculated over time for the six TV markers. These results are calculated *before* taking the visibility of the markers into account.

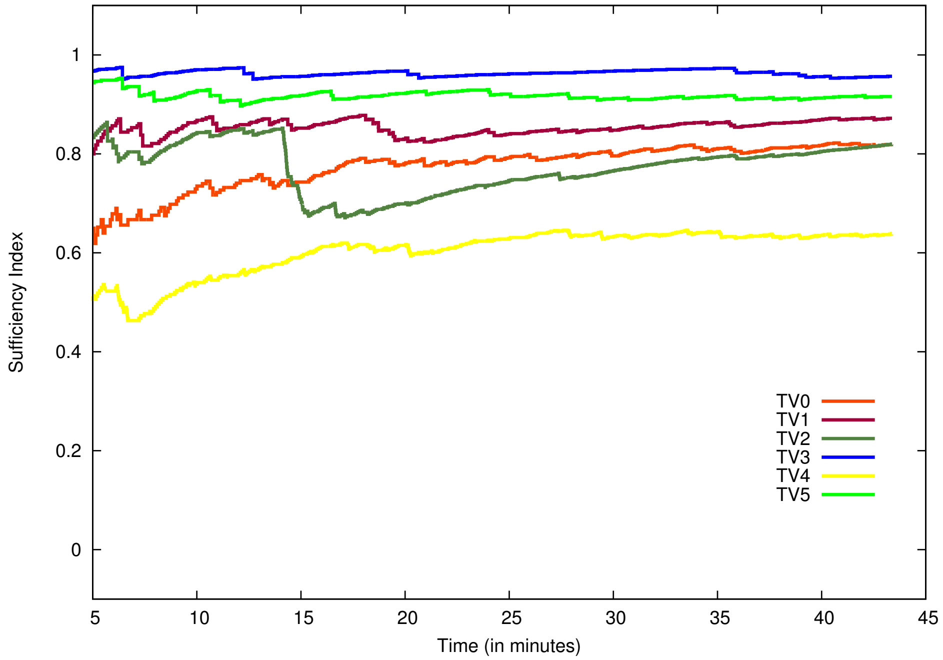


Figure 5.46: The sufficiency indexes calculated over time for the six TV markers. These results are calculated *after* taking the visibility of the markers into account.

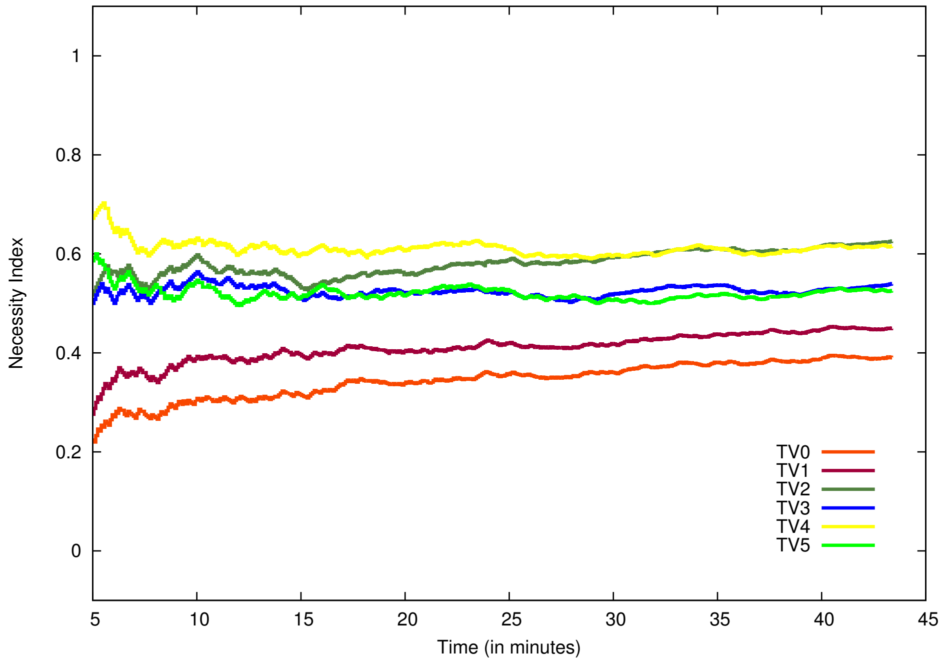


Figure 5.47: The necessity indexes calculated over time for the six TV markers. These results are calculated *before* taking the visibility of the markers into account.

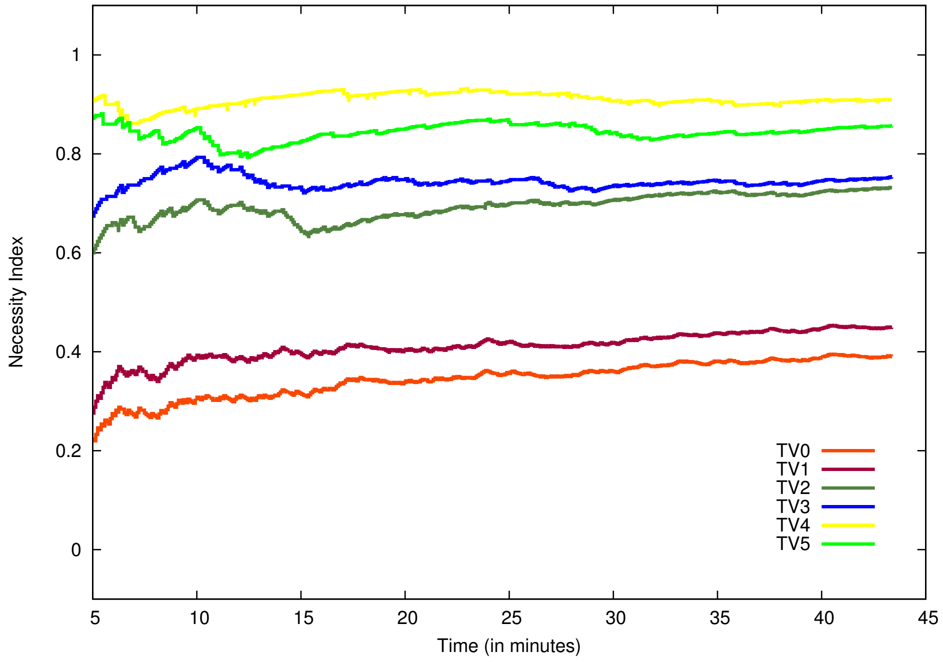
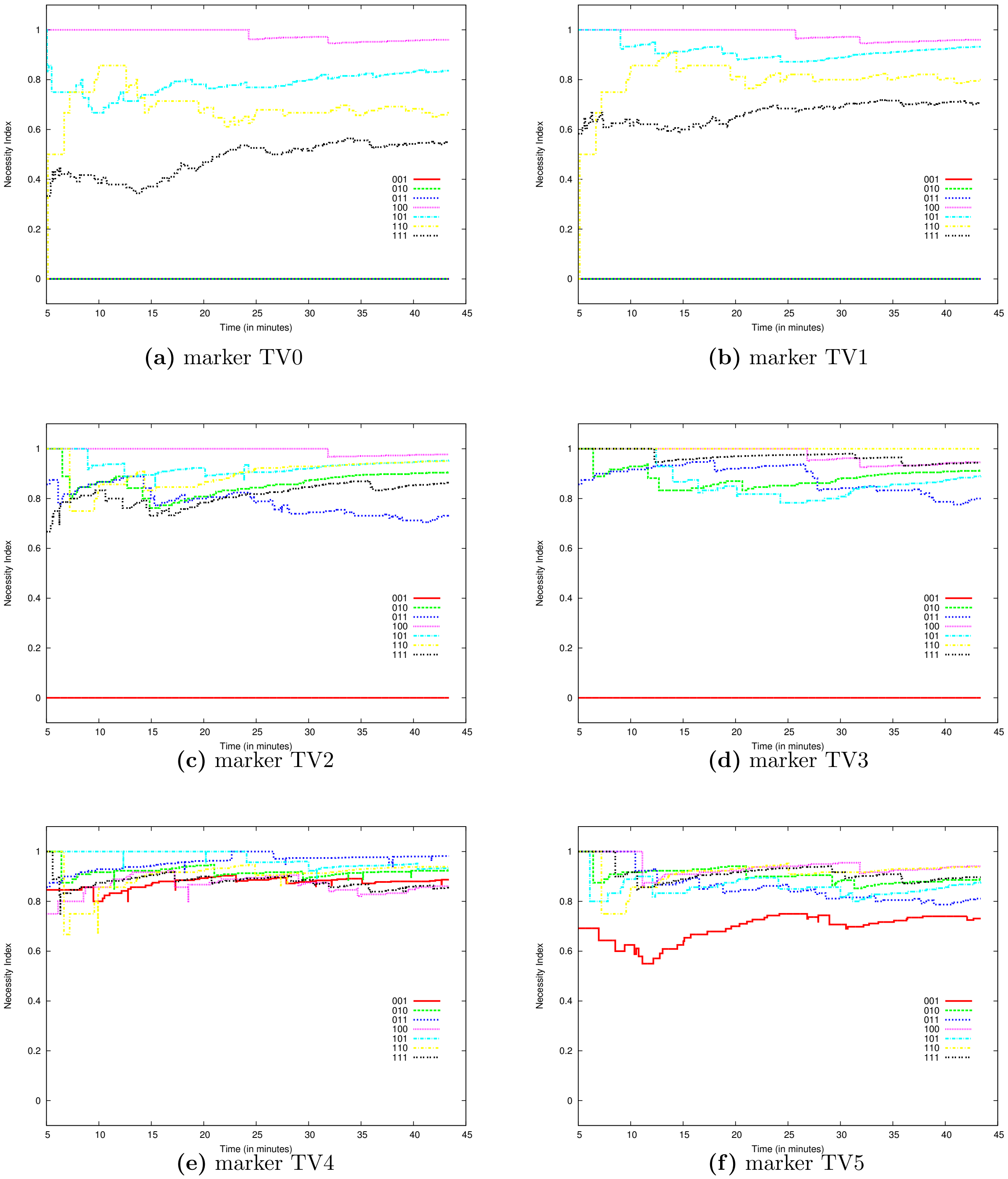


Figure 5.48: The necessity indexes calculated over time for the six TV markers. These results are calculated *after* taking the visibility of the markers into account.



(a) marker TV0

(b) marker TV1

(c) marker TV2

(d) marker TV3

(e) marker TV4

(f) marker TV5

Figure 5.49: Values of the necessity index, $N_i^m(t)$, for each of the six TV markers. Each figure shows 7 lines which correspond to one of the 7 possible types of motor commands: $001, \ldots, 111$. To be considered for classification as "self" each marker must have at the end of the trial a necessity index, $N_i^m(t) > 0.75$ for at least one motor command, m. These graphs are calculated *after* taking the visibility of the TV markers into account.

5.11 Chapter Summary

This chapter described a methodology for autonomous self-detection by a robot. The methodology is based on the detection of the temporal contingency between motor commands (efferent signals) and visual movements (afferent signals) to estimate the efferent-afferent delay of the robot. It was shown how the robot can estimate its own efferent-afferent delay from self-observation data that can be gathered while the robot performs motor babbling, i.e., random joint movements similar to the primary circular reactions described by Piaget. It was shown that the self-detection algorithm performs well for the experimental conditions described in this chapter.

This chapter also introduced a method for feature-level self-detection based on the ideas described by Watson (1994). The method maintains a probabilistic estimate across all features as to whether or not they belong to the robot's body. The probabilities are estimated based on probabilistic estimates of necessity and sufficiency. The method was successfully used by the robot to detect its self-image in a TV monitor.

The results of this section show that Watson's ideas are suitable for application on robots. However, there are some implementation details which Watson did not foresee (or maybe they were not applicable to his experimental setups with infants). For example, the size of the TV image imposes a restriction on which body markers can be seen and for which body poses. Without correcting for visibility the values of the necessity and sufficiency indexes can exhibit at most medium levels of contingency. Another factor that is not mentioned by Watson is that the self-detection algorithm must take into account the types of motor commands that are issued as not all body markers are moved by a given motor command. Without this correction the necessity indexes cannot reach the near perfect values (greater than 0.75 in the robot experiments) required for successful self-detection. Both of these modifications were implemented and tested successfully on the robot.

The experimental results described in this chapter show that a robot can successfully distinguish between its own body and the external environment. These results directly support the first research question stated in Section 1.2. The robot was able to correctly classify different visual stimuli as either “self” or “other.”

The next chapter builds upon the self-detection work presented here and uses the self-detection results to construct a sensorimotor model of the robot’s body. This model includes only the sensations (color markers) that were classified as “self.”

CHAPTER VI

EXTENDABLE ROBOT BODY SCHEMA

6.1 Introduction

The sense of body is probably one of the most important senses and yet it is one of the least well studied. It is a complex sense, which combines information coming from proprioceptory, somatosensory, and visual sensors to build a model of the body called the *body schema.* It has been shown that the brain keeps and constantly updates such a model in order to register the location of sensations on the body and to control body movements (Head and Holmes, 1911; Berlucchi and Aglioti, 1997; Berthoz, 2000; Graziano et al., 2002).

Recent studies in neuroscience have shown that this model of the body is not static but can be extended by noncorporeal objects attached to the body such as clothes, ornaments, and tools (Aglioti et al., 1997; Iriki et al., 1996, 2001). Thus, it may be the case that as far as the brain is concerned the boundary of the body does not have to coincide with anatomical boundaries (Iriki et al., 1996, 2001; Tiemersma, 1989; Ramachandran and Blakeslee, 1998).

This chapter describes a computational model for a robot body schema which has properties similar to its biological analog. The robot learns its body schema representation by combining visual and proprioceptive information. The resulting representation can be scaled, rotated, and translated. The morphing properties of the body schema be used to accommodate attached tools. They can also be used to achieve video-guided behaviors as described later in this chapter.

6.2 Related Work

6.2.1 Related Work in Neuroscience

The notion of *body schema* was first suggested by Head and Holmes (1911) who studied the perceptual mechanisms that humans use to perceive their own bodies. They define the *body schema*[1] as a postural model of the body and a model of the surface of the body. It is a perceptual model of the body formed by combining information from proprioceptory, somatosensory, and visual sensors. They suggested that the brain uses such a model in order to register the location of sensations on the body and to control body movements.

Indirect evidence supporting the existence of a body schema comes from numerous clinical patients who experience disorders in perceiving parts of their bodies often lacking sensations or feeling sensations in the wrong place (Frederiks, 1969; Head and Holmes, 1911). One such phenomenon called *phantom limb* is often reported by amputees who feel sensations and even pain as if it was coming from their amputated limb (Melzack, 1992; Ramachandran and Rogers-Ramachandran, 1996).

Direct evidence for the existence of a *body schema* is provided by recent studies which have used brain imaging techniques to identify the specialized regions of the primate (and human) brain responsible for encoding it (Berlucchi and Aglioti, 1997; Iriki et al., 1996, 2001; Graziano et al., 2000). Other studies have shown that body movements are encoded in terms of the body schema (Berthoz, 2000; Graziano et al., 2002). This seems to be the case even for reflex behaviors (Berthoz, 2000).

Perhaps the most interesting property of the body schema is that it is not static but can be modified and extended dynamically in short periods of time. Such extensions can be triggered by the use of non-corporeal objects such as clothes, ornaments, and tools. For example, Head and Holmes (1911, p. 188) suggest that the feather on a woman's hat affects her ability to move and localize in the environment. Thus, the body schema is not tied to anatomical boundaries. Instead the actual boundaries depend on the intended use of the body parts and the external objects attached to the body (Tiemersma, 1989).

The inclusion of inanimate objects into the body schema is a temporary phenomenon which is contingent upon the actual use of the objects. For example, when people drive a car they get the feeling that the boundary of the car is part of their own body (Graziano et al., 2000; Schultz, 2001). However, when they get out of the car their body schema goes

[1] The exact term that they use is "postural scheme". The term *body schema* was first used by Pick (1922) and was later made popular by Schilder (1923) who published a monograph in German entitled "Das Körperschema".

back to normal. In some cases, however, it is possible that a more permanent modification of the body schema can be established by objects such as wedding rings that are worn for extended periods of time (Aglioti et al., 1997).

It has been suggested that the pliability of the body schema plays a role in the acquisition of tool behaviors (Head and Holmes, 1911; Paillard, 1993). Recent studies conducted with primates seem to support this hypothesis (Iriki et al., 1996; Berlucchi and Aglioti, 1997; Berti and Frassinetti, 2000). Iriki et al. (1996) trained a macaque monkey to retrieve distant objects using a rake and recorded the brain activity of the monkey before, during, and after tool use. They discovered a large number of bimodal neurons (sensitive to visual and somatosensory stimuli) that appear to code the schema of the hand (Iriki et al., 1996). Before tool use the receptive fields (RF) of these neurons were centered around the hand. During tool use, however, the somatosensory RF stayed the same but the visual RF was altered to include the entire length of the rake or to cover the expanded accessible space (Iriki et al., 1996).

This modification of the visual receptive field is limited to the time of tool usage and is conditional upon the intention to use the tool. When the monkey stopped using the tool, or even continued to hold the tool without using it, the visual RF contracted back to normal (Iriki et al., 1996). In a follow-up study the monkey was prevented from directly observing its actions and instead was given feedback only through a camera image projected on a video monitor. In this case the visual RF of the bimodal neurons was projected onto the video screen (Iriki et al., 2001).

These studies suggest that the encoding of the body schema in the brain is extremely pliable and tools can easily be incorporated into it. Studies conducted with humans have reached similar conclusions (Berti and Frassinetti, 2000; Farné and Ládavas, 2000). In addition to tools, the body schema can also be modified by prosthetic limbs in amputee patients. These patients can incorporate the prosthetic limb into their body schema in such a way that they can perform the same or similar tasks as if with their real limb (Tsukamoto, 2000).

Unfortunately, little is known about the neural organization of the body schema in primate brains. One recent study, however, has made some quite striking and unexpected discoveries which seem to reject previous theories of the organization of the map of the body and how it is used to control body movements. Graziano et al. (2002) microstimulated the primary motor and premotor cortex of monkeys at behaviorally relevant time scales (500 ms). Previous studies have already established that stimulation in these areas produces muscle twitches but the stimulation times have always been quite short. The longer stimulation times allowed the true purpose of these areas to be revealed:

> *"Stimulation on a behaviorally relevant time scale evoked coordinated, complex postures that involved many joints. For example, stimulation of one site caused the mouth to open and also caused the hand to shape into a grip posture and move to the mouth. Stimulation of this site always drove the joints toward this final posture, regardless of the direction of movement required to reach the posture. Stimulation of other cortical sites evoked different postures."* (Graziano et al., 2002)

Similar results were obtained for facial expressions as well: stimulation of a particular site always produced the same final facial expression regardless of the initial facial expression. Similar results were also obtained even when one of the monkeys was anesthetized but the accuracy of the final postures in this case was less precise. Another interesting observation was that obstacles were completely ignored during arm movements induced by microstimulation. If there was an obstacle along the trajectory from the starting posture to the final posture the hand will try to go straight through as if the obstacle did not exist.

These results have led Graziano et al. (2002) to conclude that these sites in the primate brain form a coherent map of the workspace around the body. What is even more important, however, is the realization that this map plays an active part in the formation and execution of all kinds of movements. Achieving complex postures involving multiple joints may not be a highly complex task but a consequence of the encoding of this map. Interpolation between different building blocks of this map (i.e., final postures) can be used to produce highly coordinated movements across multiple joints.

Section 6.3 describes a computational model of a Self-Organizing Body-Schema (SO-BoS) developed by Morasso and Sanguineti (1995) which is adopted and extended in this chapter. This model displays similar properties to the observations made by Graziano et al.

(2002). The equivalent of final postures in Graziano et al.'s paper are called *body icons* in the model of Morasso and Sanguineti (1995). When a body icon is stimulated it acts as an attractor causing the joint configuration of the robot to converge on the one described by the body icon. Extension of SO-BoS model, motivated by the findings of (Iriki et al., 1996), will be introduced in Section 6.5 to model the extensibility of the Robot Body Schema.

6.2.2 Related Work in Robotics

The robotics work on body schemas is still in its infancy as only a few papers have attempted to tackle this subject. Yoshikawa et al. (2002) formulated a fully connected neural network model that identified the common firing patterns between tactile, visual, and proprioceptive sensors. Their model was capable of making the right associations between sensory modalities but lacked extensibility properties.

Nabeshima, Lungarella, and Kuniyoshi (2005) describe a method for changing the properties of the robot's controller (which is based on inverse kinematics) to accommodate attached tools. The extension is triggered by the coincidence in the firing of tactile sensors (at the hand which is grasping the tool) and the diminishing visual distance between the free end of the tool and some visual landmark. Their extension method requires direct physical contact with the object.

This chapter builds upon our previous work (Stoytchev, 2003) which introduced a computational model for an extendable robot body schema (RBS). The model uses visual and proprioceptive information to build a representation of the robot's body. The visual components of this representation are allowed to extend beyond the boundaries of the robot's body. The proprioceptive representation, however, remains fixed at all times and thus the robot can perform visually-guided tool movements using its extended body.

In our previous study the extension of the RBS was triggered by tactile sensations generated by objects that are attached to the robot's body, e.g., tools. The novel extension mechanism described here is triggered by the temporal contingency between the actions of the robot and the observed movements of an attached object or the self-movements in a video image (see Section 6.6), i.e., direct physical contact is no longer required.

6.3 The Self-Organizing Body Schema (SO-BoS) Model

The computational model chosen for the implementation of the robot body schema is based on the *Self-Organizing Body-Schema* (SO-BoS) model introduced in (Morasso and Sanguineti, 1995). This section provides a brief summary of the SO-BoS model and interprets it in terms of existing robotics research. The sections that follow modify and extend the SO-BoS model such that it can have extensibility properties as well.

The SO-BoS model is introduced with the help of the following notation. Let the robot have m joints which are controlled with a kinematic configuration vector $\theta = \{q_1, q_2, \ldots, q_m\}$, where each q_i represents a target joint angle. Furthermore, let there be a set of n distinct visual features, F =$\{f_1$, f_2, ..., $f_n\}$, on the surface of the robot's body that can be uniquely identified by the robot's vision system. Let the position of feature f_i in camera-centric coordinates be denoted with v_i and let the set of all such vectors for all body markers be denoted with $V = \{v_1, v_2, \ldots, v_n\}$.

The configuration space, $\mathcal{C}$, of the robot is a subset of the Cartesian product of the joint angles:

$$\mathcal{C} \subseteq q_1 \times q_2 \times \ldots \times q_m \subseteq \Re^m$$

The sensor space $\mathcal{S}$ of all sensory stimuli coming from the robot's body is a subset of the Cartesian product of all perceptual vectors:

$$\mathcal{S} \subseteq v_1 \times v_2 \times \ldots \times v_n \subseteq \Re^n$$

The main idea behind the RBS representation is to link the configuration space and the sensor space of the robot into one $\mathcal{CS}$-space ($\mathcal{CS}$-space = $\mathcal{C} \times \mathcal{S}$). This space can be used to identify the current robot configuration as well as to plan robot movements. Previous approaches described in the robotics literature have noted the usefulness of this space for planning and specifying robot movements (Hervé et al., 1991; Sharma et al., 1992; Sharma and Sutanto, 1996). However, they have used algebraic techniques to express this space as an (m+n)-dimensional manifold which is hard to use even for simple robots (Sharma and Sutanto, 1996). The current approach uses non-parametric statistical techniques to approximate the $\mathcal{CS}$-space as described below.

The robot body schema model is built around the concept of a *body icon*. A body icon is a tuple $(\tilde{\theta}_i, \tilde{V}_i)$ representing the kinematic and sensory components of a specific joint configuration (or pose) of the robot (variables with ˜ represent fixed estimates). A large number of empirically learned body icons, $\{(\tilde{\theta}_i, \tilde{V}_i), i = 1, \ldots, \mathcal{I}\}$, is used to represent the robot's body schema (Table 6.1). It is believed that the brain uses a similar representation encoded as a cortical map (Morasso and Sanguineti, 1995; Graziano et al., 2002).

Table 6.1: Body icons table. Each row of the table represents one body icon which consists of the fixed estimates for the kinematic and sensory vectors which are associated with a specific body pose.

#	**Kinematic components**	**Sensory components**
1	$\tilde{\theta}_1 = \{\tilde{q}_1^1, \tilde{q}_2^1, \ldots, \tilde{q}_m^1\}$	$\tilde{V}_1 = \{\tilde{v}_1^1, \tilde{v}_2^1, \ldots, \tilde{v}_n^1\}$
2	$\tilde{\theta}_2 = \{\tilde{q}_1^2, \tilde{q}_2^2, \ldots, \tilde{q}_m^2\}$	$\tilde{V}_2 = \{\tilde{v}_1^2, \tilde{v}_2^2, \ldots, \tilde{v}_n^2\}$
...	...	...
$\mathcal{I}$	$\tilde{\theta}_{\mathcal{I}} = \{\tilde{q}_1^{\mathcal{I}}, \tilde{q}_2^{\mathcal{I}}, \ldots, \tilde{q}_m^{\mathcal{I}}\}$	$\tilde{V}_{\mathcal{I}} = \{\tilde{v}_1^{\mathcal{I}}, \tilde{v}_2^{\mathcal{I}}, \ldots, \tilde{v}_n^{\mathcal{I}}\}$

As an example, consider the planar robot with two rigid limbs and two rotational joints shown in Figure 6.1.a. The limbs have lengths $l_1 = 0.5$ and $l_2 = 0.35$. Both joints can rotate only 180 degrees, i.e., $0 \leq q_1, q_2 \leq \pi$. The robot has two body markers: M_1, M_2. Marker M_1 (red marker) is placed on the elbow of the robot. Marker M_2 (green marker) is placed at the free end of the second limb. The positions of the two body markers are given by the vectors v_1, v_2. Both v_1 and v_2 lie in a two-dimensional space defined by the camera image. The sensory vector is given by $V = \{v_1, v_2\}$ (see Figure 6.1.b). The kinematic or joint angle vector is given by $\theta = \{q_1, q_2\}$ (see Figure 6.1.c).

Table 6.2 shows the sensory and kinematic components of three body icons which are associated with three sample body poses of the two-joint robot.

As the robot performs different movements, the positions of these markers in camera coordinates keep changing. Over time, the robot can learn which marker positions correspond to which joint angles, i.e., it can learn new body icons. As the number of body icons is increased, the body icons table begins to approximate the working envelope of the robot. Figure 6.2.a shows the $\tilde{v}_1^i$ elements of the observed sensory vectors $\tilde{V}_i$ for 400 body icons

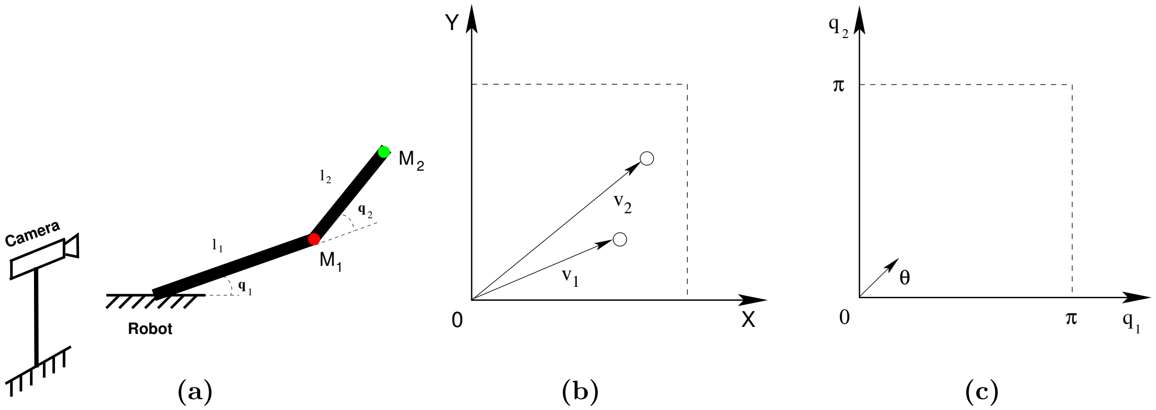
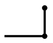
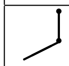


Figure 6.1: (a) The two-joint robot used in the example. The robot has two body markers M_1 and M_2 and two joint angles q_1 and q_2. (b) The coordinates of the two body body markers in visual space are given by the two vectors v_1 and v_2. (c) The motor vector, θ, for the robot configuration shown in a).

Table 6.2: Sample body icons table for the robot shown in Figure 6.1.a. Each row of the table represents the kinematic and visual vectors of a specific robot pose. The visual vectors are expressed in a coordinate system centered at the first rotational joint of the robot.

Body Pose	**Kinematic components** $\tilde{\theta}_i$		**Sensory components** $\tilde{V}_i$	
	$\tilde{q}_1^i$	$\tilde{q}_2^i$	$\tilde{v}_1^i(x,y)$	$\tilde{v}_2^i(x,y)$
	0°	90°	0.50, 0.00	0.50, 0.35
	45°	135°	0.35, 0.35	0.00, 0.35
	30°	60°	0.43, 0.25	0.43, 0.60

(i.e., $i = 1 \ldots \mathcal{I}$). These points represent a subset of all possible positions of the red (elbow) marker. Similarly, Figure 6.2.b shows the positions of the $\tilde{v}_2^i$ elements of 400 visual vectors $\tilde{V}_i$. These points represent a subset of all possible positions of the green (wrist) marker. The joint angle vectors are points lying inside a 2D square given by $0 \leq q_1, q_2 \leq \pi$ (Figure 6.3).

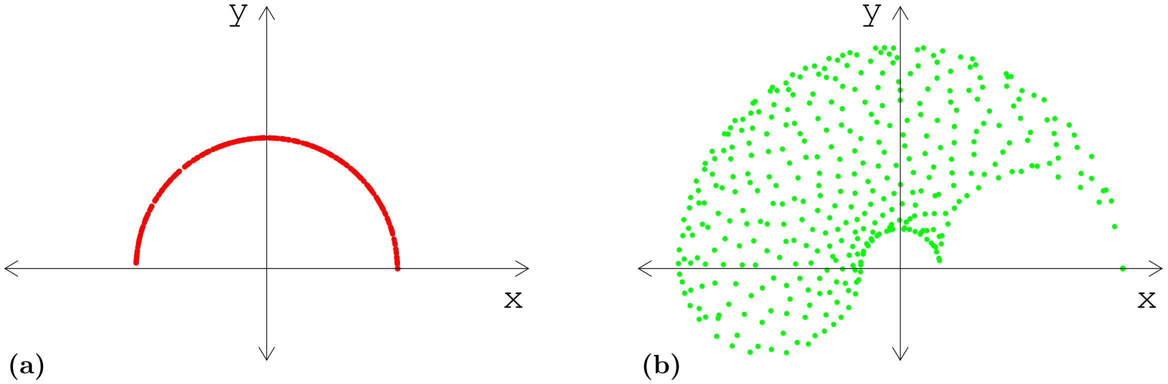


Figure 6.2: The sensory vectors $\tilde{v}_1^i$ and $\tilde{v}_2^i$ for 400 body poses of the robot shown in Figure 6.1. **(a)** $\tilde{v}_1^i$ - all observed positions of the red body marker, M_1; **(b)** $\tilde{v}_2^i$ - all observed positions of the green body marker, M_2.

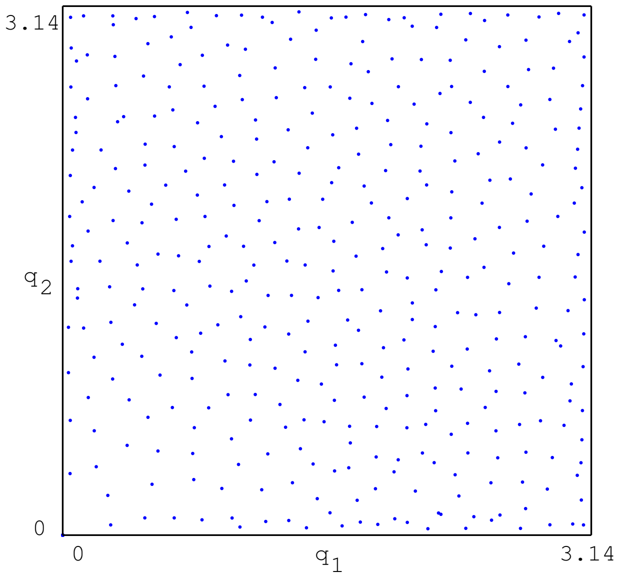


Figure 6.3: The figure shows the 400 joint vectors that correspond to the sensory vectors shown in Figure 6.2. Each point represents one of the $\tilde{\theta}_i = \{\tilde{q}_1^i, \tilde{q}_2^i\}$ joint vectors.

The second example used to demonstrate the body schema representation uses the CRS+ robot manipulator described in Section 4.2. To learn its body representation the robot performs the motor babbling procedure described in Section 5.6. Figure 6.4 shows some of the body poses picked by the motor babbling procedure. As in the previous experiments, the robot has six color markers placed on its body. The positions of these markers are tracked with a computer vision code based on color histogram matching (see Section 5.5).

Figure 6.5 shows the observed positions of all body markers in 500 body poses. As can be seen from the six plots, there is quite a bit of variation in the positions of the markers. For example, markers 0 and 1 are attached to the shoulder of the robot and thus their observed positions form an arc in camera coordinates. On the other hand, the light green marker (M5) which is located on the wrist can be observed in a large area of the camera image (see Figure 6.5.f).

As the number of body icons increases, the density with which their sensory components cover the working envelope of the robot also increases. Because the body representation is based on approximation between different body poses, however, the density does not have to be too high. Empirically it was established that 500 body icons are sufficient for the CRS robot. Nevertheless, the number of body icons required for smooth robot movements grows exponentially with the number of degrees of freedom. To overcome this limitation Section 6.4.4 considers the problem of learning nested representations.

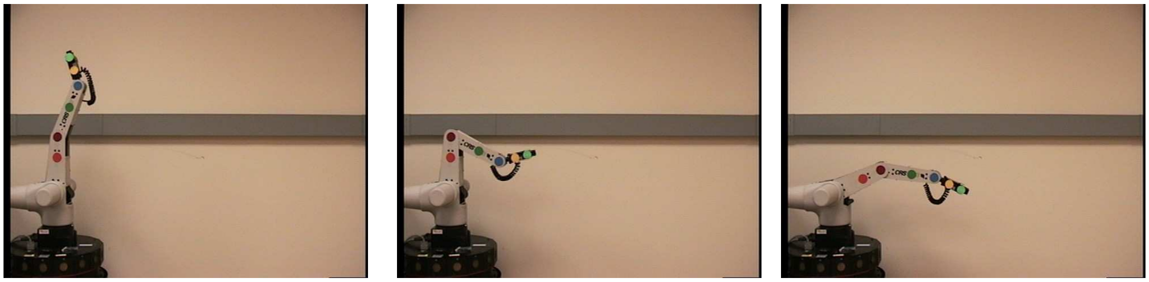
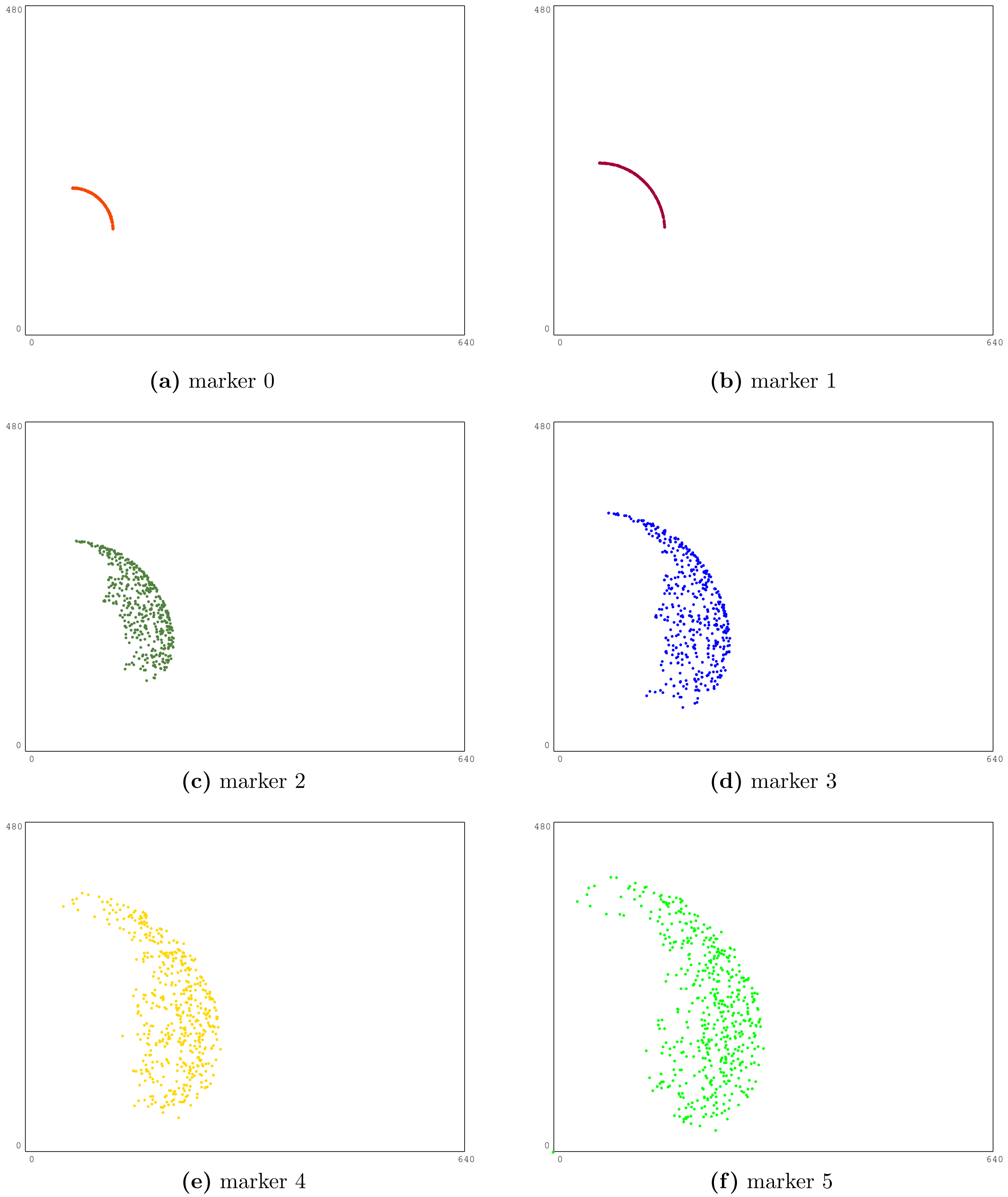


Figure 6.4: Several of the robot poses selected by the motor babbling procedure.



(a) marker 0

(b) marker 1

(c) marker 2

(d) marker 3

(e) marker 4

(f) marker 5

Figure 6.5: The six plots show the sensory components of 500 body icons learned during a single run of the motor babbling procedure. Each plot shows all 500 observed positions for a single body marker. The x and y coordinates of each point in the plots represent the observed centroid of the largest blob with a given color. The size of the camera image is 640x480.

6.3.1 Properties of the Representation

The representation of the robot body schema in terms of body icons has several properties which are described below.

The main building blocks of the SO-BoS model are processing elements (PEs) which have an activation function U_i and a preferred body icon $(\tilde{\theta}_i, \tilde{V}_i)$. The activation function of each PE is determined by the normalized gaussian or softmax function described by Morasso and Sanguineti (1995),

$$U_i(\theta) = \frac{G(\|\theta - \tilde{\theta}_i\|)}{\sum_j G(\|\theta - \tilde{\theta}_j\|)} \tag{6.1}$$

where G is a Gaussian with variance σ^2 and zero mean, θ is the current joint vector, and $\tilde{\theta}_i$ is the stored joint vector of the i^{-th} body icon. If the variance of G is small then a body icon will be activated only if its joint vector $\tilde{\theta}_i$ is close to the current joint vector θ. If σ^2 is large then more body icons will be activated for any specific query vector.

The learning algorithm described in (Morasso and Sanguineti, 1995) guarantees that each processing element has a neighborhood of other processing elements whose body icons are similar to its own. This similarity is both in terms of their θ and V vectors as well as their activation levels for a fixed joint vector (Morasso and Sanguineti, 1995). This locality property can be exploited to implement a gradient ascent strategy for moving the robot from one configuration to another as described in the next sub-section.

The mapping from joint vectors to sensory vectors (or forward kinematics) is explicit for joint vectors θ which are the same as one of the $\tilde{\theta}_i$ prototypes of the learned body icons, i.e., V is equal to $\tilde{V}_i$ (assuming zero sensory noise). For an arbitrary joint vector θ ($\theta \neq \tilde{\theta}_i, i = 1, \ldots, \mathcal{I}$), however, V is unknown. Nevertheless, it is possible to approximate the sensory vector with the following formula (Morasso and Sanguineti, 1995),

$$V^{approx}(\theta) \approx \sum_i \tilde{V}_i U_i(\theta) \tag{6.2}$$

where U_i is the activation value of the i^{-th} body icon due to proprioceptive information. The activation value is determined by the normalized Gaussian function.

Formula 6.2 can be interpreted as a two step approximation algorithm using *look up* and *interpolation*, similar to the memory-based learning approach (Atkeson and Schaal, 1995; Atkeson et al., 1997). The first step looks up the body icons that have joint vectors $\tilde{\theta}_i$ similar to the query joint vector θ. The second step sums the sensory vectors $\tilde{V}_i$ of these body icons (scaled by their activation value U_i) to approximate $V(\theta)$.

For individual components v_k of the sensory vector V, Formula 6.2 can be rewritten as:

$$v_k^{approx}(\theta) \approx \sum_i \tilde{v}_k^i U_i(\theta) \tag{6.3}$$

A flow chart diagram of Formula 6.3 is shown in Figure 6.6.

The approximation errors of Formula 6.3 can be estimated by comparing the approximated sensory vectors with their real values. Figure 6.7 shows the magnitude and direction of these errors as arrows. The base of each arrow represents the true value of the sensory vector calculated using forward kinematics. The tip of the arrow represents the approximated value using Formula 6.3. The gray points represent the $\tilde{v}_2$ sensory components of the body icons (same as in Figure 6.2.b). As Figure 6.7 shows, the approximation errors are very small except at the end of the reachability space. This is due to the fact that the number of body icons that cover this area is twice smaller than that for other locations (i.e., there are no body icons that cover the unreachable space).

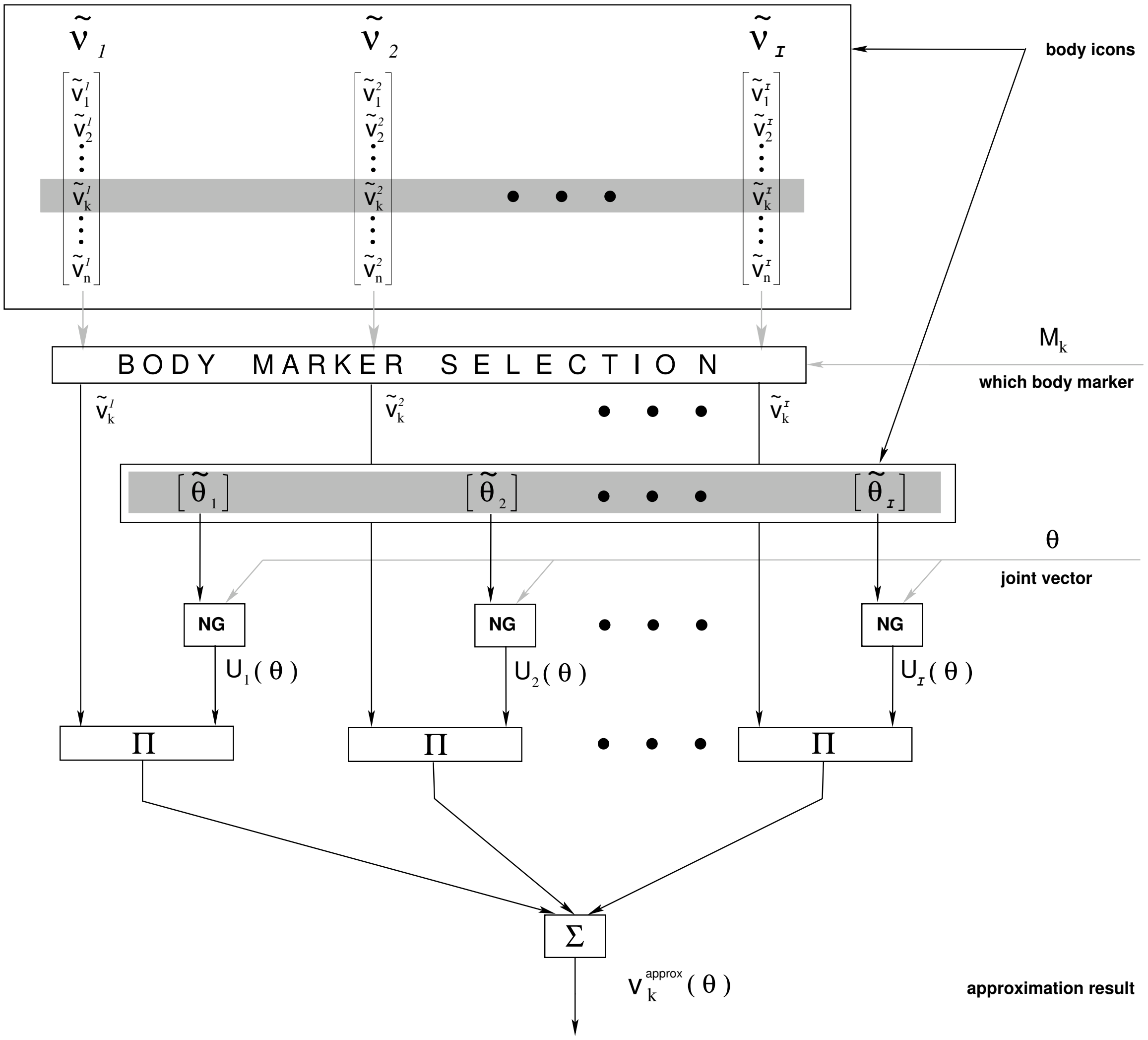


Figure 6.6: Flow chart diagram for approximating the sensory vector v_k given a joint vector θ using Formula 6.3. Notation: all $\tilde{\theta}$ and $\tilde{v}$ variables represent the stored kinematic and sensory components of the body icons; NG is the normalized Gaussian function (given by Formula 6.1) which is used to compute the activation value $U_i(\theta)$ of the i^{-th} body icon; $\sum$ and $\prod$ stand for summation and multiplication respectively.

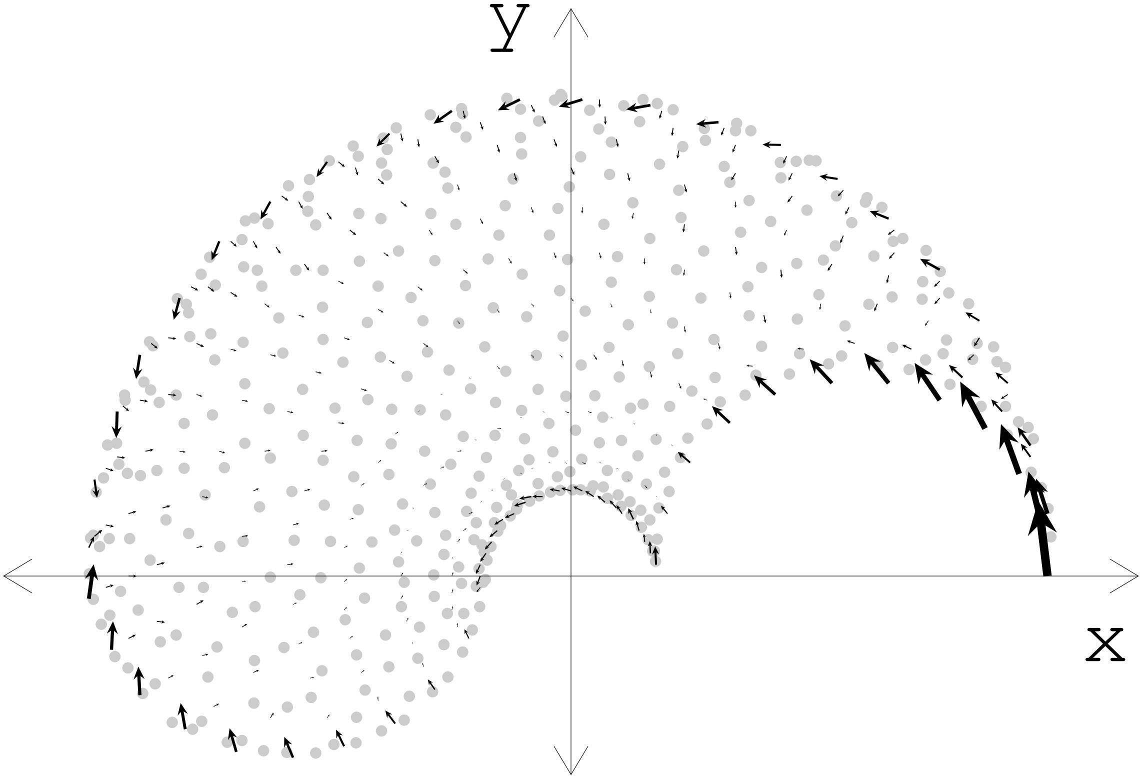


Figure 6.7: The figure shows the magnitude and direction of the approximation errors for v_2 sensor vectors obtained using Formula 6.3. The gray points represent the $\tilde{v}_2$ sensory components of the body icons (same as in Figure 6.2.b). The errors are represented as arrows. The base of each arrow indicates the true position of the sensory vector v_2 for a given query joint vector θ (calculated using forward kinematics). The tip of the arrow represents the approximated position calculated using Formula 6.3.

6.3.2 Achieving Goal Directed Movements

The representation of the robot body schema in terms of body icons can be used for control of goal-directed movements. Robot movements can be specified in Cartesian space and carried out in joint space without the need for inverse kinematics because the mapping between the two spaces is implicit in the way body icons are constructed.

One possible control strategy for moving the robot from one configuration to another is to use gradient ascent (or gradient descent) in a potential field (Morasso and Sanguineti, 1995). The gradient ascent is carried out in a potential field ξ in which the location of the target has a maximum value and all other points are assigned values in proportion to their distance from the target. The potential field is imposed on the $\tilde{\theta}_i$ components of all body icons but is computed based on the $\tilde{v}_i$ components and their distance to the goal in sensor space. The calculation of the potential field is similar to the motor schema approach to robot control (Arkin, 1998; Cameron et al., 1993). In this case, however, the potential field is discretized across all body icons. Each body icon is assigned a value $\tilde{\xi}_i$ which is a sample of the magnitude of the potential field. The global potential field can be approximated using the following equation:

$$\xi(\theta) \approx \sum_i \tilde{\xi}_i U_i(\theta) \tag{6.4}$$

The desired direction of movement for any point in the potential field is given by the corresponding gradient vector field, g, defined as

$$g(\theta) = -\nabla\xi(\theta) \tag{6.5}$$

where ∇ is the gradient operator. Thus, a gradient ascent strategy in the potential field can be performed by integrating the following equation:

$$\dot{\theta} = \gamma\nabla\xi(\theta) \tag{6.6}$$

where γ determines the step size.

Taking advantage of the body icons representation and the form of the softmax activation function, the gradient ascent strategy can be achieved using the following formula derived by Morasso and Sanguineti (1995):

$$\dot{\theta} = \gamma \sum_i (\tilde{\theta}_i - \theta)\tilde{\xi}_i U_i(\theta) \tag{6.7}$$

If more than one constraint is present the potential field can be represented as a combination of several individual fields scaled by appropriate coefficients. In other words,

$$\xi(\theta) = k_1\xi_1(\theta) + k_2\xi_2(\theta) + \ldots \tag{6.8}$$

where k_i are scalars that determine the relative weights of each field.

As an example, consider the task of moving the two-joint robot shown in Figure 6.8 such that the tip of its second limb is positioned over the goal region. In this example, the potential field (ξ^{goal}) is specified as an inverse function of the squared Euclidean distance between $\tilde{v}_2$ and v_{goal} for all body icons, i.e.,

$$\xi^{goal}(\theta) = \frac{1}{\|v_{goal} - v_2(\theta)\|^2} \tag{6.9}$$

The magnitude of each point in the potential field is computed in $\mathcal{S}$-space, but using the body icon representation the field is imposed on $\mathcal{C}$-space. Figure 6.9 shows this process. The final potential field for the configuration in Figure 6.9 is shown in Figure 6.10.a. Its corresponding gradient vector field computed using formula 6.7 is shown in Figure 6.10.b.

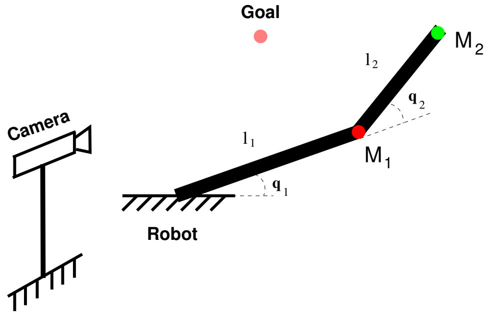


Figure 6.8: The two-joint robot used in the example. The goal is to move the tip of the second limb (body marker M_2) over the goal region.

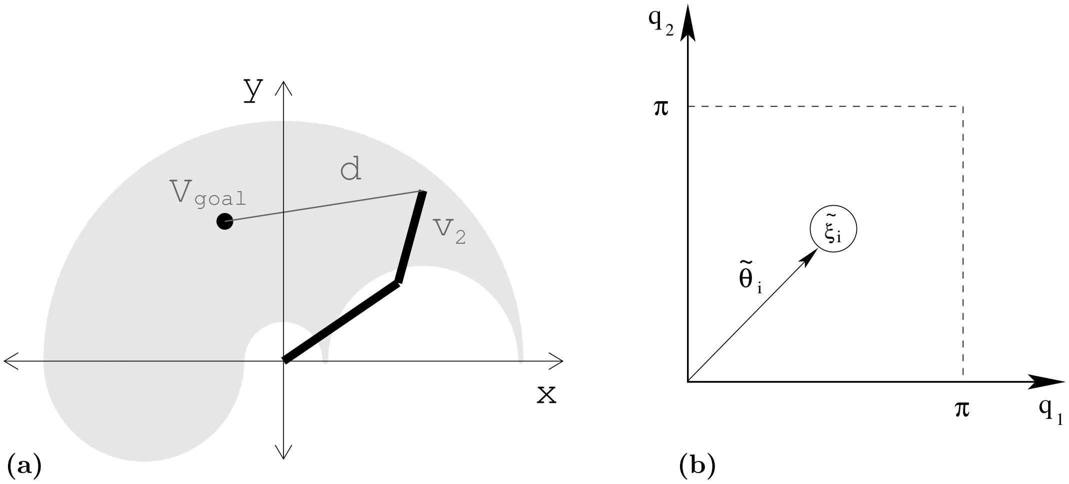


Figure 6.9: Calculation of the potential field. **(a)** For all body icons calculate the distance, d, between v_{goal} and $\tilde{v}_2$. **(b)** To each body icon assign a scalar value $\tilde{\xi}_i$ which is inversely proportional to the squared distance d. In $\mathcal{C}$-space this point is indexed by $\tilde{\theta}_i$. The final potential field is shown on Figure 6.10.a.

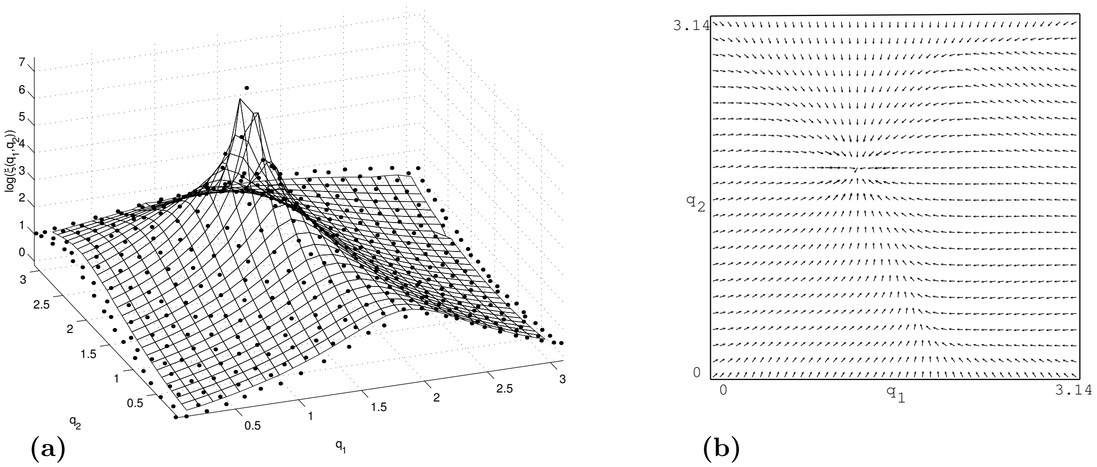


Figure 6.10: **(a)** The resulting potential field for the goal configuration shown on Figure 6.9.a. The surface shows a log plot the approximated field; the dots show the true positions of the discrete samples $\tilde{\xi}_i$. **(b)** The corresponding gradient vector field is approximated with Formula 6.7 (vector magnitudes are not to scale; the arrows have been rescaled to have uniform length in order to show the direction of the entire vector field).

6.4 Identifying Body Frames

Certain tasks can be expressed more naturally in relative frames of reference. For example, a grasping task might be easier to perform if it is expressed in a coordinate frame relative to the wrist than relative to the shoulder. Furthermore, some tasks such as handwriting are performed more easily if the joints of the arm are constrained (e.g., by putting one's arm on the surface of the desk). There is evidence that biological brains maintain and use multiple body frames in order to coordinate body movements (Newcombe and Huttenlocher, 2000; Gallistel, 1999). Gallistel (1999), for example, suggests that intelligent behavior is about learning to coordinate these frames. It is still a mystery, however, how behaviors are expressed and coordinated using multiple body frames.

This section builds upon the self-detection results described in Chapter 5 and shows how coordinate frames that are attached to different body parts of the robot can be identified and constructed automatically. The visual features that were classified as "self" are now clustered into groups based on their co-movement patterns. These clusters of body markers correspond to the rigid bodies that form the body of the robot. Each cluster is used to construct a coordinate frame (also called body frame). These body frames can help simplify the specification of robot behaviors by expressing the positions of body markers in body frames that are more natural for the given task.

6.4.1 Problem Statement

For the sake of clarity the problem of autonomously identifying body frames by a robot will be stated explicitly using the following notation. Let there be a set of visual features $F_{self} = \{f_1, f_2, \ldots, f_m\}$ that the robot can detect and track over time. These features have been autonomously identified by the robot as features belonging to the robot's body (using the method described in Chapter 5 or some other means). The body of the robot consists of a set of rigid bodies $B = \{b_1, b_2, \ldots, b_{n+1}\}$ connected by a set of joints $J = \{j_1, j_2, \ldots, j_n\}$.

The robot can detect the positions of visual features and detect whether or not they are moving at any given point in time. In other words, the robot has a set of perceptual functions $P = \{p_1, p_2, \ldots, p_m\}$, where $p_i(f_i, t) \rightarrow \{0, 1\}$. That is to say, the function p_i

returns 1 if feature f_i is moving at time t, and 0 otherwise.

The goal of the robot is to cluster the set of features into several subsets, F_1, F_2, $\ldots$, F_k such that $F_{self} = F_1 \cup F_2 \cup \ldots \cup F_k$. Furthermore, the clustering must be such that features which belong to the same subset F_j must all lie on the same rigid body. In other words, the features must be clustered according to the rigid bodies on which they lie.

6.4.2 Methodology

The methodology for identifying body frames is similar to the methodology for self-detection. In this case, however, the goal is to detect the *temporal coincidences* (i.e., events which occur at the same time) in the movements of different visual features and not the temporal contingency between motor commands and visual movements. Figure 6.11 gives an example with three visual features (red, green, and blue) and their movement patterns. Feature 1 (red) and feature 2 (green) start moving together within a small interval of time. This movement coincidence is shown as the shaded region in Figure 6.11. Feature 3 (blue) can also be observed to move but the start of its movement is not correlated with the start of the movement of the other two features.

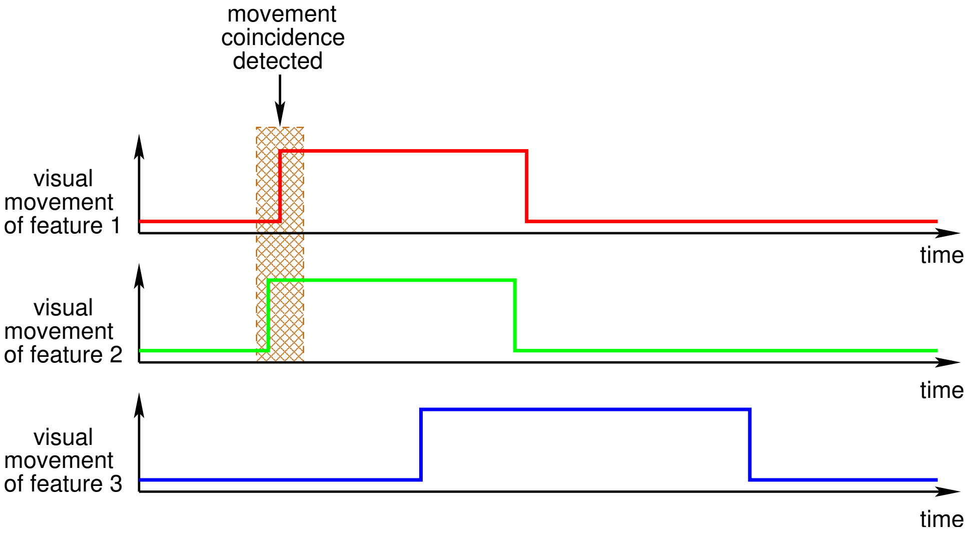


Figure 6.11: The methodology for identifying body frames is based on detecting temporal coincidences in the movements of different features. This figure shows an example with the observed movement patterns of three visual features. Feature 1 (red) and feature 2 (green) start to move within a short interval of time indicated by the shaded region. The start of movement of the third feature (blue) is not correlated with the start of movement of the other two features.

If similar movement coincidences are observed multiple times it is reasonable to conclude that feature 1 and feature 2 lie on the same rigid body because they always start to move together. Furthermore, if the features are also frequently observed to stop moving together (as it is the case in Figure 6.11) the evidence in favor of them lying on the same rigid body would be even stronger.

For every pair of features, f_i and f_j, it is possible to keep a temporal coincidence (or movement coincidence) counter, $C_{i,j}$, which indicates how many times the two features have been observed to start moving together within a short time interval Δt from each other. In the special case when $i = j$, the counter $C_{i,i}$ measures how many times feature f_i was observed to move. The movement coincidence counters can be organized in a 2D matrix as shown in Table 6.3. The matrix is symmetric about the main diagonal because $C_{i,j} = C_{j,i}$.

Table 6.3: Sample movement coincidence matrix. Each entry, $C_{i,j}$, represents a counter indicating how many times feature f_i and feature f_j have been observed to start moving together within a Δt time interval from each other. This matrix is symmetric.

	f_1	f_2	f_3	$\dots$	f_m
f_1	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$	$\dots$	$C_{1,m}$
f_2	$C_{1,2}$	$C_{2,2}$	$C_{2,3}$	$\dots$	$C_{2,m}$
f_3	$C_{1,3}$	$C_{2,3}$	$C_{3,3}$	$\dots$	$C_{3,m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
f_m	$C_{1,m}$	$C_{2,m}$	$C_{3,m}$	$\dots$	$C_{m,m}$

Based on the values of these counters it is possible to calculate two probabilities, $P_{i,j}$ and $Q_{i,j}$, that are similar to the necessity and sufficiency indexes used in the previous chapter. These probabilities are given by the following formulas:

$$P_{i,j} = \frac{C_{i,j}}{C_{i,i}} \qquad\qquad Q_{i,j} = \frac{C_{i,j}}{C_{j,j}}$$

The two probabilities can be calculated from the movement coincidence matrix if the elements in each row are divided by the main diagonal entry in that row. If this operation is performed in place, the resulting matrix will have 1's along the main diagonal, P's above the main diagonal, and Q's below the main diagonal as shown in Table 6.4.

Features f_i and f_j can be clustered together as lying on the same rigid body if both

Table 6.4: This matrix is derived from the matrix shown in Table 6.3 after dividing each entry by the value stored in the diagonal entry in the same row. The P and Q values are described in the text. This matrix is no longer symmetric.

	f_1	f_2	f_3	$\ldots$	f_m
f_1	1.0	$P_{1,2}$	$P_{1,3}$	$\ldots$	$P_{1,m}$
f_2	$Q_{1,2}$	1.0	$P_{2,3}$	$\ldots$	$P_{2,m}$
f_3	$Q_{1,3}$	$Q_{2,3}$	1.0	$\ldots$	$P_{3,m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
f_m	$Q_{1,m}$	$Q_{2,m}$	$Q_{3,m}$	$\ldots$	1.0

$P_{i,j} > T$ and $Q_{i,j} > T$, where T is a threshold value. The threshold value is not arbitrary but is set automatically to $1 - \frac{1}{N}$, where N is the number of independent motor commands (for the CRS robot $N = 7$).

The same calculations can be performed by focusing on the movement coincidences at the end instead of the beginning of visual movements. In this case another matrix similar to the one shown in Table 6.4 can be calculated. The results from the two different types of coincidences can be combined to improve the final results.

The next subsection shows experimental results when this methodology was tested on the robot. The experimental results are presented in the same form as in Table 6.4.

6.4.3 Experimental Results

The methodology described above was tested with the datasets described in Section 5.8. The datasets were color processed in the same way as described in the previous chapter. Because the timing coincidence calculations are more sensitive to sensory noise, however, the positions of the color markers were averaged over three consecutive frames and marker movements shorter than six frames (1/5-th of a second) were ignored. The body frames that were identified are shown in Figure 6.12. The clustering results used to form these frames are shown in Table 6.5 and Table 6.6 and are described below.

Table 6.5 shows results for the first dataset in which the robot is the only moving object in the environment (see Section 5.8.1). The results are presented in the form shown in Table 6.4. Based on these results three pairs of markers emerge as shown with the

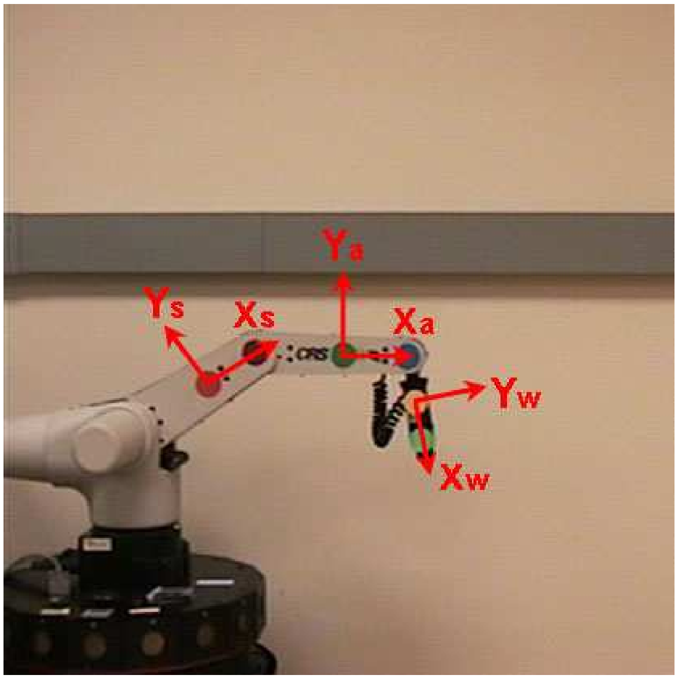


Figure 6.12: The figure shows three different body frames: shoulder frame (X_s,Y_s) formed by markers M0 and M1; arm frame (X_a, Y_a) formed by markers M2 and M3; and wrist frame (X_w, Y_w) formed by markers M4 and M5. The three frames are constructed from the robot's body markers after the markers have been clustered based on their co-movement patterns. Table 6.5 and Table 6.6 show the clustering results used to form these frames.

highlights: {M0, M1}, {M2, M3}, and {M4, M5}. For these pairs of markers both $P_{i,j}$ and $Q_{i,j}$ are greater than the threshold T, which was set to 0.857 (i.e., $1 - \frac{1}{7}$, where 7 is the number of different motor commands). The time interval Δt was set to 0.5 seconds.

Similarly, Table 6.6 displays the results for the same dataset when movement coincidences at the end of visual movements were considered. The same pairs of markers emerge in this case as well: {M0, M1}, {M2, M3}, and {M4, M5}.

Similar results were obtained for all other datasets described in Section 5.8. For all five datasets the algorithm clustered correctly the six body markers into three groups of two: {M0, M1}, {M2, M3}, and {M4, M5}.

Based on these results three body frames can be formed by the three pairs of body markers as shown in Figure 6.12. In 2D a coordinate frame can be specified with just two different points. The first point serves as the origin of the frame. The X axis of the frame is determined by the vector from the first point to the second point. The Y axis is given

Table 6.5: The highlights show three pairs of markers grouped based on their *start* of movement coincidences. They correspond to the rigid bodies of the robot's shoulder, arm, and wrist.

	M0	**M1**	**M2**	**M3**	**M4**	**M5**
M0	1.000	0.996	0.996	0.996	0.996	0.996
M1	0.889	1.000	0.989	0.992	0.989	0.989
M2	0.518	0.576	1.000	0.911	0.911	0.911
M3	0.550	0.614	0.967	1.000	0.998	0.995
M4	0.468	0.520	0.823	0.849	1.000	0.992
M5	0.467	0.519	0.821	0.845	0.990	1.000

Table 6.6: The highlights show three pairs of markers grouped based on their *end* of movement coincidences. They correspond to the rigid bodies of the robot's shoulder, arm, and wrist.

	M0	**M1**	**M2**	**M3**	**M4**	**M5**
M0	1.000	0.996	0.991	0.996	0.996	0.996
M1	0.889	1.000	0.981	0.992	0.992	0.992
M2	0.516	0.571	1.000	0.913	0.913	0.915
M3	0.550	0.614	0.969	1.000	1.000	0.995
M4	0.468	0.522	0.825	0.851	1.000	0.992
M5	0.467	0.521	0.825	0.845	0.990	1.000

by a vector perpendicular to the X vector and oriented in the positive (counterclockwise) direction. The next two subsections show how body frames can be used to learn nested RBS representations and to encode robot behaviors.

It is interesting to point out that the method described here is capable of segmenting the structure of rigid and articulated objects even if they don't belong to the robot's body. For example, Table 6.7 provides results for the dataset with two robots with uncorrelated movements described in Section 5.8.3. In this case six pairs of markers were identified. They correspond to the rigid bodies of the first and the second robot. Similar results are obtained when the end of movement coincidences are considered (see Table 6.8).

These results suggest a way to segment a visual scene into individual objects. In computer vision it can be difficult to identify which visual features constitute an object. However, if an object is defined as a set of visual features that always start and stop moving together then the method described above may be used for object segmentation, especially if the robot is allowed to push and move the objects. Future work should test this hypothesis.

Table 6.7: Six pairs of markers identified based on their *start* of movement coincidences for two robots with uncorrelated movements (see Section 5.8.3). The six pairs of markers correspond to the rigid bodies of the two robots.

	M0	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**	**M9**	**M10**	**M11**
M0	1.000	0.996	0.991	0.996	0.996	0.996	0.112	0.124	0.215	0.202	0.240	0.236
M1	0.889	1.000	0.985	0.992	0.989	0.989	0.092	0.103	0.207	0.211	0.249	0.249
M2	0.516	0.574	1.000	0.911	0.911	0.911	0.094	0.103	0.205	0.203	0.234	0.232
M3	0.550	0.614	0.967	1.000	0.998	0.995	0.090	0.097	0.206	0.209	0.239	0.237
M4	0.468	0.520	0.823	0.849	1.000	0.992	0.087	0.095	0.198	0.194	0.228	0.224
M5	0.467	0.519	0.821	0.845	0.990	1.000	0.089	0.095	0.195	0.193	0.227	0.223
M6	0.115	0.106	0.186	0.168	0.190	0.195	1.000	0.996	0.987	0.991	0.991	0.996
M7	0.116	0.108	0.183	0.163	0.187	0.187	0.896	1.000	0.980	0.980	0.988	0.984
M8	0.113	0.122	0.208	0.196	0.221	0.219	0.503	0.555	1.000	0.948	0.948	0.950
M9	0.109	0.127	0.211	0.204	0.222	0.222	0.519	0.569	0.972	1.000	1.000	0.998
M10	0.108	0.126	0.203	0.195	0.219	0.219	0.433	0.480	0.812	0.836	1.000	0.996
M11	0.105	0.124	0.198	0.191	0.212	0.212	0.429	0.471	0.803	0.823	0.983	1.000

Table 6.8: Six pairs of markers identified based on their *end* of movement coincidences for two robots with uncorrelated movements (see Section 5.8.3). The six pairs of markers correspond to the rigid bodies of the two robots.

	M0	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**	**M9**	**M10**	**M11**
M0	1.000	0.996	0.987	0.996	0.996	0.996	0.103	0.107	0.219	0.227	0.240	0.249
M1	0.889	1.000	0.977	0.992	0.992	0.992	0.088	0.096	0.188	0.192	0.230	0.238
M2	0.513	0.569	1.000	0.911	0.911	0.913	0.076	0.089	0.174	0.167	0.212	0.217
M3	0.550	0.614	0.967	1.000	1.000	0.995	0.081	0.095	0.185	0.185	0.227	0.235
M4	0.468	0.522	0.823	0.851	1.000	0.992	0.085	0.091	0.173	0.173	0.220	0.224
M5	0.467	0.521	0.823	0.845	0.990	1.000	0.087	0.097	0.173	0.173	0.219	0.219
M6	0.106	0.102	0.150	0.150	0.186	0.190	1.000	0.991	0.982	0.987	0.987	0.991
M7	0.100	0.100	0.159	0.159	0.179	0.191	0.892	1.000	0.972	0.976	0.984	0.984
M8	0.115	0.111	0.176	0.176	0.194	0.194	0.501	0.551	1.000	0.950	0.950	0.950
M9	0.123	0.116	0.174	0.181	0.199	0.199	0.516	0.567	0.975	1.000	1.000	0.998
M10	0.108	0.116	0.184	0.186	0.211	0.211	0.431	0.478	0.814	0.836	1.000	0.996
M11	0.111	0.118	0.185	0.189	0.212	0.208	0.427	0.471	0.803	0.823	0.983	1.000

6.4.4 Nested RBS Representation

Once the body frames are identified, the positions of the body markers can be expressed in different body frames and not only in a camera-centric frame.

For example, Figure 6.13 shows 500 observed positions of the light green marker (M5) relative to the arm frame (X_a, Y_a). These data points are the same data points shown in Figure 6.5.f but their coordinates are expressed in the arm frame and not in the camera-centric frame. This new view of the data clearly shows the possible positions of the wrist marker relative to the arm. The locations of the dark green (M2) and dark blue (M3) markers, which determine the coordinate frame, are shown as large circles.

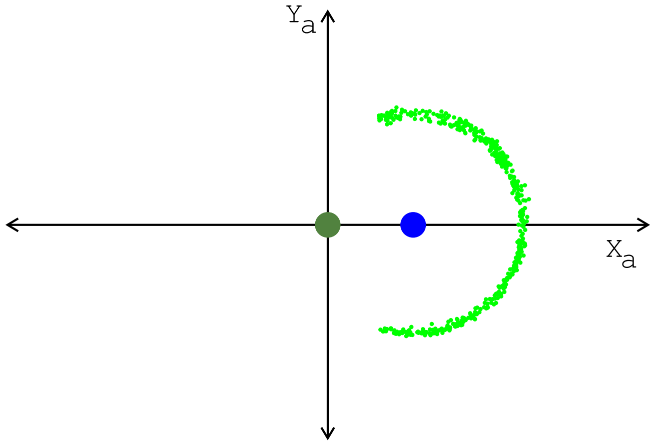


Figure 6.13: The figure shows 500 observed positions of the green body marker (M5) when its coordinates are expressed in the arm body frame (X_a, Y_a) and not in the camera-centric frame as shown in Figure 6.5.f. The circular pattern clearly shows the possible positions of the wrist relative to the arm.

It is possible to take this approach one step further and to define a nested RBS representation. For example, a separate body icons table can be learned only for the two wrist markers (M4 and M5). The joint vectors of these body icons will contain only joint angles of the wrist joint. The sensory vectors, on the other hand, will contain only the positions of the wrist markers. An important detail in this case, however, is that the visual coordinates of the two wrist markers must be expressed in arm frame coordinates and not in camera centric coordinates. In other words, the body icons table will have the following format:

Table 6.9: Body icons table for the wrist only. Each row of the table represents the observed joint and sensory vectors for a specific wrist pose and the observed positions of the two wrist markers, M4 and M5, calculated in arm frame coordinates.

#	**Kinematic components**	**Sensory components**
1	$\tilde{\theta}_1 = \{\tilde{q}_w^1\}$	$\tilde{V}_1 = \{\tilde{v}_4^1, \tilde{v}_5^1\}$
2	$\tilde{\theta}_2 = \{\tilde{q}_w^2\}$	$\tilde{V}_2 = \{\tilde{v}_4^2, \tilde{v}_5^2\}$
...	...	...
$\mathcal{I}$	$\tilde{\theta}_{\mathcal{I}} = \{\tilde{q}_w^{\mathcal{I}}\}$	$\tilde{V}_{\mathcal{I}} = \{\tilde{v}_4^{\mathcal{I}}, \tilde{v}_5^{\mathcal{I}}\}$

The nested RBS representation has several advantages. First, it reduces the dimensions of the sensory and joint vectors and thus requires less memory to store all body icons. Second, because of the reduced dimensionality a smaller number of body icons must be learned before a certain level of approximation accuracy can be reached. Finally, the nested representation can be used to specify different control laws or behaviors for different joint groups of the robot.

The next subsection shows an example in which the wrist and the arm are controlled independently.

6.4.5 Behavioral Specification Using a Nested RBS

The body markers and the body frames provide a convenient way for specification, monitoring, and execution of robot behaviors. This section demonstrates how robot behaviors can be encoded using a nested RBS.

The RBS model provides a blueprint of the possible configurations of the robot's body. This blueprint can be used to localize individual body parts in space. It can also be used for control and planning of body movements. These movements can be specified as the desired final positions of specific body markers relative to other visual features.

Because the robot has access to the sensorimotor stimuli that are produced by its own body its behaviors can be expressed in terms of desired final positions for some of these stimuli. This approach is consistent with the ideas of Berthoz (2000) who argues that even the reflex behaviors of living organisms are encoded in terms of their body schemas.

Complex behaviors can be constructed by combining multiple primitive behaviors. Typically the primitive behaviors are sequenced or fused together. As an example, consider the grasping behavior composed of six steps shown in Figure 6.14. The behavior is specified with a finite state automaton (FSA). The six states of the FSA are linked by perceptual triggers (Arkin, 1998) that determine when the robot should switch to the next state.

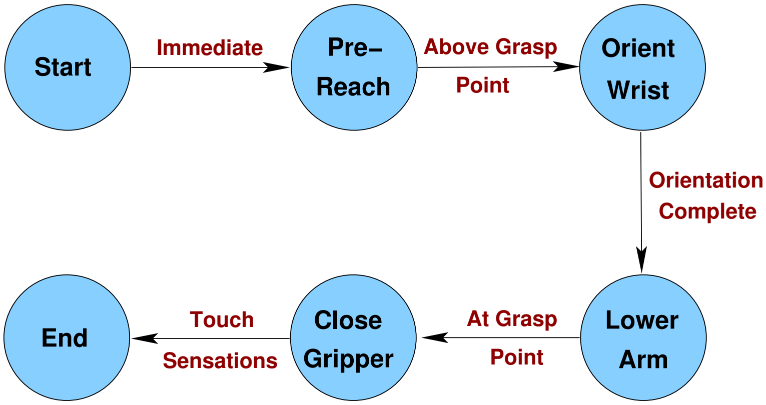


Figure 6.14: A finite state automaton (FSA) describing a grasping behavior. The six states of the FSA are linked by perceptual triggers that determine when the robot should switch to the next state.

Figures 6.15, 6.16, and 6.17 show the sequence of movements as the simulated robot attempts to grasp a stick object. During the *pre-reach* phase the position of the yellow body marker is controlled in such a way as to position the marker above the target grasp point (Figure 6.15). During the *orient wrist* phase only the wrist is moved relative to the arm frame (Figure 6.16). The green spheres in Figure 6.16 show the sensory components of the body icons for the light green wrist marker relative to the arm frame. The red spheres show the most highly activated body icons that are closest to the grasp point. Next, the arm is lowered by controlling both the position of the yellow marker and the light green marker so that the wrist remains perpendicular to the table (Figure 6.17). In other words, the positions of two different body markers are controlled in two different body frames simultaneously. Finally, the gripper is closed until the simulated tactile sensors are triggered. This completes the grasping behavior.

A similarly encoded grasping behavior is used in Section 6.6 with the real robot. The behavior is hand-coded and not learned. Learning of behaviors using the RBS representation is a good topic for future work.

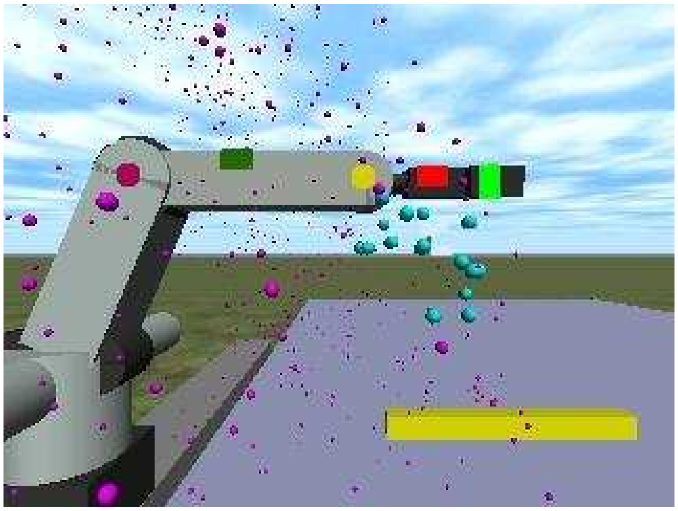


Figure 6.15: Example of the *pre-reach* behavior in progress. The purple spheres represent the sensory components of the yellow marker for all body icons. The most highly activated components are colored in cyan and are clustered above the target grasp point on the stick.

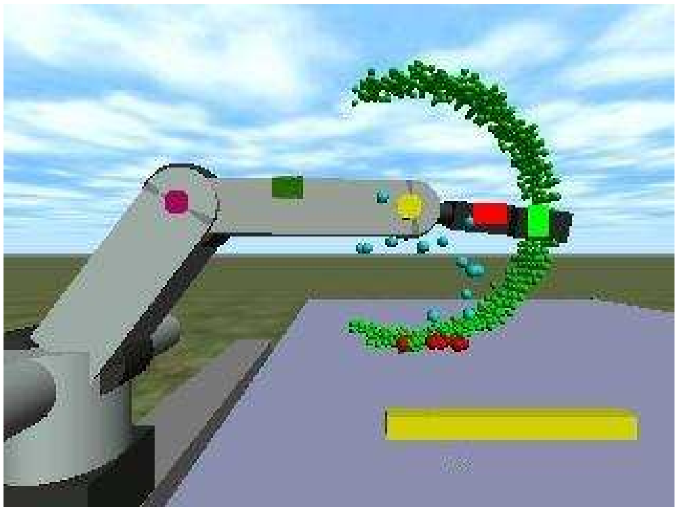


Figure 6.16: Example of the *orient-wrist* behavior in progress. Once the arm is positioned above the grasp point the wrist of the robot is moved relative to the arm frame. The green spheres show the possible positions of the light green marker relative to the arm frame. The red spheres correspond to the wrist positions of the most highly activated body icons.

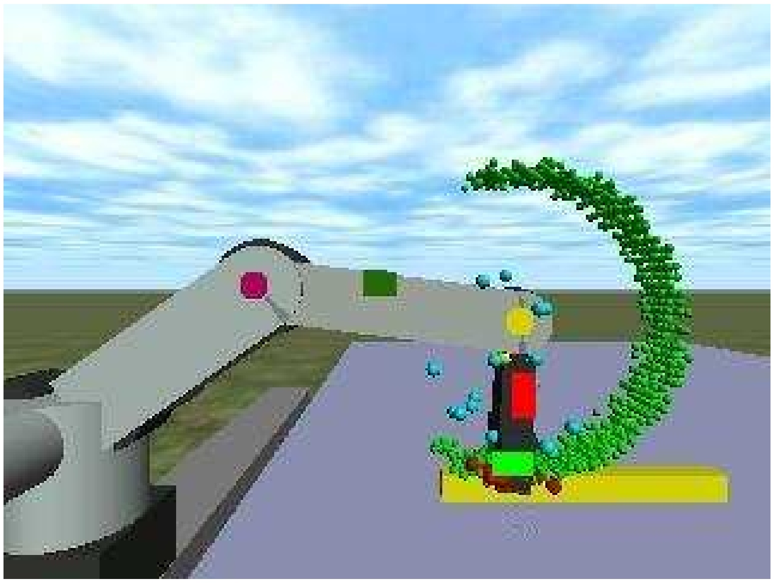


Figure 6.17: Example of the *lower-arm* behavior in progress. The behavior controls the positions of both the yellow marker and the light green marker. As a result, the arm is lowered toward the grasp point while the wrist is rotated so that it remains perpendicular to the table. The positions of two body markers are controlled simultaneously in two different body frames using two separate sets of body icons.

6.5 Extending the Robot Body Schema

The body schema model described so far is not pliable, i.e., it cannot be extended in order to assimilate an external object into the robot's body representation. This section modifies that model to have extensibility properties similar to those of the biological body schema.

Iriki et al. (1996) trained monkeys to use rake-shaped tools to extend their reach. They reported that during tool use the bimodal neurons which encode the monkey's body schema change their visual receptive fields such that they begin to fire in the expanded area reachable by the tool. There are two main criteria that must be met before the extension of the body representation can take place. First, the movements of the tool must be temporally correlated with the movements of the monkey's arm. Even a small variation or an unpredictable time delay can disrupt this process (Iriki, 2003). Second, there must be an intention to use the object as a tool; just holding the object is not sufficient to trigger an extension of the body (Iriki et al., 1996, 2001).

The model described here was inspired by the work of Iriki et al. (1996, 2001). Their experiments with monkeys, however, serve only as an inspiration to the robotics work. While the robot was capable of replicating some of their experiments, the robotics work does not attempt to model how the monkey's brain works. Instead, the goal of this work is to test, at the functional level, some of the building blocks that form the complex system for representing the body of a robot and how the body representation can be extended and remapped depending on task demands.

The basic idea behind the extension method is to morph the body representation of the robot so it is better suited for a task that requires the use of an external object. For example, if the robot has to control the position of the tip of a stick object it might be more convenient to perform this operation if the stick is treated as an extension of the robot's wrist. The alternative approach would be to treat the stick as different from the body and to reconcile two different control strategies.

The extension or morphing mechanism is triggered by near-perfect temporal coincidence between self movements and external object movements. The robot needs to detect whether the movements of the stick coincide with the movements of the robot's wrist. If the answer is yes, then the markers of the stick should move with the same efferent-afferent delay of the robot. Because of that, the stick can be assimilated into the robot's body schema for the duration of its usage. If there are any unpredictable or variable time delays then the external object will not be assimilated into the robot's body representation. Thus, tools that are not solid, e.g., ropes and rubber hoses will not trigger an extension of the body. This is a limitation of the current approach as it only applies to rigid bodies. However, Japanese monkeys have similar limitations as reported by Iriki (2003).

After the extension mechanism is triggered it is necessary to calculate the parameters of the body transformation. This transformation may involve rotation, translation, and scaling. These parameters are calculated based on the observed spatial relation between the external object and a corresponding body part. If such observation data is not available from prior experience it can be gathered after a few motor babbling commands (10 were used in the robot experiments reported below). These observations produce two sets of points and the corresponding one-to-one mapping between them. In other words, for each of the 10 observed body poses there is a corresponding pose of the object targeted for assimilation. Based on this information the robot calculates the parameters of the transformation between the two. Once the transformation parameters are calculated the transformation is applied to all body icons in the affected body schema (there could be more than one body schema involved as they could be nested).

The RBS model represents the positions of the robot's body markers in various body poses as a point cloud formed by the visual components of the body icons. The main idea behind the extension method is to morph the positions of the visual components of the body icons but to keep the kinematic components the same. Because the visual components are represented with a point cloud, applying the transformation is equivalent to transforming each of these points individually. After the transformation is completed, the transformed point cloud can be used to control robot movements in the same way as before.

The following two subsections show two examples of body extensions triggered by a tool and a TV image. The next section explains how the extended body representation can be used to achieve video-guided behaviors.

6.5.1 Example 1: Extension Triggered by a Tool

This example demonstrates how the robot can establish a temporary equivalence between some of its own body markers (frames) and the markers of an external object. This equivalence is based on the movement coincidences between the two sets of markers. The mapping between the two is temporary and can only last for as long as the robot is grasping the tool. If the tool is dropped the movement coincidence will stop and this will violate the first condition for extension identified by Iriki et al. (1996). Given that the movement coincidence is present, the mapping is used next to morph the body representation to include the attached tool. The experimental procedure is described next.

Figure 6.18 shows six frames from a short sequence (3500 frames or less than 2 minutes) in which the robot waves a stick. Each frame shows an intermediary body pose at which the robot stops for a short period of time before performing the next movement. The robot performed the movements shown in Figure 6.18 four times. This resulted in a total of 20 wrist movements and 8 arm movements (4 up and 4 down).

The stick object has two color markers which will be referred to as S0 and S1. Their colors are: red (S0) and orange (S1). The two markers are placed at the tip of the stick and at its mid-point, respectively. As in the previous chapter, the robot can see a point-light display of the movements of different color markers.

The color segmentation results for the frames shown in Figure 6.18 are presented in Figure 6.19. Table 6.10 shows the *start* of movement coincidence results between the robot's body markers and the markers of the stick. The cells highlighted in green indicate that the stick markers, S0 and S1, move in the same way as the wrist markers, M4 and M5. The cells highlighted in cyan correspond to the arm frame. Because the shoulder of the robot does not move much in this short sequence the two shoulder markers, M0 and M1, are not grouped together (this was not the case for the longer sequences described in the previous section). Similarly, Table 6.11 shows the results for the *end* of movement coincidences. Once again, the markers of the stick have the same movement patterns as the markers of the wrist.

These results indicate that even after a few body movements it is possible to identify whether an external object can be reliably controlled by the robot. Furthermore, it is possible to identify the body frame to which the object is attached (in this case the wrist frame).

Once the mapping is established the next step is to perform the actual extension where the position of one or more of the robot's body markers are temporarily transformed to coincide with the positions of the new stick markers. Section 6.6 describes how the transformation parameters can be calculated.

Table 6.10: *Start* of movement coincidence results for a short sequence in which the robot waves a stick tool (see Figure 6.18). The entries highlighted in green show that the two stick markers, S0 and S1, start to move at the same time as the two wrist markers, M4 and M5. The shoulder of the robot does not move much in this short sequence and therefore markers M0 and M1 are not grouped together. The two arm markers, M2 and M3, are grouped together after only 8 movements.

	M0	**M1**	**M2**	**M3**	**M4**	**M5**	**S0**	**S1**
M0	1.000	1.000	1.000	1.000	0.000	1.000	0.000	1.000
M1	0.091	1.000	0.455	0.182	0.182	0.182	0.091	0.273
M2	0.125	0.625	1.000	1.000	1.000	1.000	0.875	1.000
M3	0.125	0.250	1.000	1.000	1.000	1.000	0.875	1.000
M4	0.000	0.100	0.400	0.400	1.000	0.950	0.950	0.950
M5	0.050	0.100	0.400	0.400	0.950	1.000	0.950	1.000
S0	0.000	0.050	0.350	0.350	0.950	0.950	1.000	0.950
S1	0.050	0.150	0.400	0.400	0.950	1.000	0.950	1.000

Table 6.11: *End* of movement coincidence results for the short sequence in which the robot waves a stick tool (see Figure 6.18). The results are similar to the ones shown in Table 6.10.

	M0	**M1**	**M2**	**M3**	**M4**	**M5**	**S0**	**S1**
M0	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
M1	0.091	1.000	0.182	0.091	0.091	0.091	0.091	0.091
M2	0.000	0.250	1.000	1.000	1.000	1.000	1.000	1.000
M3	0.000	0.125	1.000	1.000	1.000	1.000	1.000	1.000
M4	0.000	0.050	0.400	0.400	1.000	0.950	0.950	0.950
M5	0.000	0.050	0.400	0.400	0.950	1.000	1.000	1.000
S0	0.000	0.050	0.400	0.400	0.950	1.000	1.000	1.000
S1	0.000	0.050	0.400	0.400	0.950	1.000	1.000	1.000

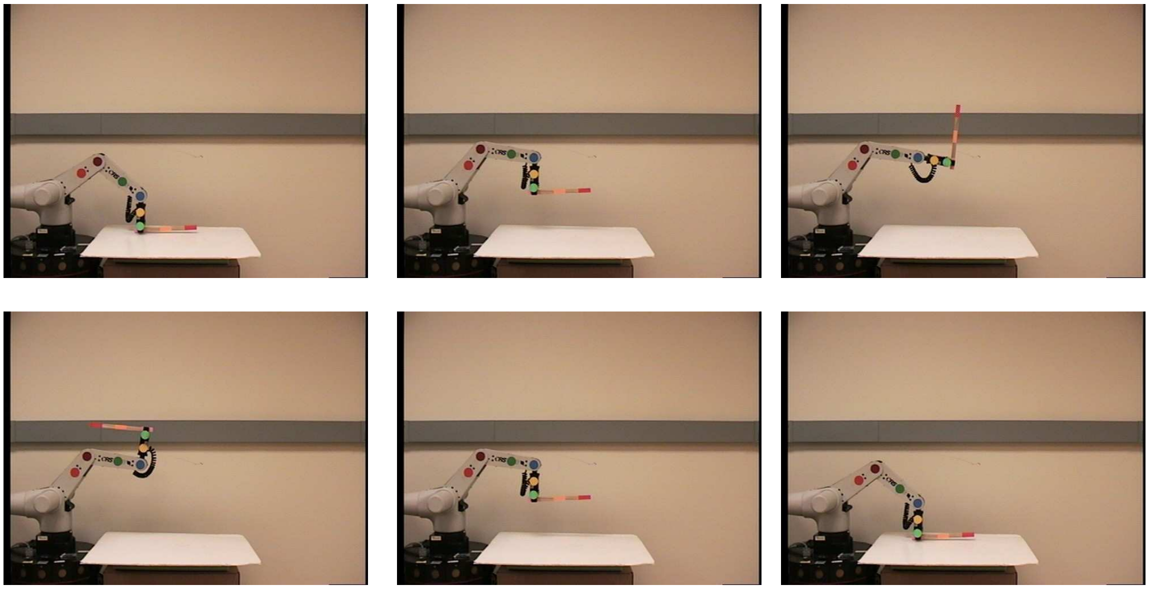


Figure 6.18: Frames from a short sequence (less than 2 minutes) in which the robot waves a stick tool. The stick object has two color markers which can be detected by the robot.

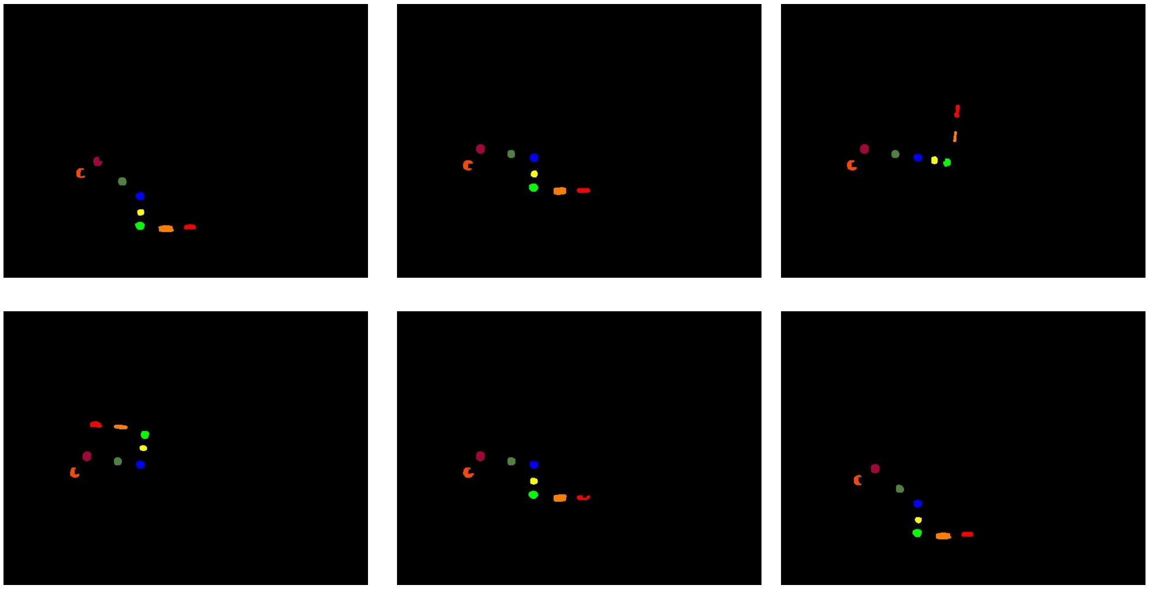


Figure 6.19: Color segmentation results for the robot poses shown in Figure 6.18.

6.5.2 Example 2: Extension Triggered by a Video Image

This section shows that the movement coincidences between two sets of markers can be used successfully to establish marker correspondence. Once the correspondence is established it can be used to remap the positions of the robot's body icons onto the TV monitor. The TV sequence described in Section 5.10 is used for this example.

Chapter 5 showed how the robot can detect that the TV markers must be associated with its own body. The methodology described in that chapter, however, did not address the problem of mapping between the TV markers and the robot's markers. In other words, it did not address the issue whether the wrist marker in the TV image really corresponds to the robot's own wrist marker. Solving this problem is important because the colors of the markers in the TV may be slightly different from their real-world counterparts (which was indeed the case in the TV experiments).

In the previous example (Section 6.5.1) the robot was in direct physical contact with the external object before the body extension could be triggered. In the second example described here physical contact is neither possible nor required (see Figure 6.20). This is not an obstacle since the extension of the body is triggered by a movement coincidence detector instead of physical contact detector.

Table 6.12 shows the *start* of movement coincidence results for the TV sequence. Figure 6.20 shows an alternative way to visualize these results. The results indicate that the body frames of the robot in the TV image were mapped correctly to the body frames of the real robot. The only exception is the yellow marker as its position detection is very noisy because its color is very similar to the color of the background wall in the TV image. This omission is not critical for the rest of the TV experiments because the yellow marker is not used to encode any of the robot behaviors (e.g., grasping). Some matching ambiguities remain because the timing coincidences method establishes correspondences between body frames and not individual body markers (see Figure 6.20). These are resolved by matching to the nearest marker in feature space (in this case HSV color space). Similar results were obtained for two other TV sequences. The next section builds on these results and describes the remaining details of the extension process.

Table 6.12: Mapping between body frames and TV frames based on *start* of movement coincidence results for the TV sequence. The highlighted areas show the body markers and the TV markers that were grouped together. Only the yellow TV marker could not be matched with any of the real markers because of position detection noise. The results are corrected for marker visibility. Similar results were obtained for two other TV sequences.

	M0	**M1**	**M2**	**M3**	**M4**	**M5**	**TV0**	**TV1**	**TV2**	**TV3**	**TV4**	**TV5**
M0	1.000	0.996	1.000	0.987	0.987	0.982	0.987	0.991	0.899	0.933	0.855	0.921
M1	0.886	1.000	0.980	0.980	0.980	0.976	0.922	0.984	0.885	0.924	0.866	0.899
M2	0.563	0.620	1.000	0.995	0.993	0.988	0.566	0.618	0.880	0.929	0.902	0.893
M3	0.553	0.617	0.990	1.000	0.998	0.990	0.560	0.607	0.883	0.929	0.906	0.897
M4	0.456	0.509	0.813	0.821	1.000	0.992	0.462	0.499	0.714	0.768	0.895	0.871
M5	0.452	0.505	0.807	0.813	0.988	1.000	0.458	0.495	0.710	0.760	0.891	0.873
TV0	0.941	0.987	0.958	0.954	0.954	0.945	1.000	0.983	0.877	0.882	0.824	0.905
TV1	0.872	0.973	0.965	0.957	0.953	0.946	0.907	1.000	0.877	0.905	0.841	0.901
TV2	0.496	0.552	0.857	0.862	0.854	0.846	0.512	0.549	1.000	0.886	0.838	0.815
TV3	0.550	0.615	0.982	0.989	0.989	0.982	0.565	0.615	0.975	1.000	0.930	0.925
TV4	0.298	0.338	0.544	0.548	0.660	0.655	0.305	0.334	0.534	0.520	1.000	0.587
TV5	0.446	0.495	0.784	0.791	0.937	0.934	0.467	0.509	0.772	0.771	0.919	1.000

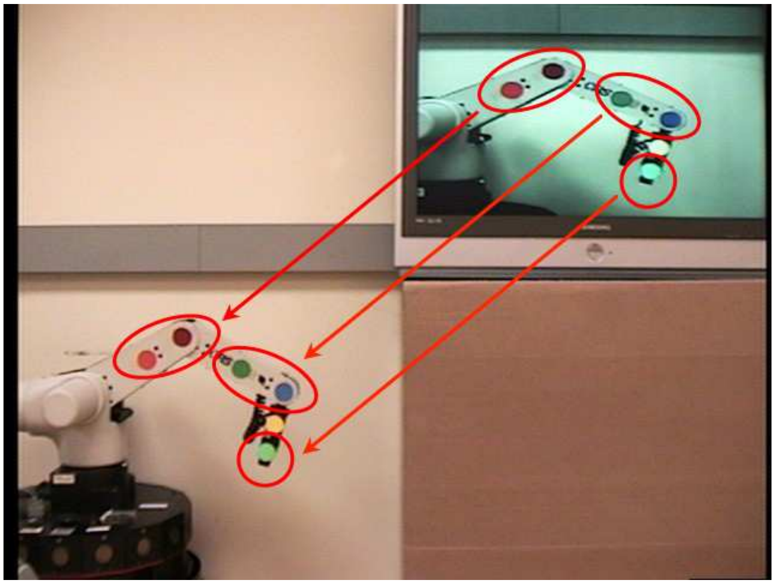


Figure 6.20: Visual representation of the matching markers based on start of movement coincidence results from Table 6.12.

6.6 Achieving Video-Guided Behaviors

Humans are capable of performing many behaviors in which they receive visual feedback about their own actions only through indirect means, e.g., through a mirror reflection or a real-time video image. Some examples include: getting dressed in front of the mirror, driving a car in reverse using the rear view mirrors, playing a video game using a joystick to control a virtual character, and using a mouse to position the mouse pointer on a computer monitor. Behaviors like these are so common that we perform many of them on a daily basis without even thinking about their complexity.

Some primates are also capable of performing similar behaviors. For example, consider the task shown in Figure 6.21 which is described by Iriki et al. (2001). The hands of the monkey and the incentive object (a piece of apple) are placed under an opaque panel such that they cannot be observed directly by the monkey. In order to reach and grasp the incentive object the monkey must use the real-time video feedback of its own movements captured by a camera and projected on a TV monitor.

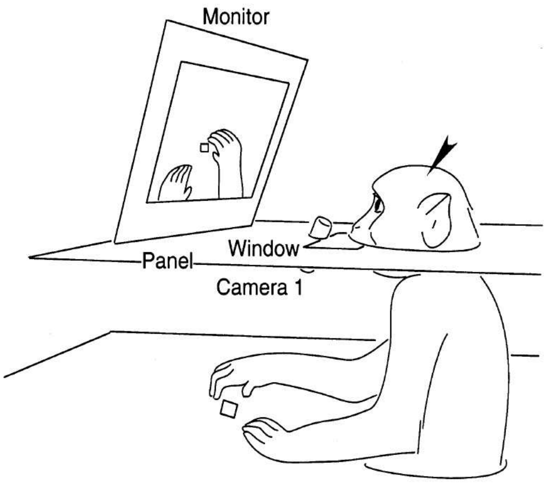


Figure 6.21: The figure shows the experimental setup that was used by Iriki et al. (2001). The setup consists of a TV monitor that displays real-time images captured by the camera. An opaque panel prevents the monkey from observing the movements of its hands directly. Instead, it must use the TV image to guide its reaching behaviors in order to grasp the food item. During the initial training phase a transparent window located close to the eye level of the monkey was left open so that it can observe the movements of its hands directly as well as in the TV monitor. From Iriki et al. (2001).

To solve this problem the monkey must solve at least three sub-problems. First, it must realize that the TV monitor displays a real-time video of its own hands and not, say, a recording of the movements of another monkey. Second, the monkey must figure out the similarity transformation (translation, rotation, and scaling) between the position of its real arm (estimated from proprioceptive informations as it cannot be seen directly) and the image of the arm in the TV monitor. Finally, the monkey must use the video image to guide its hand toward the incentive object.

This section describes a computational framework that was used successfully by a robot to solve the task described above. The robot solves the first sub-problem by detecting the temporal contingency between its own motor commands and the observed self movements in the video image. As the video image is projected in real time the visual self-movements detected in it occur after the expected proprioceptive-to-visual efferent-afferent delay of the robot (see Chapter 5). The second sub-problem is solved by estimating the similarity transformation (translation, rotation, and scaling) between two sets of points. The first set consists of the positions of specific body locations which are estimated from proprioceptive information as they cannot be observed directly. The second set consists of the observed positions of the same body locations in the video. Once the similarity transformation is calculated the robot can extend its body schema and use it to perform the grasping movement without direct visual feedback of its own body. The setup for the robot experiments is described in Section 6.6.2, after the related work with animals is presented.

6.6.1 Similar Experiments with Animals

Menzel et al. (1985) reported for the first time the abilities of chimpanzees to perform video-guided reaching behaviors. Their experimental setup is shown in Figure 6.22. The chimpanzees in their study were also capable of detecting which of two TV monitors shows their self image and which shows a recording from a previous trial. They succeeded even when the TV image was rotated by 180°.

Experiments in which the visual feedback comes from a mirror instead of a video image have also been performed. Itakura (1987) reported that Japanese monkeys can reach for

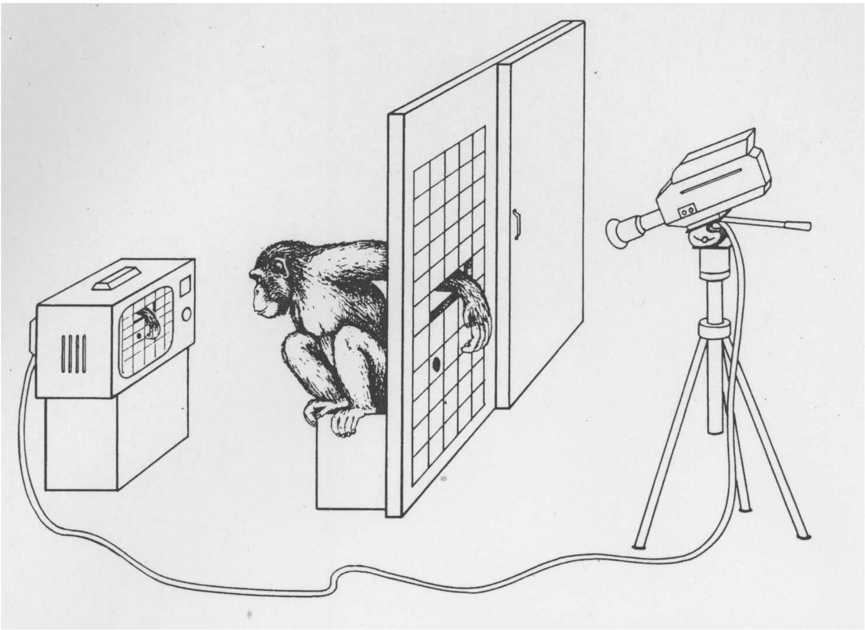


Figure 6.22: The experimental setup used by Menzel et al. (1985).

targets that can only be observed in the mirror image. Epstein et al. (1981) were able to train pigeons to peak a spot on their body that could be seen only in a mirror. After Gallup's discovery that chimpanzees can self-recognize in the mirror (Gallup, 1970) there has been a flood of studies that have used mirrors in primate experiments. These studies are far too numerous to be summarized here. See (Barth et al., 2004) for a comprehensive summary.

More recently, Iriki et al. (2001) have performed reaching experiments with Japanese monkeys (see Figure 6.21) while simultaneously recording the firing patterns of neurons located in the intraparietal sculus that are believed to encode the body schema of these monkeys. Their results show that these neurons, which fire when the monkey observes its hand directly, can be trained to fire when the hand is observed in the TV image as well. Furthermore, they showed that the visual receptive fields of these neurons shift, expand, and contract depending on the position and magnification of the TV image. In order to learn these skills, however, the "monkey's hand-movement had to be displayed on the video monitor without any time delay [...] the coincidence of the movement of the real hand and the video-image of the hand seemed to be essential" (Iriki et al., 2001, p. 166).

This section was inspired by the work of Iriki et al. (2001) which describes some ingenuous experiments with Japanese monkeys that have learned to solve the task shown in Figure 6.21. Their study, however, serves only as an inspiration to the robotics work. While our robot was capable of replicating some of the experiments reported in (Iriki et al., 2001) this section does not attempt to model how the brain works. Instead, the goal of this section is to study, at the functional level, the building blocks that form the complex system for representing the body of a robot and how the body representation can be extended and remapped depending on the current task.

6.6.2 Experimental Setup

The experimental setup for the robot experiments (which are described below) is shown in Figure 6.23. All experiments were performed using the CRS+ A251 manipulator arm described in Chapter 4. The robot has 5 degrees of freedom (waist roll, shoulder pitch, elbow pitch, wrist pitch, wrist roll) plus a gripper. For the purposes of the current experiments, however, the two roll joints were not allowed to move away from their 0° positions. In other words, the movements of the robot were restricted to the vertical plane. The mobile base of the robot was disabled and remained fixed during these experiments.

The experimental setup uses 2 cameras. The first camera (Sony EVI-D30) is the only camera through which the robot receives its visual input. The image resolution was set to 640x480. The second camera (Sony Handycam DCR-HC40) was placed between the robot and the first camera such that it can capture approximately $\frac{1}{3}$ of the working envelope of the robot. The frames captured by the second camera were displayed in real-time on a TV monitor (Samsung LTN-406W 40-inch LCD Flat Panel TV).

As in the previous experiments, six color markers were placed on the robot's body. The positions of the markers were tracked with a computer vision code which performs histogram matching in HSV color space using the openCV library (an open source computer vision package). The position of each marker was determined by the centroid of the largest blob that matched the specific color. The same procedure was applied to track the positions of the color markers in the TV. The robot control code and the color tracker were run on a

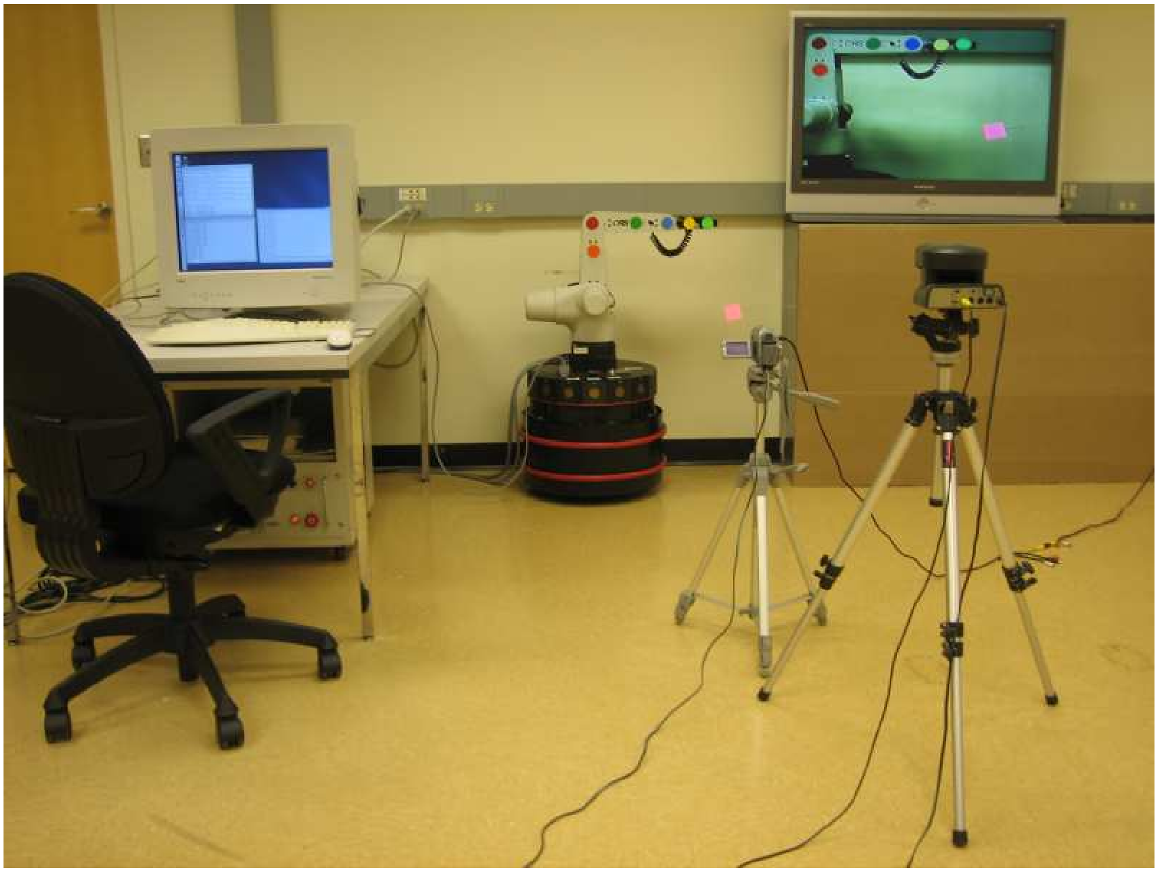


Figure 6.23: Experimental setup for the robot experiments described in this section.

Pentium IV machine (2.6 GHz, 1 GB RAM), running Linux (Fedora Core 4).

The colors of the body markers on the real robot arm and the colors of the body markers in the TV image looked slightly different when viewed through the Sony EVI-D30 camera (right camera in Figure 6.23). The color calibration and contrast settings of the TV were adjusted to make this difference as small as possible. Despite these efforts, however, the appearance of the two sets of colors was never the same. To overcome this problem while still using color tracking as the primary sensing modality two sets of color histograms were used: one for the real body markers and one for their image in the TV monitor.

In this configuration the robot can observe both its real arm as well as the image of its arm in the TV monitor (see Figure 6.24.a). The original experimental design called for an opaque panel similar to the one shown in Figure 6.21. The introduction of such a large object into the frame, however, changed the auto color calibration of the camera (which

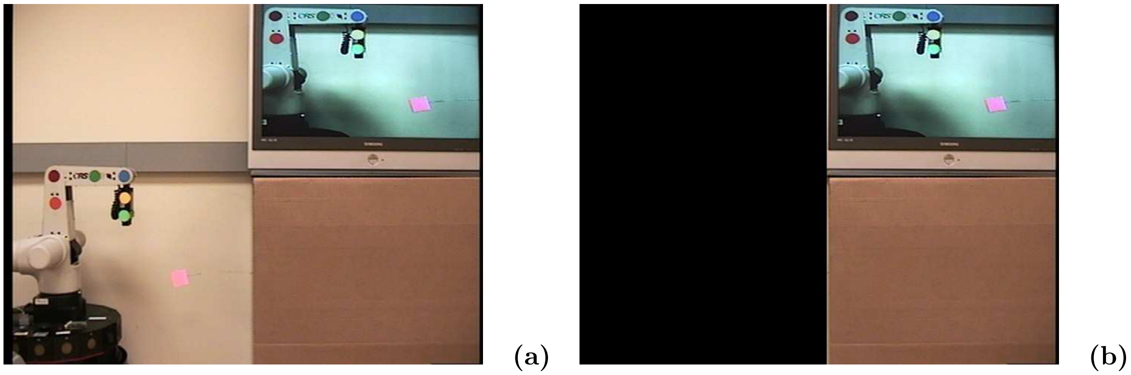


Figure 6.24: **(a)** Field of view of the robot's camera (Sony EVI-D30) in the setup shown in Figure 6.23; **(b)** What the robot sees during the testing experiments described below.

could not be turned off). Because this negatively impacted the quality of the color tracking results it was decided to use a digital version of an opaque panel instead. In other words, the left half of each frame captured by camera 1 was erased (zeroed) before it was processed (see Figure 6.24.b).

Figure 6.25 shows 500 observed positions for the blue wrist marker. These positions were recorded from real robot data while the robot was performing motor babbling. The next sub-section shows how these positions can be extended to map onto the TV image.

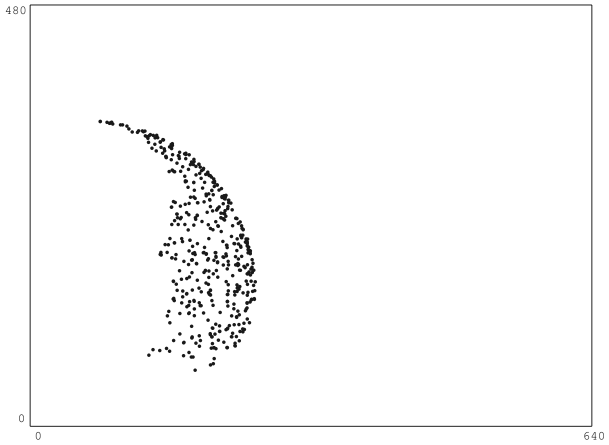


Figure 6.25: The figure shows the visual components, $\tilde{v}_i$, corresponding to the blue body marker (see Figure 6.23) in 500 body icons.

6.6.3 Calculating the Similarity Transformation

This sub-section describes the method for calculating the similarity transformation (translation, rotation, and scaling) between the position of the real robot arm and the image of the arm in the TV monitor. The parameters of this transformation are used to extend the visual components of the body icons in the RBS model. After the extension the TV image can be used for video-guided behaviors.

The method calculates the similarity transformation using two sets of points. The first set consists of ten different positions of a specific body marker (the blue wrist marker was used). These positions are estimated from proprioceptive information using Formula 6.2 as the body marker cannot be observed directly (see Figure 6.24.b). The second set consists of the ten corresponding positions of the same body marker but observed in the video image. The robot gathers the two sets while performing motor babbling. If the wrist marker cannot be detected for some joint configuration (e.g., because it is out of the TV frame or because of sensory noise which reduces the visible size of the marker below the detection threshold of 35 pixels) the robot picks a new random joint vector and continues the motor babbling.

After the two sets of data points are gathered the similarity transformation parameters are calculated using the method described by Umeyama (1991). The method calculates the translation, rotation, and scaling between one set of points and another which is a common problem in computer vision. For the sake of completeness, Umeyama's results are summarized below without their proofs.

The problem and its solution can be stated as follows. Given two sets of points $X = \{x_i\}$ and $Y = \{y_i\}$ $(i = 1, 2, \dots, n)$ in m-dimensional space (typically m=2 or 3) find a similarity transformation with parameters (R: rotation, t: translation, and c: scaling) that minimizes the mean squared error, $e^2(R, t, c)$, between the two sets of points. Where the error is given by:

$$e^2(R, t, c) = \frac{1}{n} \sum_{i=1}^{n} \|y_i - (cRx_i + t)\|^2$$

Umeyama (1991) showed that the optimum transformation parameters can be determined uniquely as follows:

$$\begin{aligned} R &= USV^T \\ t &= \mu_x - cR\mu_x \\ c &= \frac{1}{\sigma_x^2} tr(DS) \end{aligned}$$

where

$$\begin{aligned} \mu_x &= \frac{1}{n}\sum_{i=1}^{n} x_i \\ \mu_y &= \frac{1}{n}\sum_{i=1}^{n} y_i \\ \sigma_x^2 &= \frac{1}{n}\sum_{i=1}^{n} \|x_i - \mu_x\|^2 \\ \sigma_y^2 &= \frac{1}{n}\sum_{i=1}^{n} \|y_i - \mu_y\|^2 \\ \Sigma_{xy} &= \frac{1}{n}\sum_{i=1}^{n} (y_i - \mu_y)(x_i - \mu_x)^T \end{aligned}$$

In the above formulas, μ_x and μ_y are the mean vectors of X and Y; σ_x^2 and σ_y^2 are the variances around the mean vectors of X and Y; and Σ_{xy} is the covariance matrix of X and Y. The matrix $D = \text{diag}(d_i), d_1 \geq d_2 \geq \cdots \geq d_m \geq 0$ is determined from the the singular value decomposition of Σ_{xy} which is given by UVD^T. The matrix S is given by

$$S = \begin{cases} I & \text{if } \det(\Sigma_{xy}) \geq 0 \\ \text{diag}(1, 1, \cdots, 1, -1) & \text{if } \det(\Sigma_{xy}) < 0 \end{cases}$$

when rank $(\Sigma_{xy}) = m$ and by

$$S = \begin{cases} I & \text{if } \det(U)\det(V) = 1 \\ \text{diag}(1, 1, \cdots, 1, -1) & \text{if } \det(U)\det(V) = -1 \end{cases}$$

when rank $(\Sigma_{xy}) = m - 1$.

For all experiments described in this section the singular value decompositions were performed using the code given in *Numerical Recipes in C* (Press et al., 1992) (pp. 67-70).

6.6.4 Experimental Results

Three experimental conditions (shown in Figure 6.26) were used to test the applicability of the RBS extension method for video guided-behaviors. Similar test conditions were used by Iriki et al. (2001) in their experiments with Japanese monkeys. The test conditions differ by the rotation and zoom of the second camera which affects the orientation and the size of the TV image. In the first test condition the TV image is approximately equal to the image of the real robot (see Figure 6.26.a). In the second test condition camera 2 is rotated and thus the TV image is also rotated (see Figure 6.26.b). The rotation angle is 50° (of the camera) or $-50°$ (of the TV image). In the third test condition the camera is horizontal but its image is zoomed in (see Figure 6.26.c). The zoom factor is 1.6. This value was calculated from the image in Figure 6.26.c by averaging the horizontal (1.85) and vertical (1.35) zoom ratios since the Sony Handycam does not provide those to the user and the TV introduces some image scaling as well.

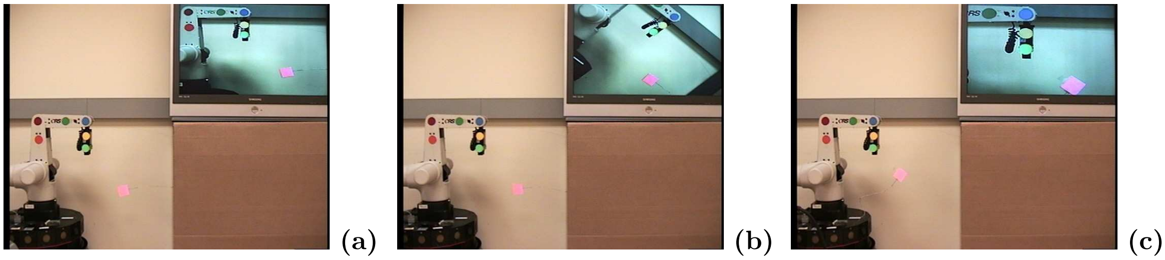


Figure 6.26: The figure shows three views from the robot's camera, one for each of the three experimental conditions. The image of the robot in the TV is: a) approximately the same size as the real robot; b) rotated by negative 50°; and c) scaled (zoomed in) by a factor of 1.6. During the actual experiments, however, the robot cannot see its own body as show in Figure 6.27.

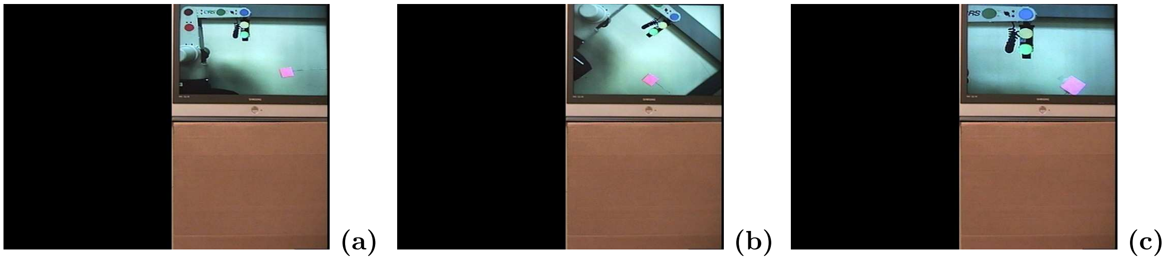


Figure 6.27: The figure shows what the robot sees during the experiments in each of the three test conditions. The left half of each frame (see Figure 6.26) was digitally erased (zeroed) before it was processed. The three images also show the incentive object (pink square) which the robot was required to grasp without observing its position directly. Instead, the robot had to use the TV image to guide its grasping behaviors.

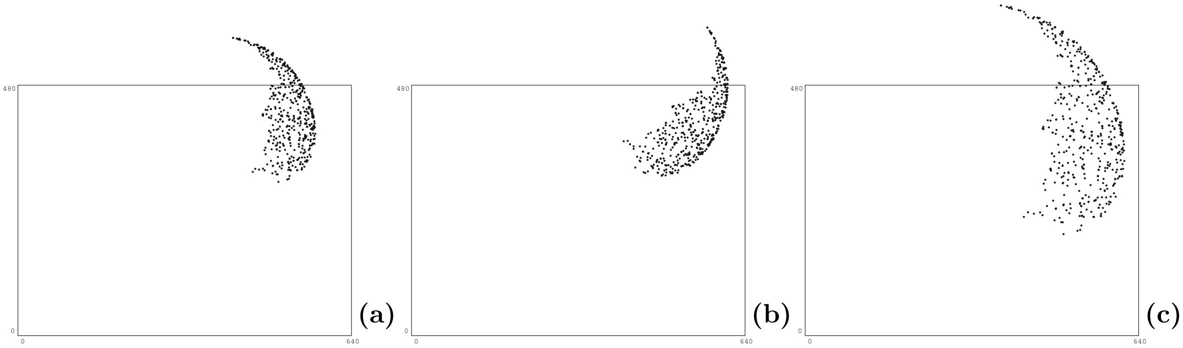


Figure 6.28: The figure shows the extended positions of the body icons (visual components for the blue wrist marker only) after the extension of the RBS in each of the three test conditions. By comparing this figure with Figure 6.25 it is obvious that the visual components of the body icons are: (a) translated; (b) rotated and translated; and (c) scaled, rotated and translated relative to their original configuration. Furthermore, the new positions coincide with the positions in which the blue marker can be observed in the TV. Because the extended positions are no longer tied to the camera coordinates some of them may fall outside the 640×480 camera image.

Tables 6.13, 6.14, and 6.15 show the estimated transformation parameters for each of the 3 test conditions in ten separate trials. Because stereo vision was not used there are only 2 meaningful translation parameters (t_x and t_y) and only one rotation parameter (θ_z). The scale factor was also estimated using Umeyama's method. As with any real robot data the results are slightly noisy. Nevertheless, the transformation parameters were estimated correctly in all trials as indicated by their small standard deviations.

Table 6.13: Transformation parameters (normal test case).

Trial #	t_x	t_y	θ_z	Scale
Trial 1	339.1	245.0	−1.20°	0.908
Trial 2	366.6	230.7	+4.87°	0.868
Trial 3	303.6	216.1	−0.57°	1.070
Trial 4	339.2	262.0	−3.04°	0.846
Trial 5	334.2	231.5	−0.74°	0.950
Trial 6	345.0	236.0	+1.55°	0.903
Trial 7	322.6	200.0	+3.44°	1.055
Trial 8	344.1	237.5	+0.92°	0.906
Trial 9	317.2	244.4	−3.38°	0.954
Trial 10	356.0	246.1	+0.46°	0.864
Mean	**336.8**	**234.9**	**+0.23°**	**0.932**
Stdev	**18.5**	**17.2**	**2.61°**	**0.077**

For each of the three condition the robot was tested with the task of grasping an incentive object (pink object in Figure 6.26 and Figure 6.27). During the experiments the object could only be seen in the TV image (see Figure 6.27). For each of the three test conditions the grasping experiment was performed five times. The robot successfully grasped the incentive object in 5 out of 5 trials for condition 1; 4 out of 5 trials for condition 2; and 0 out of 5 trials for condition 3.

One possible reason why the robot failed completely in the zoom condition is due to the poor quality of the color tracking results at this high level of magnification. This result is counter-intuitive as one would expect just the opposite to be true. However, when the image of the robot's arm is really large the auto color calibration of the Sony Handycam (which could not be turned off) is affected even by the smallest movements of the robot. Shadows and other transient light effects are also magnified. Thus, it proved difficult to

Table 6.14: Transformation parameters (rotation test case).

Trial #	t_x	t_y	θ_z	Scale
Trial 1	287.7	420.5	$-49.4°$	0.893
Trial 2	244.5	430.9	$-50.8°$	1.035
Trial 3	278.8	426.5	$-50.2°$	0.931
Trial 4	234.2	427.6	$-49.5°$	1.065
Trial 5	246.1	437.1	$-52.8°$	1.019
Trial 6	278.9	421.4	$-49.6°$	0.918
Trial 7	250.4	399.0	$-44.0°$	1.014
Trial 8	260.5	441.7	$-53.8°$	0.971
Trial 9	250.7	425.7	$-50.1°$	1.007
Trial 10	257.3	419.1	$-48.2°$	0.995
Mean	**258.9**	**425.0**	**-49.84°**	**0.985**
Stdev	**17.5**	**11.6**	**2.64°**	**0.055**

Table 6.15: Transformation parameters (zoomed in test case).

Trial #	t_x	t_y	θ_z	Scale
Trial 1	187.1	67.7	$-1.78°$	1.677
Trial 2	201.0	94.3	$-2.64°$	1.584
Trial 3	190.6	80.5	$-2.46°$	1.653
Trial 4	188.9	62.7	$-0.92°$	1.697
Trial 5	236.1	55.2	$+3.33°$	1.594
Trial 6	217.2	63.8	$+2.12°$	1.621
Trial 7	198.9	95.2	$-3.38°$	1.588
Trial 8	206.2	126.6	$-5.57°$	1.498
Trial 9	222.7	105.8	$-2.01°$	1.512
Trial 10	219.6	53.4	$+1.72°$	1.635
Mean	**206.8**	**80.5**	**-1.16°**	**1.606**
Stdev	**16.5**	**24.3**	**2.76°**	**0.065**

track the body markers in the zoomed in test condition using the current tracking method. The transformation parameters for this test case were estimated correctly (see Table 6.15) because the ten data points needed for the calculation are collected only when the wrist marker can be observed in the TV(which was possible for some body configurations but not others).

The failure in the rotation test case was also due to poor color tracking results.

6.7 Chapter Summary

This chapter introduced the notion of *extended robot body schema* and described a computational model that implements this notion.

The RBS model is learned from self-observation data (visual and proprioceptive) gathered during a motor babbling phase. The RBS provides the robot with a sensorimotor model of its own body that can also be used to control robot movements. Through the process of extension the RBS can accommodate changes in the configuration of the robot triggered by attached objects. The extension is triggered by the near perfect correlation between self-movements and movements of external object.

This chapter also described how the extension of the robot body schema can be triggered by the temporal contingency between the actions of the robot and observed self-movements in the video image. It reported an experiment in which the robot was able to detect its self-image in a TV monitor and use that real-time video image to guide its own actions in order to reach an object observable only through the TV image. The extension of the RBS allows the robot to use the TV image to guide its arm movements as if it were observing it directly. This constitutes the first-ever reported instance of a video-guided behavior by a robot.

Future work can build upon the principles described in this chapter and extend the domains in which robots can use video-guided behaviors. For example, using a mouse to position a cursor on a computer monitor or using a rear view camera (or mirror) to back up a car are just two possible applications.

The results described in this chapter demonstrate that a robot can learn a sensorimotor model of its own body from self-observation data, in answer to the second research question stated in Section 1.2. The results also show that this pliable body model can be used to facilitate goal-oriented tasks such as video-guided behaviors.

CHAPTER VII

LEARNING THE AFFORDANCES OF TOOLS

7.1 Introduction

A simple object like a stick can be used in numerous tasks that are quite different from one another. For example, a stick can be used to strike, poke, prop, scratch, pry, dig, etc. It is still a mystery how animals and humans learn these affordances (Gibson, 1979) and what are the cognitive structures used to represent them.

This chapter introduces a novel approach to representing and learning tool affordances by a robot. The tool representation described here uses a behavior-based approach to *ground* the tool affordances in the behavioral repertoire of the robot. The representation is learned during a behavioral babbling stage in which the robot randomly chooses different exploratory behaviors, applies them to the tool, and observes their effects on environmental objects. The chapter shows how the *autonomously* learned affordance representation can be used to solve tool-using tasks by dynamically sequencing the exploratory behaviors based on their expected outcomes. The quality of the learned representation is tested on extension-of-reach tool-using tasks.

7.2 Affordances and Exploratory Behaviors

James Gibson defined affordances as “perceptual invariants” that are directly perceived by an organism and enable it to perform tasks (Gibson, 1979) (also see Section 2.3.2). Gibson is not specific about the way in which affordances are learned but he suggests that some affordances are learned in infancy when the child experiments with external objects.

The related work on animal object exploration presented in Section 2.1.1 indicates that animals use stereotyped exploratory behaviors when faced with a new object (Power, 2000; Lorenz, 1996). This set of behaviors is species specific and may be genetically predetermined. In fact, Glickman and Sroges (1966) observed that the exploratory behaviors frequently overlap with the consumatory behaviors for the species. For some species of animals these tests include almost their entire behavioral repertoire: “A young corvide bird, confronted with an object it has never seen, runs through practically all of its behavioral patterns, except social and sexual ones.” (Lorenz, 1996, p. 44).

Recent studies with human subjects also suggest that the internal representation for a new tool used by the brain might be encoded in terms of specific past experiences (Mah and Mussa-Ivaldi, 2003). Furthermore, these past experiences consist of brief feedforward movement segments used in the initial exploration of the tool (Mah and Mussa-Ivaldi, 2003). A tool task is later learned by dynamically combining these sequences.

Thus, the properties of a tool that an animal is likely to learn are directly related to the behavioral and perceptual repertoire of the animal. Furthermore, the learning of these properties should be relatively easy since the only requirement is to perform a (small) set of exploratory behaviors and observe their effects. Based on the results of these “experiments” the animal builds an internal representation for the tool and the actions that it affords. Solving tool tasks in the future is based on dynamically combining the exploratory behaviors based on their expected results.

The next section formulates a behavior-grounded computational model of tool affordances based on these principles.

7.3 Behavior-Grounded Tool Representation

7.3.1 Robots, Tools, and Tasks

The notion of robotic tool use brings to mind four things: 1) a robot; 2) an environmental object which is labeled a tool; 3) another environmental object to which the tool is applied (labeled an attractor); and 4) a tool task. For tool use to occur all four components need to be present. In fact, it is meaningless to talk about one without taking into account the other three. What might be a tool for one robot may not be a tool for another because of differences in the robots’ capabilities. Alternatively, a tool might be suitable for one task (and/or object) but completely useless for another. And finally, some tasks may not be within the range of capabilities of a robot even if the robot is otherwise capable of using tools. Thus, the four components of tool use must always be taken into consideration together.

This is compatible with Gibson's claim that objects afford different things to people with different body sizes. For example, an object might be graspable for an adult but may not be graspable for a child. Therefore, Gibson suggests that a child learns "his scale of sizes as commensurate with his body, not with a measuring stick" (Gibson, 1979, p. 235). Section 3.4 gives some additional examples and relates them to the subjectivity principle.

Because of these arguments, any tool representation should take into account the robot that is using the tool. In other words, the representation should be *grounded* in the behavioral and perceptual repertoire of the robot. The main advantage of this approach is that the tool's affordances are expressed in concrete terms (i.e., behaviors) that are available to the robot's controller. Note that this is in sharp contrast with other theories of intelligent systems reasoning about objects in the physical world (Hayes (1985); Stark and Bowyer (1996)). They make the assumption that object properties can be expressed in abstract form (by a human) without taking into account the specific robot that will be using them.

Another advantage of the *behavior-grounded* approach is that it can handle changes in the tool's properties over time. For example, if a familiar tool becomes deformed (or a piece of it breaks off) it is no longer the same tool. However, the robot can directly test the accuracy of its representation by executing the same set of exploratory behaviors that was used in the past. If any inconsistencies are detected in the resulting observations they can be used to update the tool's representation. Thus, the accuracy of the representation can be directly tested by the robot as mandated by the verification principle (see Section 3.2).

7.3.2 Theoretical Formulation

The previous section presented a justification for using exploratory behaviors to learn the affordances of tools. This section describes a theoretical formulation of these ideas. The tool representation introduced here uses a behavior-based approach (Arkin, 1998) to *ground* the tool affordances in the existing behavioral repertoire of the robot. The behavior-grounded approach is formulated using the following notation.

Let $\beta_{e_1}, \beta_{e_2}, \ldots, \beta_{e_k}$ be the set of exploratory behaviors available to the robot. Each behavior, has one or more parameters that modify its outcome. Let the parameters for behavior β_{e_i} be given as a parameter vector $E_i = [e^i_1, e^i_2, \ldots e^i_{p(i)}]$, where $p(i)$ is the number of parameters for this behavior. The behaviors, and their parameters, could be learned by imitation, programmed manually, or learned autonomously by the robot. For the purposes of this chapter, however, the issue of how these behaviors are selected and/or learned will be ignored.

In a similar fashion, let $\beta_{b_1}, \beta_{b_2}, \ldots, \beta_{b_m}$ be the set of binding behaviors available to the robot. These behaviors allow the robot to attach tools to its body. The most common binding behavior is grasping. However, there are many examples in which a tool can be controlled even if it is not grasped (e.g., by holding it between one's teeth). Therefore, the term *binding* will be used. The parameters for binding behavior β_{b_i} are given as a parameter vector $B_i = [b^i_1, b^i_2, \ldots b^i_{q(i)}]$.

Furthermore, let the robot's perceptual routines provide a stream of observations in the form of an observation vector $O = [o_1, o_2, \ldots, o_n]$. It is assumed that the set of observations is rich enough to capture the essential features of the tasks to which the tool will be applied.

A change detection function, $\mathcal{T}(O(t'), O(t'')) \rightarrow \{0, 1\}$, that takes two observation vectors as parameters is also defined. This function determines if an "interesting" observation was detected in the time interval $[t', t'']$. In the current set of experiments $\mathcal{T} = 1$ if the attractor was moving during the execution of the last exploratory behavior. The function $\mathcal{T}$ is defined as binary because movement is either detected or it is not. Furthermore, movement detection falls in the general category of events that can grab one's attention. O'Regan and Noë (2001) refer collectively to events like these as the *grabbiness* property of the environment.

With this notation in mind, the functionality of a tool can be represented with an *Affordance Table* of the form:

Binding Behavior	**Binding Params**	**Exploratory Behavior**	**Exploratory Params**	O^{start}	O^{end}	**Times Used**	**Times Succ**

In each row of the table, the first two entries represent the binding behavior that was used. The second two entries represent the exploratory behavior and its parameters. The next two entries store the observation vector at the start and at the end of the exploratory behavior. The last two entries are integer counters used to estimate the probability of success of this sequence of behaviors.

The meanings of these entries are best explained with an example. Consider the following sample row

Binding Behavior	Binding Params	Exploratory Behavior	Exploratory Params	O^{start}	O^{end}	Times Used	Times Succ
β_{b_1}	$\tilde{b}_1^1$	β_{e_3}	$\tilde{e}_1^3, \tilde{e}_2^3$	$\tilde{O}(t')$	$\tilde{O}(t'')$	4	3

in which the binding behavior β_{b_1} which has one parameter, was performed to grasp the tool. The specific value of the parameter for this behavior was $\tilde{b}_1^1$ (a $\tilde{}$ sign is used to represent a specific fixed value). Next, the exploratory behavior β_{e_3} was performed with specific values $\tilde{e}_1^3$ and $\tilde{e}_2^3$ for its two parameters. The value of the observation vector prior to the start of β_{e_3} was $\tilde{O}(t')$ and its value after β_{e_3} has completed was $\tilde{O}(t'')$. This sequence of behaviors was performed four times. In three of these trials, the observed movement direction of the attractor was similar to the stored movement direction in the table row. Two attractor movements are considered to be similar if their angular movement directions are within 40° of each other (also see Section 7.5.2). Therefore, the replication probability of this affordance is 75%. Section 7.6 and Figure 7.8 provide more information about the organization of the affordance table.

Initially the affordance table is blank. When the robot is presented with a tool it performs a *behavioral babbling* routine, which picks binding and exploratory behaviors at random, applies them to the tools and objects, observes their effects, and updates the table. New rows are added to the table only if $\mathcal{T}$ was one while the exploratory behavior was performed. The integer counters are not updated during learning trials. Prior to testing trials the two counters of all rows are initialized to 1 and are later updated based on actual experience.

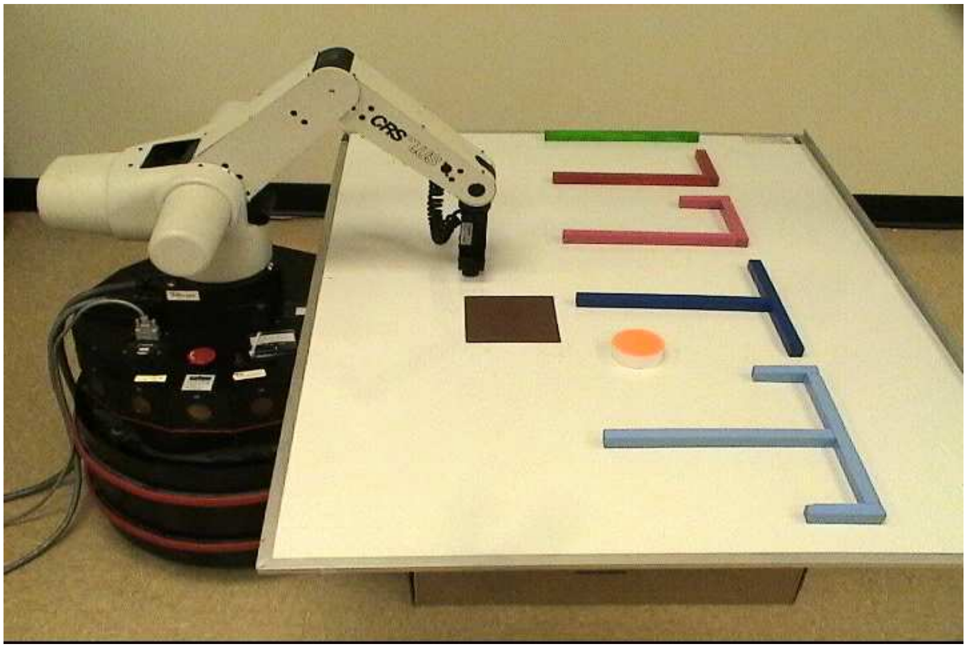


Figure 7.1: The robot and the five tools used in the experiments.

7.4 Experimental Setup

All experiments were performed using the mobile manipulator described in Section 4.2. Five tools were used in the experiments: stick, L-stick, L-hook, T-stick, and T-hook (Figure 7.1). The tools were built from pine wood and painted with spray paint. The choice of tools was motivated by the similar tools that Köhler's used in his experiments with chimpanzees(Köhler, 1931). An orange hockey puck was used as an attractor object.

A Sony EVI-D30 camera was mounted on a tripod overlooking the robot's working area (Figure 7.2). The robot's wrist, the tools, and the attractor were color coded so that their positions can be uniquely identified and tracked using computer vision (see Figures 7.3 and 7.4). The computer vision code was run at 15Hz in 640x480 resolution mode.

To ensure consistent tracking results between multiple robot experiments the camera was calibrated every time it was powered up. The calibration was performed using Roger Tsai's method (Tsai, 1986, 1987) and the code given in (Willson, 1995). A 6 × 6 calibration pattern was used (Figure 7.5). The pattern consists of small color markers placed on a cardboard, 5 inches apart, so that they form a square pattern. The pixel coordinates of the 36 uniformly colored markers were identified automatically using color segmentation

(Figure 7.6). The pixel coordinates of the markers and their world coordinates (measured in a coordinate system attached to the table) are passed to the camera calibration procedure which calculates the intrinsic and extrinsic parameters of the camera. These parameters can then be used in a mapping function which assigns to each (x,y) in camera coordinates a (X,Y,Z) location in world coordinates, where Z is a parameter supplied by the user (e.g., Z=0 is the height of the table).

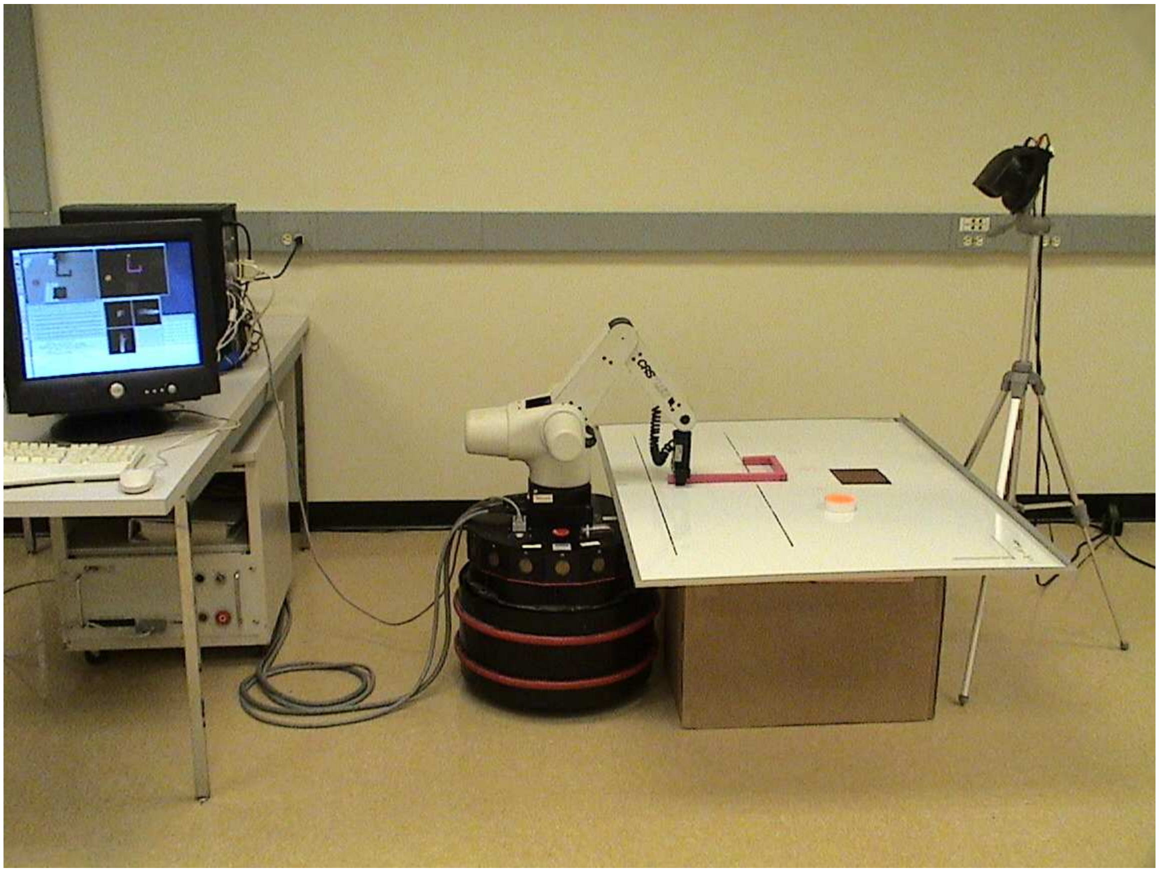


Figure 7.2: Experimental setup.

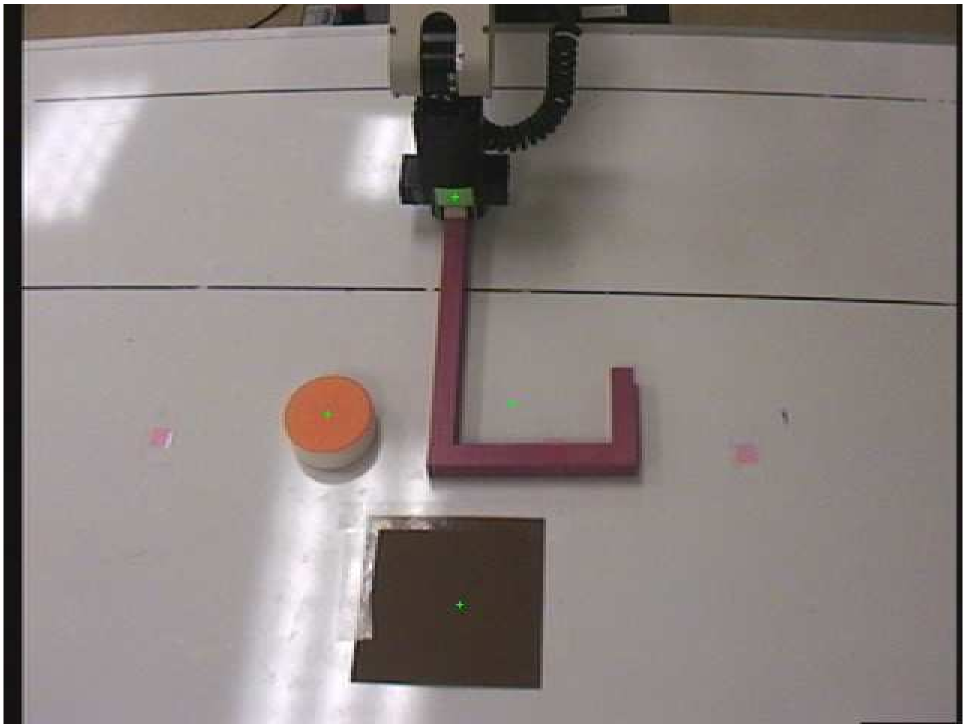


Figure 7.3: Color tracking: raw camera image.

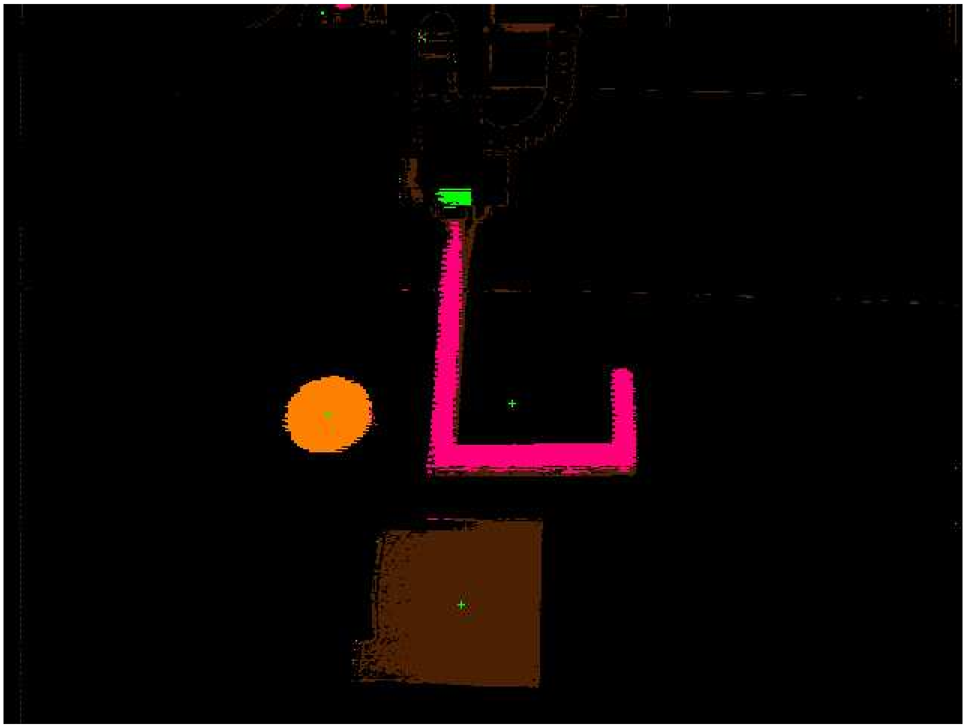


Figure 7.4: Color tracking: segmentation results.

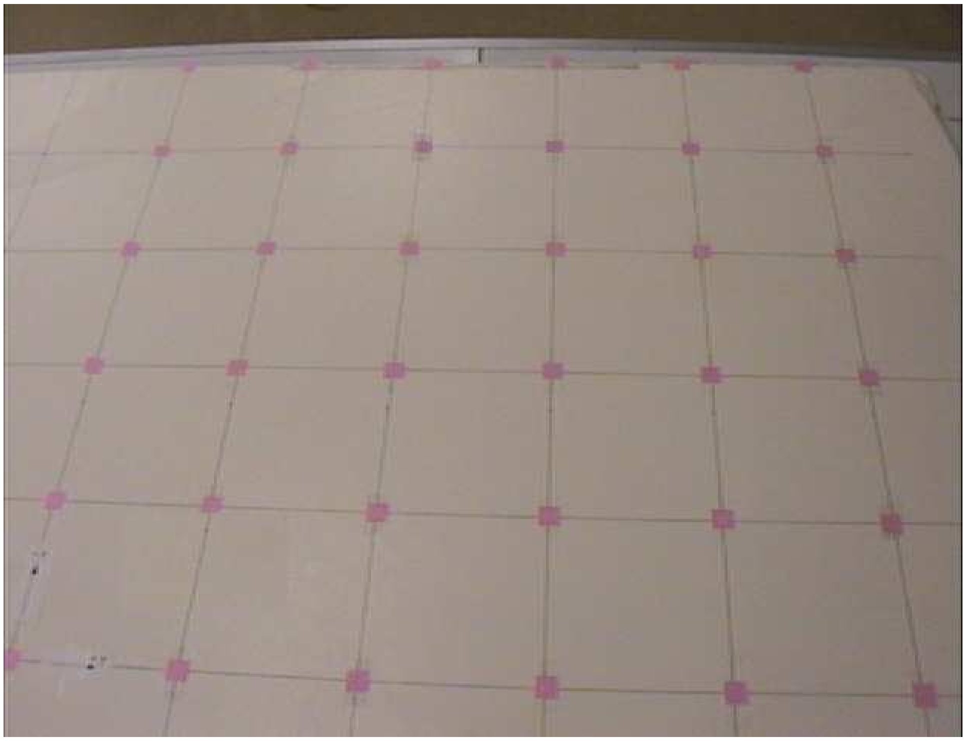


Figure 7.5: The 6×6 pattern used for camera calibration.

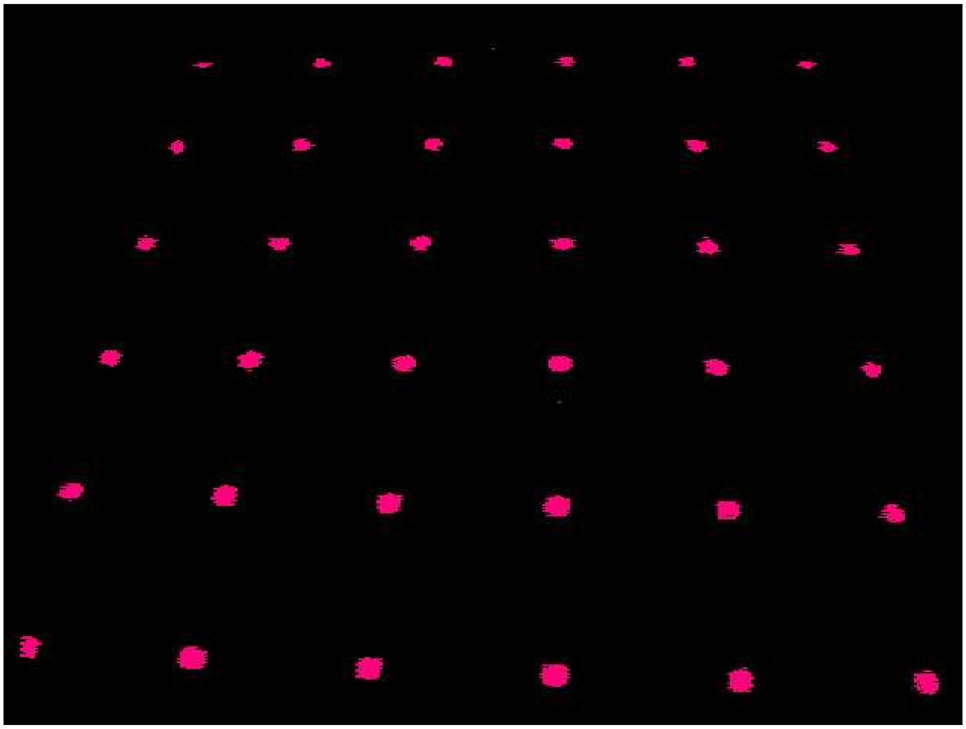


Figure 7.6: Results of color segmentation applied to the calibration pattern.

7.5 Exploratory Behaviors

All behaviors used here were encoded manually from a library of *motor schemas* and *perceptual schemas* (Arkin, 1998) developed for this specific robot. The behaviors result in different arm movement patterns as described below.

Exploratory Behaviors	**Parameters**
Extend arm	offset_distance
Contract arm	offset_distance
Slide arm left	offset_distance
Slide arm right	offset_distance
Position wrist	x,y

The first four behaviors move the arm in the indicated direction while keeping the wrist perpendicular to the table on which the tool slides. These behaviors have a single parameter which determines how far the arm will travel relative to its current position. Two different values for this parameter were used (2 and 5 inches). The *position wrist* behavior moves the manipulator such that the centroid of the attractor is at offset (x, y) relative to the wrist.

7.5.1 Grasping Behavior

There are multiple ways in which a tool can be grasped. These represent a set of affordances which we will call *binding affordances*, i.e., the different ways in which the robot can attach the tool to its body. These affordances are different from the *output affordances* of the tool, i.e., the different ways in which the tool can act on other objects. This chapter focuses only on output affordances, so the binding affordances were specified with only one grasping behavior. The behavior takes as a parameter the location of a single grasp point located at the lower part of the tool's handle.

7.5.2 Observation Vector

The observation vector has 12 real-value components. In groups of three, they represent the position of the attractor object in camera-centric coordinates, the position of the object relative to the wrist of the robot, the color of the object, and the color of the tool.

Observation	**Meaning**
o_1, o_2, o_3	X,Y,Z positions of the object (camera-centric)
o_4, o_5, o_6	X,Y,Z positions of the object (wrist-centric)
o_7, o_8, o_9	R,G,B color components of the object
o_{10}, o_{11}, o_{12}	R,G,B color components of the tool

The change detection function $\mathcal{T}$ was defined with the first three components, o_1, o_2, o_3. To determine if the attractor is moving, $\mathcal{T}$ calculates the Euclidean distance and thresholds it with an empirically determined value (0.5 inches). The *times-successful* counter is incremented if the observed attractor movement is within 40 degrees of the expected movement stored in the affordance table.

7.6 Learning Trials

The third research question stated in Section 1.2 asked whether a robot can use exploratory behaviors to both learn and represent the functional properties or affordances of tools. This section describes one procedure which meets these criteria and was successfully used by the CRS+ robot to learn the affordances of the five tools shown in Figure 7.1. A flowchart diagram for this procedure (which was used during the learning trails) is shown in Figure 7.7. The procedure used during the testing trials is described in the next section.

During the learning trials the robot was allowed to freely explore the properties of the five stick tools shown in Figure 7.1. The exploration of each tool consists of trying different exploratory behaviors, observing their results, and filling up the affordance table for the tool. A new entry is added to the affordance table only if the attractor object is observed to move while the exploratory behavior is being performed. If the object is not affected by the exploratory behavior then the affordance table remains unchanged. Thus, object movement acts both as an attention grabbing mechanism which triggers an update of the affordance representation and also as an indicator of the effect of the tool on the object (see Section 7.3.2). The initial positions of the attractor object and the tool were random. If the attractor was pushed out of tool reach by the robot then the learning trial was temporarily suspended while the attractor was manually placed in a new random position. The learning trials were limited to one hour of run time for every tool.

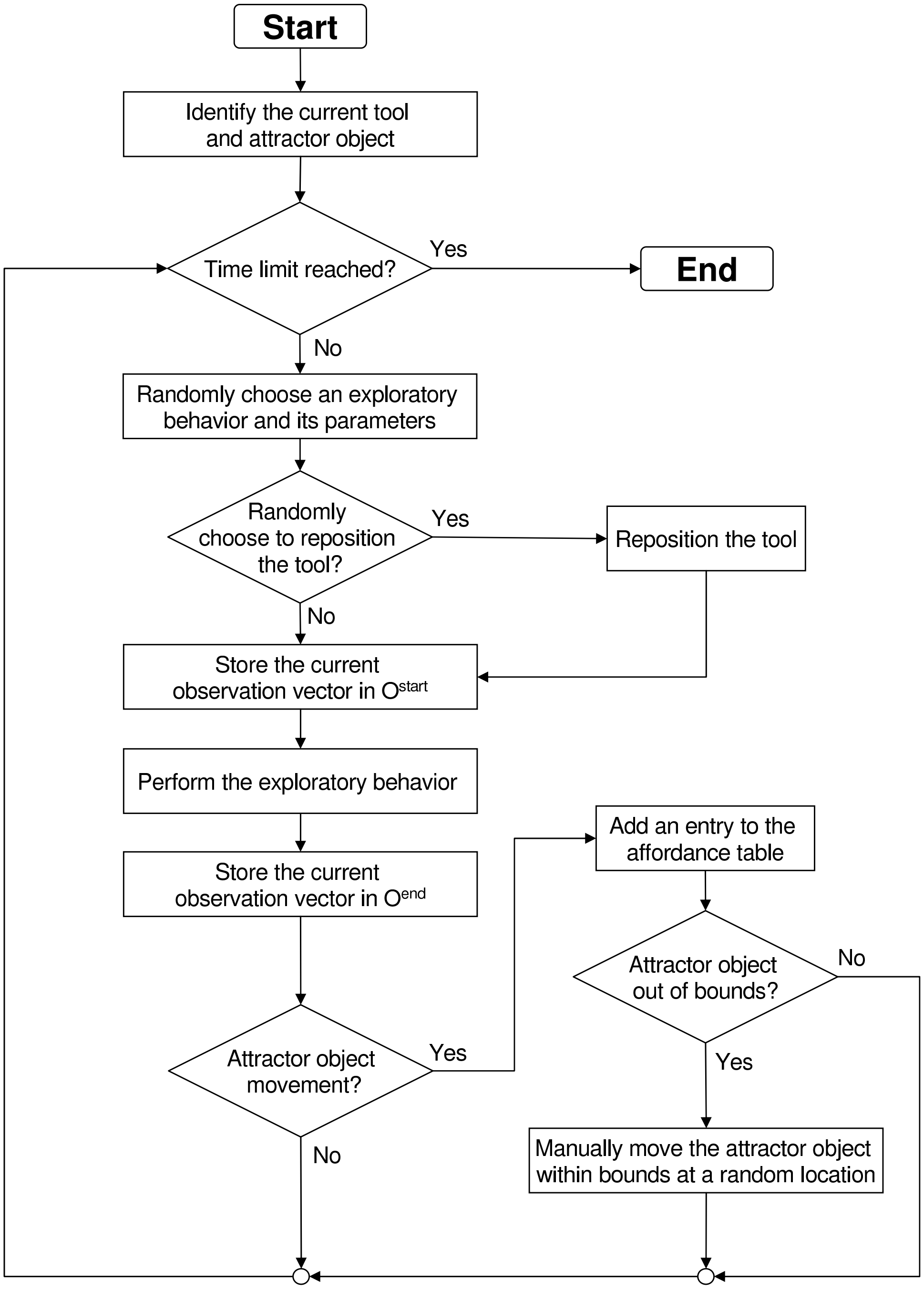

Start
Identify the current tool and attractor object
Time limit reached?
Yes
End
No
Randomly choose an exploratory behavior and its parameters
Randomly choose to reposition the tool?
Yes
Reposition the tool
No
Store the current observation vector in O^start
Perform the exploratory behavior
Store the current observation vector in O^end
Add an entry to the affordance table
Attractor object out of bounds?
No
Attractor object movement?
Yes
Yes
Manually move the attractor object within bounds at a random location
No

Figure 7.7: Flowchart diagram for the exploration procedure used by the robot to learn the affordances of a specific tool when the tool is applied to an attractor object.

7.6.1 What Is Learned

Figure 7.8 illustrates what the robot can learn about the properties of the T-hook tool based on a single exploratory behavior. In this example, the exploratory behavior is “Contract Arm” and its parameter is “5 inches.” The two observation vectors are stylized for the purposes of this example. The information that the robot retains is not the images of the tool and the puck but only the coordinates of their positions as explained above. If a different exploratory behavior was selected by the robot it is possible that no movement of the puck will be detected. In this case the robot will not store any information (row) in the affordance table.

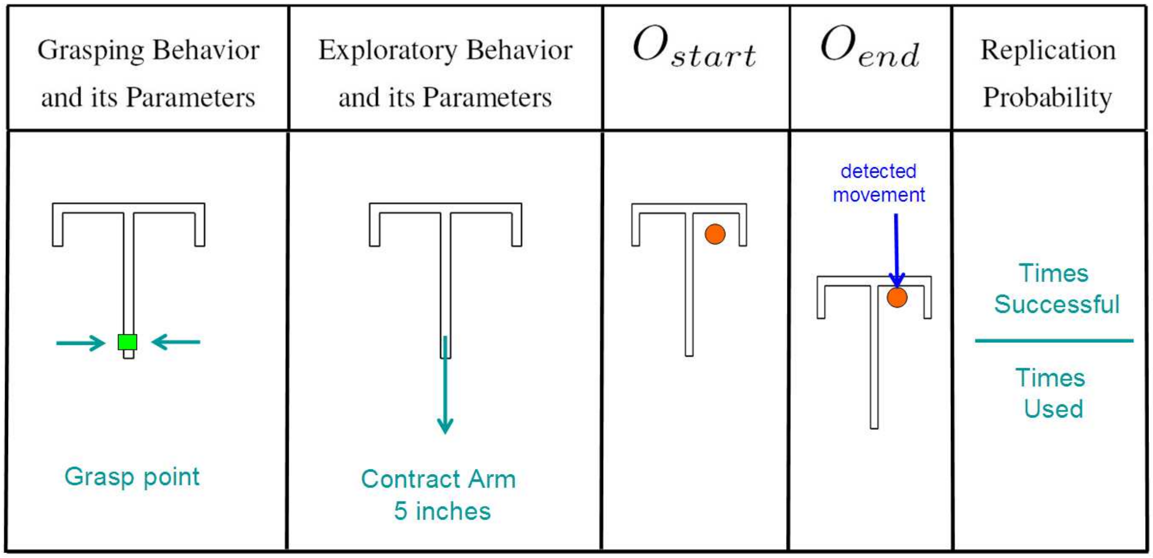


Figure 7.8: Contents of a sample row of the affordance table for the T-hook tool.

When the robot performs multiple exploratory behaviors a more compact way to represent this information is required. A good way to visualize what the robot learns is with graphs like the ones shown in Figure 7.9. The figures show the observed outcomes of the exploratory behaviors when the T-hook tool was applied to the hockey puck while the robot was performing behavioral babbling. Each of the eight graphs shows the observed movements of the attractor object when a specific exploratory behavior was performed. The movements of the attractor object are shown as arrows. The start of each arrow corresponds to the initial position of the attractor relative to the wrist of the robot (and thus relative to the grasp point) just prior to the start of the exploratory behavior. The arrow represents the observed distance and direction of movement of the attractor in camera coordinates at the end of the exploratory behavior. In other words, each of the arrows shown in Figure 7.9 represents one observed movement of the puck similar to the "detected movement" arrow show in Figure 7.8. The arrows in Figure 7.9 are superimposed on the initial configuration of the tool and not on its final configuration as in Figure 7.8.

This affordance representation can also be interpreted as a *predictive* model of the results of the exploratory behaviors. In other words, the affordances are represented as the expected outcomes of specific behaviors. This interpretation of affordances is consistent with the idea that biological brains are organized as predictive machines that anticipate the consequences of actions – their own and those of others (Berthoz, 2000, p. 1). It is also consistent with some recent findings about the internal representation of the functional properties of novel objects and tools in humans. As Mah and Mussa-Ivaldi (2003) note, "if the brain can predict the effect of pushing or pulling an object this is effectively an internal model of the object that can be used during manipulation". A recent result in the theoretical AI literature also shows that the state of a dynamic system can be represented by the outcomes of a set of tests (Singh et al., 2002; Littman et al., 2002). The tests consist of action-observation sequences. It was shown that the state of the system is fully specified if the outcomes of a basis set of test called *core tests* are known in advance (Littman et al., 2002).

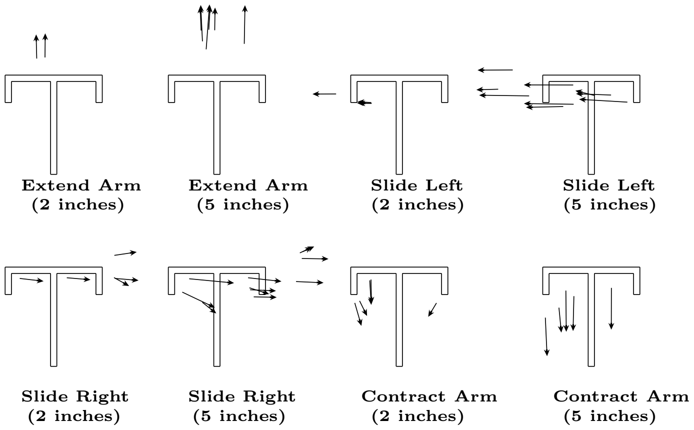


Figure 7.9: Visualizing the affordance table for the T-hook tool. Each of the eight graphs show the observed movements of the attractor object after a specific exploratory behavior was performed multiple times. The start of each arrow corresponds to the position of the attractor in wrist-centered coordinates (i.e., relative to the tool's grasp point) just prior to the start of the exploratory behavior. The arrow represents the total distance and direction of movement of the attractor in camera coordinates at the end of the exploratory behavior.

7.6.2 Querying the Affordance Table

After the affordance table is populated with values it can be queried to dynamically create behavioral sequences that solve a specific tool task. The behaviors in these sequences are the same behaviors that were used to fill the table. Section 7.7 and Figure 7.10 describe the test procedure employed by the robot during tool-using tasks. This subsection describes only the search heuristic used to select the best affordance for the current task configuration which is required by the procedure shown in Figure 7.10.

During testing trials, the best affordance for a specific step in a tool task was selected using a greedy heuristic search. The query method that was adopted uses empirically derived heuristics to perform multiple nested linear searches through the affordance table as described below. Each successive search is performed only on the rows that were not eliminated by the previous searches. Thus, at each level the search for the best tool affordance that is applicable to the current situation is focused only on the affordances that have already met the previous search criteria. Four nested searches were performed in the order shown below:

1) Select all rows that have observation vectors consistent with the colors of the current tool and object.

2) From the remaining rows select those with probability of success greater than 50%, i.e., select only those rows that have a replication probability (times_successful/times_used) greater than $\frac{1}{2}$ (the reasons for choosing this threshold value are described below).

3) Sort the remaining rows (in increasing order) based on the expected distance between the attractor object and the goal region if the behavior associated with this row were to be performed.

4) From the top 20% of the sorted rows choose one row which minimizes the re-positioning of the tool relative to its current location (the repositioning rule is described in Section 7.7).

As it was mentioned above the greedy one-step-lookahead heuristic was derived empirically. The performance of the heuristic was fine tuned for speed of adaptation in the presence of uncertainty which is important when multiple robot trials have to be performed. For example, the threshold value of 50% used in step 2 above was chosen in order to speed up the elimination of outdated affordances when the geometry of the tool suddenly changes (see the experiment described in Section 7.7.2). With this threshold value it takes only one unsuccessful behavioral execution in order to eliminate an affordance from further consideration. Future work should attempt to formulate a more principled approach to this affordance-space planning problem, preferably using performance data derived from tool-using experiments with animals and humans (e.g., Mah and Mussa-Ivaldi (2003)).

7.7 Testing Trials

This section shows how the affordance representation described in the previous section can be used to solve tool-using tasks. In other words, this section answers affirmatively the second part of research question #3 stated in Section 1.2.

Figure 7.10 shows the flowchart diagram for the procedure used by the robot during the testing trials. The experiments described in the following subsections require the robot to move the attractor object over a color-coded goal region. Thus, the testing procedure starts with identifying the position of the goal region. The current tool and attractor object are also identified by their unique colors. Each testing trial ends when the attractor object is placed over the goal region or after a timeout interval has expired.

The procedure shown in Figure 7.10 uses the affordance representation for the currently available tool which is represented with the help of an affordance table (e.g., see Figure 7.9). At each step the robot selects the best tool affordance applicable in the current situation using the greedy heuristic described in Section 7.6.2. Based on the observation vector O^{start} associated with the selected affordance the robot decides whether it needs to reposition the tool or not. The tool is repositioned if the current position of the attractor object relative to the tool (i.e., relative to the green wrist marker shown in Figure 7.3) is more than 4 inches away from the attractor position stored in O^{start} for the best affordance.

Next the robot performs the exploratory behavior associated with the best affordance and compares its outcome with the outcome stored in the affordance table. If the observed object movement matches the movement stored in the affordance table then both counters, *times_used* and *times_successful*, for this affordance are incremented (see Figure 7.8). If the behavior has no effect on the attractor object (i.e., the object is not observed to move) then the replication probability for this affordance is reduced (i.e., only the *times_used* counter is incremented which effectively reduces the value of the fraction *times_successful/times_used*). Similarly, if the effect of the behavior on the object is different from the expected effect based on previous experience (i.e., if the direction of movement of the attractor is not within 40 degrees of the expected movement direction – see Section 7.5.2) then the replication probability of the affordance is also reduced.

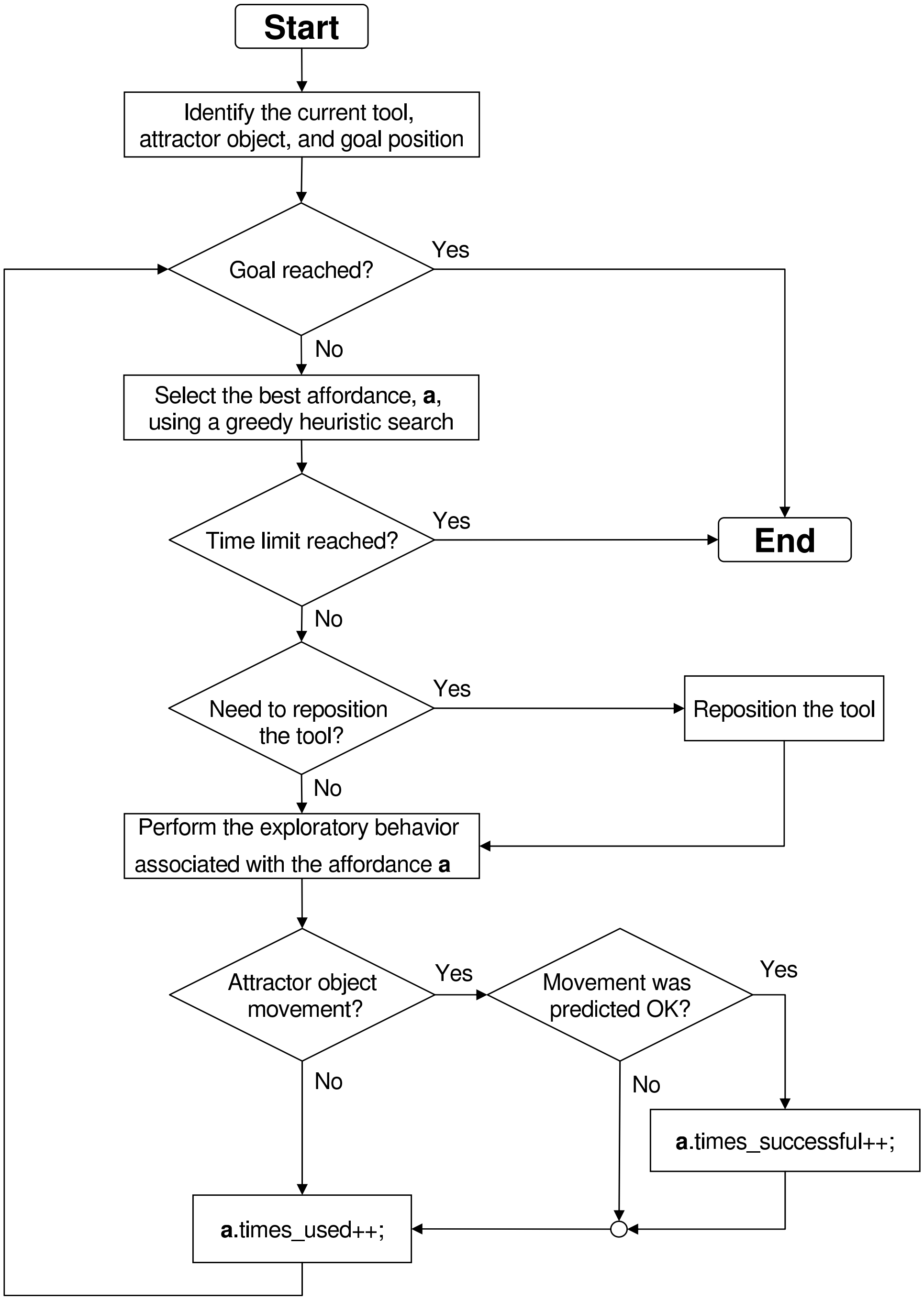

Start
Identify the current tool, attractor object, and goal position
Goal reached?
Yes
No
Select the best affordance, a, using a greedy heuristic search
Time limit reached?
Yes
End
No
Need to reposition the tool?
Yes
Reposition the tool
No
Perform the exploratory behavior associated with the affordance a
Attractor object movement?
Yes
Movement was predicted OK?
Yes
No
No
a.times_successful++;
a.times_used++;

Figure 7.10: Flowchart diagram for the procedure used by the robot to solve tool-using tasks with the help of the behavior-grounded affordance representation.

Two following two subsections describe two types of experiments that were performed. They measured the quality of the learned representation and its adaptation abilities when the tool is deformed, respectively.

7.7.1 Extension of Reach

In this experiment the robot was required to pull the attractor over a color coded goal region. Four different goal positions were defined. The first goal is shown in Figure 7.1 (the dark square in front of the robot). The second goal was located farther away from the robot (see Figure 7.2). To achieve it the robot had to push the attractor away from its body. Goals 3 and 4 were placed along the mid-line of the table as shown in Figure 7.11.

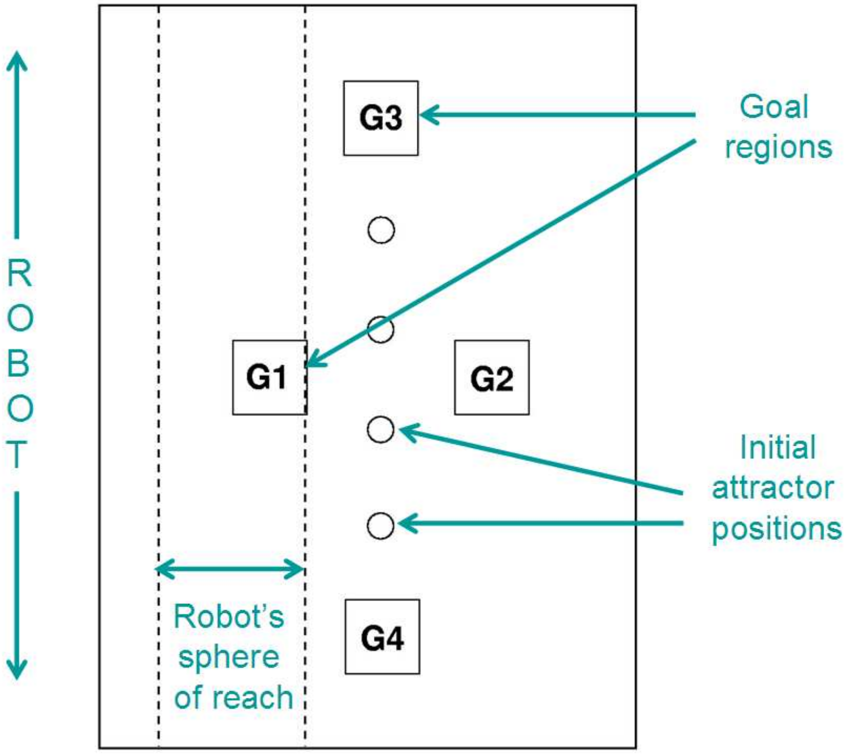


Figure 7.11: The figure shows the positions of the four goal regions (G1, G2, G3, and G4) and the four initial attractor positions used in the extension of reach experiments. The two dashed lines indicate the boundaries of the robot's sphere of reach when it is not holding any tool.

In addition to that there were 4 initial attractor positions per goal. The initial positions are located along the mid-line of the table, 6 inches apart (Figure 7.11). The tool was always placed in the center of the table. A total of 80 trials were performed (4 goals × 4 attractor positions × 5 tools). The table below summarizes the results per goal, per tool. Each table entry represents the number of successful trials with a given tool and goal configuration. A trial is considered a success if the puck is placed over the goal region . Four is the possible maximum value as there are four initial position for the puck for each goal configuration.

Tool	**Goal 1**	**Goal 2**	**Goal 3**	**Goal 4**
Stick	0	2	4	4
L-stick	4	2	4	4
L-hook	4	3	4	4
T-stick	3	3	4	4
T-hook	4	4	4	4

As can be seen from the table, the robot was able to solve this task in the majority of the test cases. The most common failure condition was due to pushing the attractor out of tool's reach. This failure was caused by the greedy one-step-lookahead heuristic used for selecting the next tool movement. If the robot plans the possible movements of the puck for 2 or 3 moves ahead these failures will be eliminated. A notable exception is the Stick tool, which could not be used to pull the object back to the near goal. The robot lacks the required exploratory behavior (*turn-the-wrist-at-an-angle-and-then-pull*) that is required to detect this affordance of the stick. Adding the capability of learning new exploratory behaviors can resolve this problem.

The results of this experiment show that the behavior-grounded tool representation can indeed be used to solve tool-using tasks (second part of research question #3).

7.7.2 Adaptation After a Tool Breaks

The second experiment was designed to test the flexibility of the representation in the presence of uncertainties. The uncertainly in this case was a tool that can break. For example, Figure 7.12 shows the tool transformation which occurs when a T-hook tool loses one of its hooks. The result is a L-hook tool. This section describes the results of an experiment in which the robot was exposed to such tool transformation after it had already learned the affordances of the T-hook tool.

To simulate a broken tool, the robot was presented with a tool that has the same color

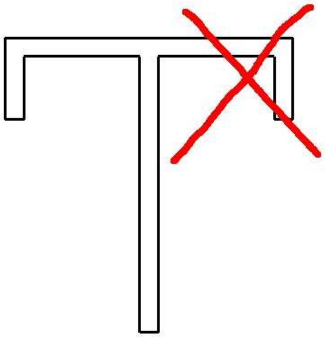


Figure 7.12: A T-hook missing its right hook is equivalent to an L-hook.

ID as another tool with a different shape. More specifically, the learning was performed with a T-hook which was then replaced with an L-hook. Because color is the only feature used to recognize tools the robot believes that it is still using the old tool.

The two tools differ in their upper right sections. Whenever the robot tried to use affordances associated with the missing parts of the tool they did not produce the expected attractor movements. Figure 7.13 shows frames from a sequence in which the robot tried in vain to use the upper right part of the tool to move the attractor towards the goal. After several trials the replication probability of the affordances associated with that part of the tool was reduced and they were excluded from further consideration. Figure 7.14 shows the rest of this sequence in which the robot was able to complete the task with the intact left hook of the tool.

A total of 16 trials similar to the one shown in Figure 7.13 were performed (i.e., 4 goal regions × 4 initial attractor positions). In each of these experiments the robot started the testing trial with the original representation for the T-hook tool and modified it based on actual experience. The robot was successful in all 16 experiments, i.e., the robot was able to place the attractor over the target goal region with the "broken" tool in all 16 experiments.

The results of this experiment show that the behavior-grounded tool representation can be autonomously, tested, verified, and corrected by the robot as mandated by the verification principle. This is an important contribution of this dissertation.

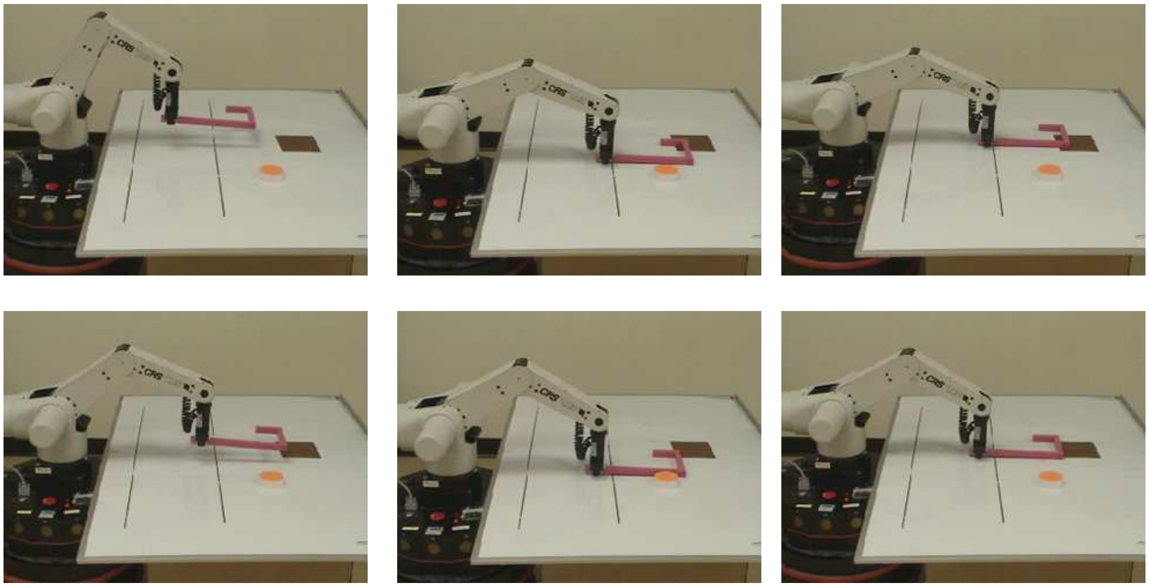


Figure 7.13: Using a broken tool (Part I: Adaptation) - Initially the robot tries to move the attractor towards the goal using the missing right hook. Because the puck fails to move as expected the robot reduces the replication probability of the affordances associated with this part of the tool.

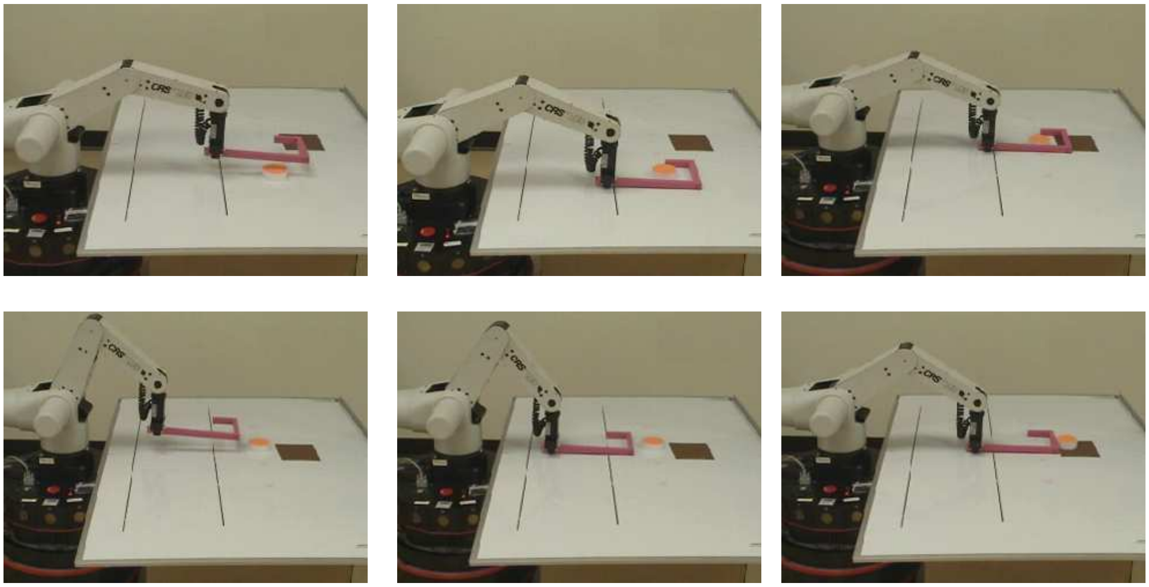


Figure 7.14: Using a broken tool (Part II: Solving the task) - After adapting to the modified affordances of the tool, the robot completes the task with the intact left hook.

7.8 Discussion: Behavioral Outcomes Instead of Geometric Shapes of Tools

It is important to emphasize that in the behavior-grounded approach taken here the representation of the tool is encoded entirely in terms exploratory behaviors. It is also important to note that the robot lacks information about the geometric shape of the tool. The shape of the tool is neither extracted nor used by the robot. The only perceptual feature that the robot detects about a given tool is its color, which serves as an unique ID for the tool.

The affordance table for the tool contains information only about the possible movements of the attractor object given a certain exploratory behavior (i.e., it contains only the arrows in Figure 7.9). The shape of the tool is shown in Figure 7.9 only for visualization purposes and in order to make the results more human readable. Thus, a more accurate visualization of the affordance table is the one shown in Figure 7.15. In the previous figure the boundary of the tool was presented only to help the human observer.

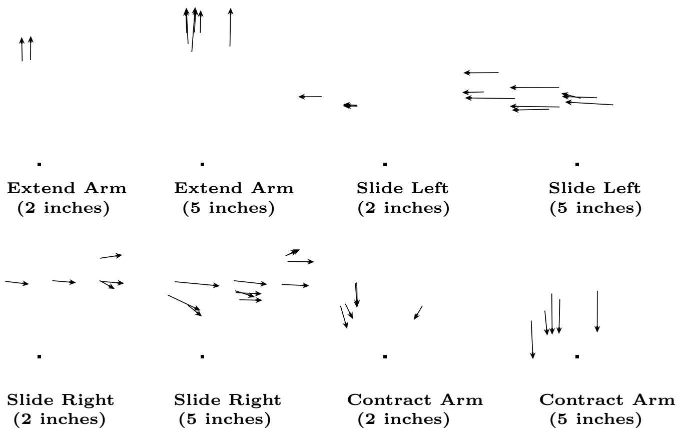


Figure 7.15: An alternative way to visualize the affordance table for the T-hook tool. The eight graphs show the same information as Figure 7.9. In this case, however, the shape of the tool (which is not detected by the robot) is not shown. The black square shows the position of the robot's wrist which is also the position of the grasp point (i.e., the square is the green body marker in Figure 7.3). This view of the affordance table is less human readable but shows better the representation of the tool from the point of view of the robot. Here the tool is represented only in terms of exploratory behaviors without extracting the shape of the tool.

To make this point even more obvious two other tools were designed – the V-stick and the Y-stick – as shown in Figure 7.16 and Figure 7.17. The two new tools are made of the same pine wood as the other five tools. Their top parts, however, are made of thin metal. While, their shapes are distinct to a human observer, to the robot they are indistinguishable. The reason for this is that the metal parts of the two tools are too thin to be detected by the robot's color tracking algorithm. Thus, the metal parts of the tools are indistinguishable from the background tracking noise. The color tracking results for the two tools are shown in Figure 7.18 and Figure 7.19.

The robot learned the affordances of both tools in the same way as it did for the other five tools. In other words, it was allowed to "play" with each tool for one hour. If the attractor object was pushed out of tool reach during that period the learning trial was suspended for a short time and the attractor was placed back on the table in a new random location.

The learned affordance representations for both tools are shown in Figure 7.20 and Figure 7.21. As can be seen from these figures, even though the robot cannot observe the top parts of the tools it nevertheless learned that they act in different ways as indicated by the movements of the attractor object. Furthermore, the robot was able to successfully use these invisible parts of the tools. This example shows once again that the behavior-grounded approach relies only coupling of robot behaviors and their observable outcomes.

For example, Figure 7.22 shows several frames from a sequence in which the robot was able to use the V-Stick successfully to push the attractor to the distant goal (goal G2 in Figure 7.11). This task was not always solvable with a straight stick as it was explained above. The affordances of the V-Stick, however, make this task solvable. The robot performs several pushing movements with the V-Stick alternating the right and left contact surface between the tool and the puck. As a result the puck takes a zig-zag path to the goal.

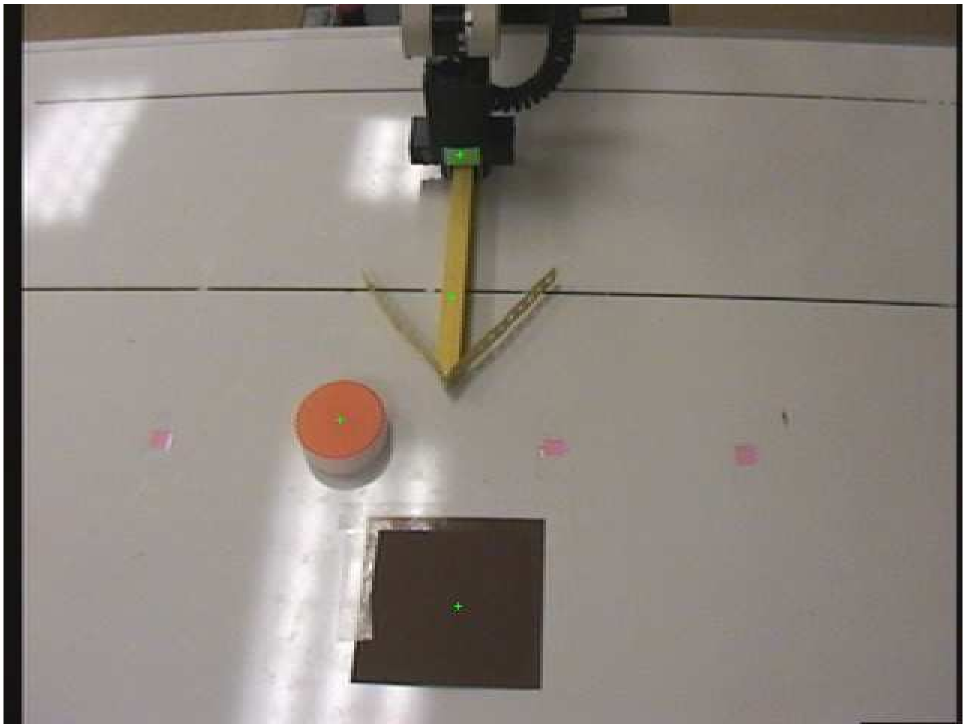


Figure 7.16: The robot holding the V-Stick tool.

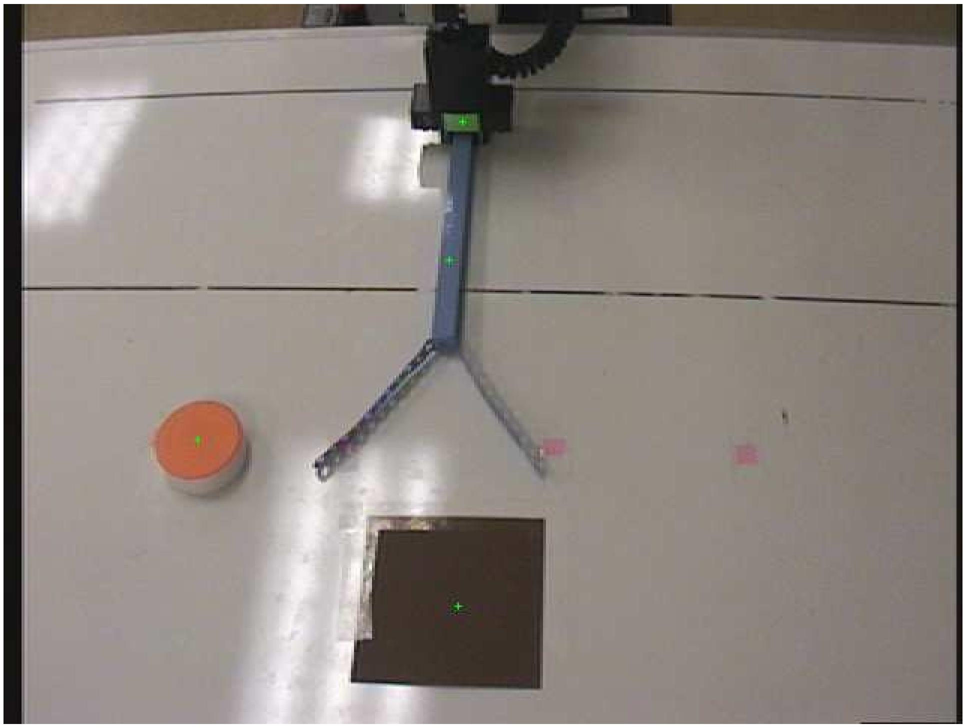


Figure 7.17: The robot holding the Y-stick tool.

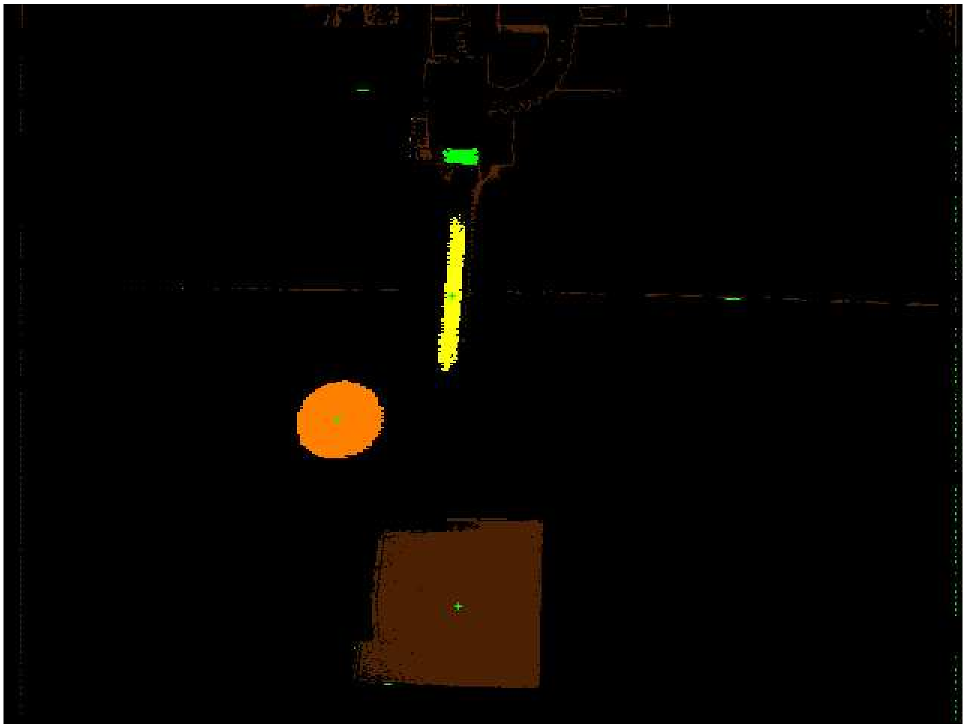


Figure 7.18: Color segmentation results for the V-stick tool.

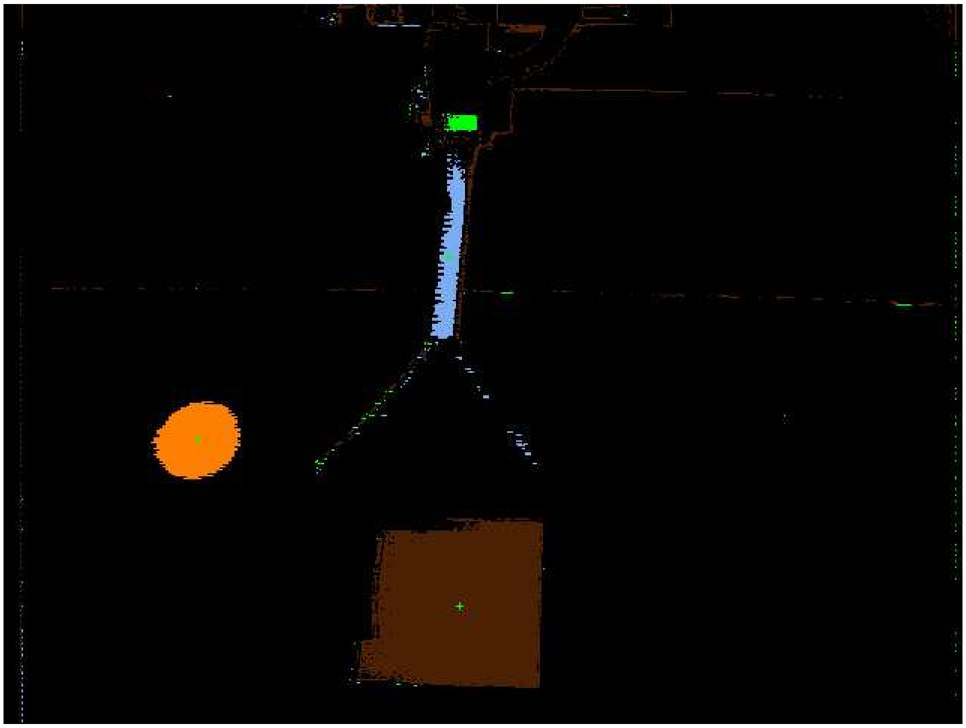


Figure 7.19: Color segmentation results for the Y-stick tool.

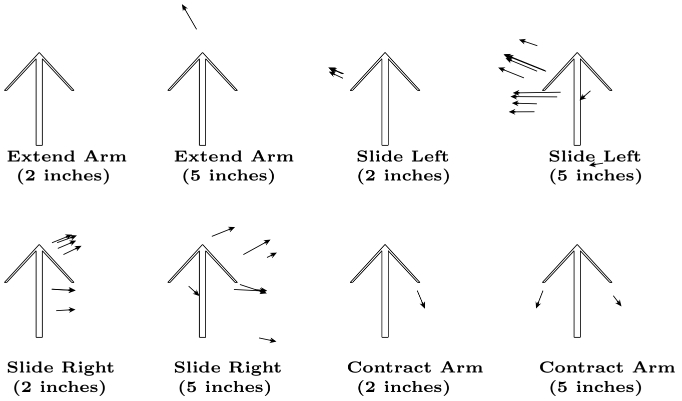


Figure 7.20: The learned affordance representation for the V-Stick tool.

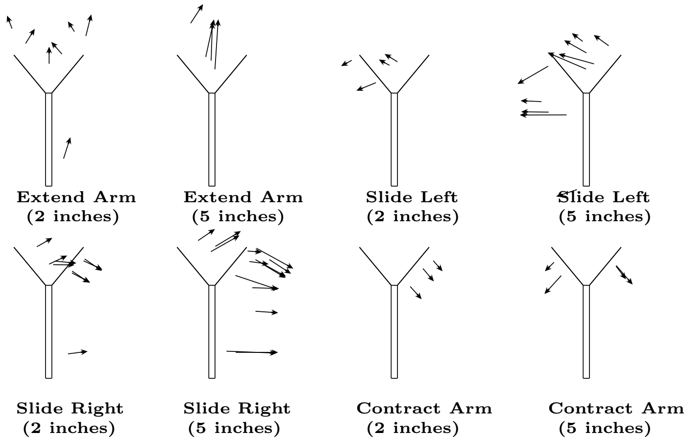


Figure 7.21: The learned affordance representation for the Y-Stick tool.

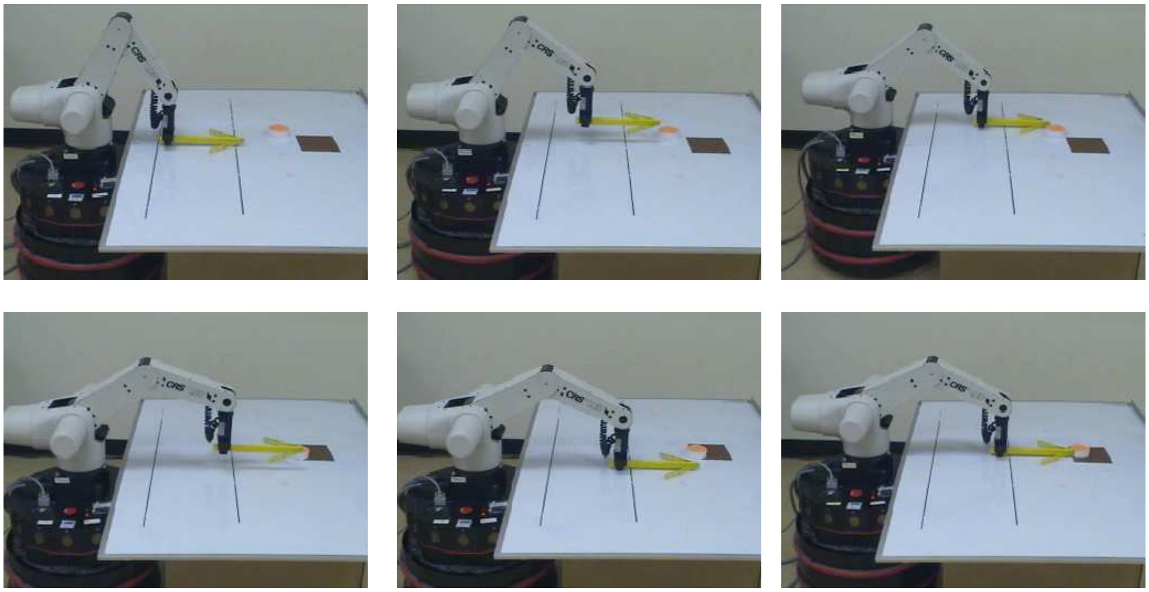


Figure 7.22: Frames from a sequence in which the robot uses the V-Stick to push the puck towards the away goal. The robot performs several pushing movements with the V-Stick alternating the right and left contact surface between the tool and the puck. As a result the puck takes a zig-zag path to the goal.

7.9 Chapter Summary

This chapter introduced a novel approach to representing and learning tool affordances by a robot. The affordance representation is *grounded* in the behavioral and perceptual repertoire of the robot. More specifically, the affordances of different tools are represented in terms of a set of exploratory behaviors and their resulting effects. It was shown how this representation can be used to solve tool-using tasks by dynamically sequencing exploratory behaviors based on their expected outcomes. The results described in this chapter answer affirmatively the third research question stated in Section 1.2 which asked whether a robot can use exploratory behaviors to both learn and represent the functional properties or affordances of tools.

The behavior-grounded approach represents the tools' affordances in concrete terms (i.e., behaviors) that are available to the robot's controller. Therefore, the robot can directly test the accuracy of its tool representation by executing the same set of exploratory behaviors that was used in the past. If any inconsistencies are detected in the resulting observations they can be used to update the tool's representation. Thus, the accuracy of the representation can be directly tested and verified by the robot as mandated by the verification principle (see Section 3.2). It was demonstrated how the robot can use this approach to adapt to changes in the tool's properties over time, e.g., tools that can break.

A shortcoming of the behavior-grounded approach is that there are tool affordances that are unlikely to be discovered since the required exploratory behavior is not available to the robot. This problem has also been observed in animals, e.g., macaque monkeys have significant difficulties learning to push an object with a tool away from their bodies because this movement is never performed in their normal daily routines Ishibashi et al. (2000). This problem can be resolved, however, if the ability to learn new exploratory behaviors is added.

There are some obvious extensions to this work that are left for future work. First, the current implementation starts the exploration of a new tool from scratch even though it may be similar to an already explored tool. Adding the ability to rapidly infer the affordances of a new tool from its shape similarity to previous tools would be a nice extension.

Second, the current implementation uses a purely random behavioral babbling exploration procedure. Different strategies that become less random and more focused as information is structured by the robot during the exploration could be used to speed up the learning process.

Third, the behavior-grounded approach should be compared experimentally with planners for pushing objects (e.g.,Mason (2001)). It is expected that the behavior-grounded method would approach asymptotically the accuracy of these planners as the number and diversity of the exploratory behaviors is increased. It is also expected, however, that the behavior-grounded approach would excel in situations that cannot be predicted by the planners, e.g., tools that can break or objects whose center of mass can shift between trials.

CHAPTER VIII

CONCLUSIONS AND FUTURE WORK

This dissertation makes several important contributions to the field of robotics. The main contributions of the dissertation can be broadly summarized as follows:

- It demonstrates a method for autonomous self-detection in robots (Chapter 5).
- It demonstrates a model for extendable robot body schema which can be used to achieve video-guided behaviors (Chapter 6).
- It demonstrates a behavior-grounded method for learning the affordances of tools which can also be used to solve tool-using tasks (Chapter 7).

These contributions are described and analyzed in more detail below.

8.1 Self-Detection in Robots

Chapter 5 described a method for autonomous self-detection by a robot. The method detects the temporal contingency between motor commands (efferent signals) and visual movements (afferent signals) to estimate the efferent-afferent delay of the robot. In compliance with the verification principle and the subjectivity principle, Chapter 5 showed how the robot can estimate its own efferent-afferent delay from self-observation data. It was shown how the robot can estimate this delay while performing random joint movements similar to the primary circular reactions described by Piaget (1952).

Using the value of this delay a robot can learn which features belong to its own body and which to the environment, which is an important contribution of this research. It was shown that the self-detection algorithm is robust under a variety of test conditions. For example, the algorithm is not affected by the presence of other moving objects in the environment; the algorithm performed well even when a second independently moving robot was present in the environment. Using this approach, the robot was also able to detect its self-image in a TV monitor, which is another important contribution of this research.

The experimental results described in Chapter 5 show that a robot can successfully distinguish between its own body and the external environment. These results directly support the first research question stated in Section 1.2. The robot was able to correctly classify different visual stimuli as either “self” or “other.”

Another contribution of this dissertation was applying the concept of temporal contingency from the Psychology literature (e.g., the work of Watson (1985, 1994)) to the robotics field and implementing a computational model which uses this concept on a real robot. The results described in Chapter 5 demonstrate that Watson’s general framework is appropriate for application on robots. The results also showed, however, that his model has some shortcomings which were reported here for the first time and resolved successfully. Future studies can use this modification for self-detection experiments with either robots or infants.

One limitation of the current implementation is that it makes the assumption that visually distinct features (color markers) exist on the surface of the robot’s body which can be reliably identified and tracked. The color markers were chosen in order to solve the tracking and correspondence problems in a computationally efficient way. In retrospect, this decision was probably not the most optimal one. Reliable color tracking for extended periods of time remains a challenging problem. The number of distinct colors that can be tracked reliably for long periods of time also turned out to be quite small, which constrained the experimental designs. Future work should focus on either eliminating the need for distinct perceptual features or adding the ability to learn these features from actual experience.

8.2 Extendable Body Schema Model for Robots

Chapter 6 described an extendable body schema model for robots. The basic model is based on the *Self-Organizing Body Schema* model described by Morasso and Sanguineti (1995). Chapter 6 modified their model such that the body schema model has extensibility properties similar to its biological analog.

In this chapter, the extension mechanism is triggered by the movement coincidence between perceptual features associated with the robot's body and perceptual features associated with an external object. The external object can even be an image of the robot in a TV monitor. Thus, no physical contact is required to trigger the extension mechanism.

This chapter also described an experiment involving a reaching task. The robot, however, was not allowed to observe its own movements directly. Instead, it was only able to observe its own movements through a real-time TV image. The robot was able to detect the temporal contingency between its own motor actions and the movements of the visual features in the TV monitor. Based on this information, the robot was able to extend its body schema and use the video image as feedback in order to execute the reaching behavior. The robot was able to reach for the object even when the TV image was rotated by 50 degrees. This experiment constitutes the first-ever demonstration of a video-guided behavior by a robot. This experiment also showed the usefulness of the body schema for control of robot behaviors when the robot can receive feedback about its own actions only through indirect means (e.g., through a TV monitor).

One shortcoming of the current implementation is that it relies on the existence of unique visual features that are placed on the robot's body. Color tracking was also unreliable under large magnifications of the TV image. Future work should focus on modifying the body schema model to work with features that are not based on color and which are not necessarily unique.

Another shortcoming of the current implementation is that it does not take into account the dynamics of the robot and the attached tool. For example, a long stick may have the potential to extend the reach of the robot but if the stick is too heavy this extension may not be achievable for some or all joint configurations of the robot.

The results described in Chapter 6 demonstrate that a robot can learn a sensorimotor model of its own body from self-observation data, which answers affirmatively the second research question stated in Section 1.2. The results described in this chapter also show that the pliable body representation can be used to facilitate the execution of goal-oriented tasks such as video-guided behaviors, which is an important contribution of this dissertation.

Future work can build upon the principles described in Chapter 6 to achieve video-guided robot behaviors for other tasks. For example, one task with many potential practical applications is using a mouse to position a mouse cursor on a computer monitor. Another task with numerous practical applications to use a rear view camera (or mirror) to back up a car or a truck.

8.3 Learning of Tool Affordances

Chapter 7 described a novel method for learning tool affordances by a robot. The tool affordances are represented in terms of the expected outcomes of exploratory behaviors. In other words, the affordances of the tools are grounded in the behavioral repertoire of the robot. Because the robot learns the tool representation using the verification principle the robot can independently test and verify its own representation. Furthermore, the robot can autonomously correct this representation when the tools don't behave as expected. For example, the robot was able to relearn the affordances of a tool when the tool was broken. Thus, the verification principle gives the robot the ability to autonomously detect errors in its own representation. Consequently, the robot has the ability to perform error driven learning. A robot that was programmed to use a specific tool in a specific way would never be able to recover from such a failure.

The work presented in Chapter 7 represents one of the first computational models of tool affordances. When I first started working in this area the term *affordance* was not even familiar to many roboticists. Over the last couple of years, however, the interest in this area has been growing steadily and it seems that soon affordance research may become mainstream robotics. For example, in 2006 in Dagstuhl, Germany there was an workshop entitled: *Towards Affordance-Based Robot Control.*[1]

The results presented in Chapter 7 also demonstrate that a robot can use exploratory behaviors to both learn and represent the functional properties or affordances of tools. This answers affirmatively the third research question stated in Section 1.2.

One shortcoming of the behavior-grounded approach is that there are tool affordances

[1] See http://www.dagstuhl.de/de/programm/kalender/semhp/?semid=28425

that are unlikely to be discovered by the robot because it lacks the required exploratory behavior. For example, if the robot does not have a pushing behavior in its behavioral repertoire it will never be able to discover the pushing affordance of a new tool. This problem can be solved, however, if the ability to learn new exploratory behaviors is added. The best way to achieve this would be through social interaction with humans who are experienced tool users. Thus, future work should address the problem of social learning of exploratory behaviors through human-robot interaction.

Another shortcoming of the current implementation is that the robot uses a purely random behavioral babbling exploration procedure to learn the affordances of all tools. A better option would be to use human feedback during the learning process to narrow down the exploration space. For example, the human can suggest which behaviors the robot should use to explore the tool. The human can also focus the attention of the robot to a specific part of the tool which is critical for solving a specific task.

Yet another shortcoming of the current implementation is that the robot explores all new tools in the same way. It start the exploration with an empty affordance table and adds new affordances as they are discovered through behavioral babbling. If a new tool is similar to an already explored tool the robot should be able to explore this similarity to speed up the exploration process. Currently, this is not possible because the method ignores the shape of the tools completely. The shape is neither detected nor used in any way. The tool representation is entirely based on the observable outcomes that the movements of the tool produce on the attractor object. A natural extension would be to learn to associate the affordances with specific local shapes of the tools. This can later be used to infer the affordances of a novel tool without actually performing any exploratory behaviors.

8.4 Future Work

This dissertation began with the vision that one day there will intelligent robots which can overcome the limitations of their bodies by using tools and thus would be more useful than the robots of today. This dissertation made a small step in that direction. There are still significant challenges, however, which must be overcome before we can have truly autonomous tool-using robots. This section describes four broad areas for future work that can bring us even closer to that vision:

- **Other Modes of Tool Use -** The experiments described in this dissertation have focused on only one tool-using mode: extension of reach. Designing robot algorithms and representations which can address the amplification of force and the control of liquids tool-using modes seem to be a fruitful area of future work. Both of these modes, however, are very challenging to implement on a robot. For example, learning that a hammer can be used to crush objects is a non-trivial task because the crushed object will be totally transformed after the action of the hammer. The sensing challenges for liquids are even more difficult. Robots that can learn to control liquids autonomously would be very useful, especially as household robots that can assist with everyday chores or nurse robots that can take care of elderly people in homes and hospitals.
- **Tool Modification -** The experiments described in this dissertations used only rigid and non-deformable tools. Both humans and animals, however, are capable of using deformable tools. For example, under lab conditions a New Caledonian crow was observed to consistently bend a piece of wire into a hook in order to extract food rewards that would be impossible to get otherwise (Weir et al., 2002). There is not a single robot today that can autonomously solve this task.
- **Tool Manufacture -** An even more challenging area for future work is tool manufacture. While tool modification focuses on modifying a single tool, tool manufacture focuses on combining two tools to build a new tool. Tool manufacture in animals was first reported by Köhler (1931) who observed that one of his chimpanzees was able to combine two hollow sticks into a longer stick by inserting one into the other.
- **Tools That Extend Sensory Abilities -** Another fruitful area of future work deals with tools that extend the sensory abilities of the robot such as - magnifying glasses, night vision goggles, or earphones. For example, a magnifying glass can be used to study the shapes and forms of small objects. Before we can have robots that can achieve this, however, we will need to design some very pliable models for the organization of the robot's senses. The

challenges from a perceptual perspective are also significant.

In all of the tasks mentioned above the real challenge is not solving the individual task because for any concretely specified task an engineered solution can be designed. Instead, the real challenge is how to design robot algorithms which can solve tasks like these autonomously. In other words, the algorithms and the internal representations that the robot uses must be independent of any task.

Finally, I believe that research in autonomous tool using robots is only going to increase in importance as robots begin to spread to human inhabited environments. Some of that has already began to happen. However, the really exciting applications of autonomous tool using robots are still ahead of us.

Bibliography

Aboaf, E. W., Drucker, S. M., and Atkenson, C. G. Task-level robot learning: Juggling a tennis ball more accurately. In *IEEE International Conference on Robotics and Automation*, pages 1290–1295, Scottsdale, AZ, 1989.

Aglioti, S., Smania, N., Manfredi, M., and Berlucchi, G. Disownership of left hand and objects related to it in a patient with right brain damage. *NeuroReport*, 8:293–296, 1997.

Aguilar, M. and Stiles, W. S. Saturation of the rod mechanism of the retina at high levels of stimulation. *Optica Acta.*, 1:59–65, 1954.

Aiyama, Y., Inaba, M., and Inoue, H. Pivoting: A new method of graspless manipulation of object by robot fingers. In *IEEE/RSJ International conference on Intelligent Robots and Systems*, pages 136–143, Yokohama, Japan, 1993.

Alcock, J. The origin of tool-using in egyptian vultures(neophron percnopterus). *Ibis*, 112:524, 1970.

Alcock, J. The evolution of the use of tools by feeding animals. *Evolution*, 26:464–473, 1972.

Alcock, J. *Animal Behavior: An Evolutionary Approach.* Sinauer, Assoc. Inc., Sunderland, MA., 1975.

Allen, P. Mapping haptic exploratory procedures to multiple shape representations. In *IEEE International Conference on Robotics and Automation*, pages 1679–1684, 1990.

Ambrose, R., Aldridge, H., Askew, R., and Burridge, R. Robonaut: Nasa's space humanoid. *IEEE Intelligent Systems and Their Applications*, 15(4):57–63, 2000.

Arkin, R. C. *Behavior-based Robotics.* MIT Press, 1998.

Asada, H. and Asari, Y. The direct teaching of tool manipulation skills via the impedance identification of human motions. In *IEEE International Conference on Robotics and Automation*, pages 1269–1274, 1988.

Atkeson, C. G., Moore, A. W., and Schaal, S. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

Atkeson, C. G. and Schaal, S. Memory-based neural networks for robot learning. *Neurocomputing*, 9(3):243–69, 1995.

Ayer, A. J. *Language Truth and Logic.* Dover Publications, Inc., New York, 1952.

Bahrick, L. E. and Watson, J. S. Detection of intermodal proprioceptive-visual contingency as a basis of self perception in infancy. *Developmental Psychology*, 21:963–973, 1985.

Balch, T. *Behavioral Diversity in Learning Robot Teams.* PhD thesis, Georgia Institute of Technology, 1998.

Ballard, D. H. and Brown, C. M. *Computer Vision.* Prentice Hall, 1982.

Barnett, S. A. Exploratory behavior. *British Journal of Psychology*, 49:289–310, 1958.

Barr, A. H. Superquadratics and angle preserving transformations. *IEEE Computer Graphics and Applications*, 1:11–23, 1981.

Barth, J., Povinelli, D., and Cant, J. Bodily origins of self. In Beike, D., Lampinen, J. L., and Behrend, D. A., editors, *The Self and Memory*, pages 11–43. Psychology Press, New York, 2004.

Beck, B. B. *Animal Tool behavior: The use and manufacture of tools by animals.* Garland STMP Press, New York, 1980.

Bentivegna, D. and Atkeson, C. Learning from observation using primitives. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1988 –1993, 2001.

Berlucchi, G. and Aglioti, S. The body in the brain: neural bases of corporeal awareness. *Trends in Neuroscience*, 20(12):560– 564, 1997.

Berlyne, D. E. Novelty and curiosity as determinants of exploratory behavior. *British Journal of Psychology*, 41:68–80, 1950.

Berlyne, D. E. Curiosity and exploration. *Science*, pages 25–33, 1966.

Berthoz, A. *The brains's sense of Movement.* Harvard University Press, Cambridge, MA, 2000. Translated by Giselle Weiss.

Berti, A. and Frassinetti, F. When far becomes near: Remapping of space by tool use. *Journal of Cognitive Neuroscience*, 12(3):415–420, 2000.

Bicici, E. and St. Amant, R. Reasoning about the functionality of tools and physical artifacts. (Under review; available from http://www.csc.ncsu.edu/faculty/stamant/papers/index.html) (Date accessed: March 2007), 2007.

Bingham, H. C. Selective transportation by chimpanzee. *Comparative psychology monographs*, 5(26), 1929.

Boesch, C. and Boesch, H. Optimization for nut-cracking with natural hammers by wild chimpanzees. *Behavior*, 83:265–286, 1983.

Boesch, C. and Boesch, H. Tool use and tool making in wild chimpanzees. *Folia Primatologica*, 54:86–99, 1990.

Bogoni, L. *Identification of functional features through observations and interactions.* PhD thesis, University of Pennsylvania, 1995.

Bogoni, L. and Bajcsy, R. Interactive recognition and representation of functionality. *Computer Vision and Image Understanding*, 62(2):194–214, 1995.

Botvinick, J. and Cohen, J. Rubber hands 'feel' touch that eyes can see. *Nature*, 391: 756, 1998.

Bowlby, J. *Attachment (Volume I).* Basic Book, New York, 1969.

Brady, M., Agre, P. E., Braunegg, D. J., and Connell, J. H. The mechanics mate. In O'Shea, T., editor, *Advances in Artificial Intelligence*, pages 79–94. Elsevier Science Publishers, New York, 1985.

Brady, M. and Asada, H. Smoothed local symmetries and their implementation. *International Journal of Robotics Research*, 3(3), 1984.

Brooks, R., Breazeal, C., Marjanovic, M., Scassellati, B., and Williamson, M. The cog project: Building a humanoid robot. In Nehaniv, C., editor, *Computation for Metaphors, Analogy, and Agents (Lecture Notes in Artificial Intelligence 1562)*, pages 52–87. Springer, New York, 1999.

Brooks, R. and Stein, L. Building brains for bodies. *Autonomous Robots*, 1(1):7–25, November 1994.

Brooks, R. A. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 17:285–348, 1980.

Bruce, J., Balch, T., and Veloso, M. Fast and inexpensive color image segmentation for interactive robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Japan, October 2000.

Calvin, W. A stone's throw and its launch window: Timing precision and its implications for language and hominid brains. *Journal of Theoretical Biology*, 104:121– 135, 1983.

Cameron, J., MacKenzie, D., Ward, K., Arkin, R. C., and Book, W. Reactive control for mobile robot manipulation. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 228–235, Atlanta, GA, May 1993.

Campbell. *Human Ecology.* Aldine, New York, 1984.

Campbell, B. *Human Evolution: An Introduction to Man's Adaptations.* Aldine, New York, third edition, 1985.

Catania, A. C. Reinforcement schedules and psychophysical judgments: A study of some temporal properties of behavior. In Schoenfeld, W. N., editor, *The theory of reinforcement schedules*, pages 1–42. Appleton-Century-Crofts, New York, 1970.

Chevalier-Skolnikoff, S. and Liska, J. Tool use by wild and captive elephants. *Animal Behavior*, 46:209–219, 1993.

Clark, A. *Being there : putting brain, body, and world together again.* MIT Press, Cambridge, MA, 1997.

Collins, D. *The Human Revolution: from Ape to Artist.* Phaidon Press, Oxford, 1976.

Connell, J. H. and Brady, M. Generating and generalizing models of visual objects. *Artificial Intelligence*, 31:159–183, 1987.

CRSplus. *Technical Manual for A100/200 Series Small Industrial Robot System.* CRS Plus Inc., Burlington, Ontario, Canada, 1990.

Cutkosky, M. R. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279, 1989.

Darwin, C. *The descent of man and selection in relation to sex*, volume I. Murray, London, 1871.

de Waal, F. *Bonobo: The Forgotten Ape.* University of California Press, Berkeley, 1997.

de Waal, F. *The ape and the sushi master : cultural reflections by a primatologist.* Basic Books, New York, 2001.

de Waal, F. B. M., Dindo, M., Freeman, C. A., and Hall, M. J. The monkey in the mirror: Hardly a stranger. *Proceedings of the National Academy of Sciences (PNAS)*, 102(32):11140–11147, 2005.

Demasio, A. R. *Descartes' Error: Emotion, Reason and the Human Brain.* Gosset/Putnam Pres, New York, 1994.

Dennett, D. C. *The Intentional Stance.* MIT Press, 1989.

Diamond, J. M. Guns, germs, and steel: The fates of human societies, 1999.

DiManzo, M., Trucco, E., Giunchiglia, F., and Ricci, F. Fur: Understanding functional reasoning. *International Journal of Intelligent Systems*, 4:431–457, 1989.

Donald, B. R., Gariepy, L., and Rus, D. Distributed manipulation of objects using ropes. In *International Symposium on Experimental Robotics (ISER)*, 1999.

Donald, B. R., Gariepy, L., and Rus, D. Distributed manipulation of multiple objects using ropes. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 450–475, 2000.

Durst, R. and Krotkov, E. Object classification from acoustic analysis of impact. Technical Report CMU-RI-TR-93-16, Robotics Institute, Carnegie Mellon University, June 1993.

Eddy, T. J., Gallup, G. G. J., and Povinelli, D. J. Age differences in the ability of chimpanzees to distinguish mirror-images of self from video images of others. *Journal of Comparative Psychology*, 110(1):38–44, March 1996.

Ellis, J. Orangutan tool use at the oklahoma city zoo. *The Keeper*, 1:5–6, 1975.

Epstein, R., Lanza, R. P., and Skinner, B. F. "self-awareness" in the pigeon. *Science*, 212:695–696, 1981.

Erdmann, M. A. An exploration of nonprehensile two palm manipulation using two zebras. In Laumond, J.-P. and Overmars, M., editors, *Algorithms for Robotic Motion and Manipulation : 1996 Workshop on the Algorithmic Foundations of Robotics*, pages 239–254, Wellesley, MA, 1996. A. K. Peters.

Erdmann, M. A. and Mason, M. T. An exploration in sensorless manipulation. *IEEE Journal of Robotics and Automation*, 4(4):369–379, August 1986.

Farné, A. and Ládavas, E. Dynamic size-change of hand peripersonal space following tool use. *NeuroReport*, 11(8):1645–1649, June 2000.

Fitzpatrick, P., Metta, G., Natale, L., Rao, S., and Sandini, G. Learning about objects through action - initial steps towards artificial cognition. In *IEEE ICRA*, Taipei, Taiwan, May 12-17, 2003.

Flom, R. and Bahrick, L. E. Is featural information necessary for 5-month-olds' perception of motion specifying the self? In *Society for Research in Child Development*, Minneapolis, MN, April 2001.

Frederiks, J. A. M. Disorders of the body schema. In Vinken, P. J. and Bruyn, G. W., editors, *Handbook of Clinical Neurology: Disorders of Speech Perception and Symbolic Behavior*, volume 4, pages 207–240. North Holland, 1969.

Freud, S. *New introductory lectures on psychoanalysis*. Norton, New York, 1965.

Fujita, M., Kuroki, Y., Ishida, T., and Doi, T. Autonomous behavior control architecture of entertainment humanoid robot sdr-4x. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 960– 967, Oct. 27-31 2003.

Gallistel, C. Time has come. *Neuron*, 38(2):149–150, 2003.

Gallistel, C. R. Coordinate transformations in the genesis of directed action. In Bly, B. O. M. and Rummelhart, D. E., editors, *Cognitive science*, pages 1–42. Academic, New York, 1999.

Gallistel, C. R. and Gibbon, J. Time, rate and conditioning. *Psychological Review*, 107:289–344, 2000.

Gallup, G., Anderson, J., and Shillito, D. The mirror test. In Bekoff, M., Allen, C., and Burghardt, G., editors, *The Cognitive Animal: Empirical and Theoretical Perspectives on Animal Cognition*. MIT Press, Cambridge, MA, 2002.

Gallup, G. G. Self recognition in primates: A comparative approach to the bidirectional properties of consciousness. *American Psychologist*, 32:329–338, 1977.

Gallup, G. G., Wallnau, L. B., and Suarez, S. D. Failure to find self-recognition in mother-infant rhesus monkey pairs. *Folia Primatologica*, 32(210-219), 1980.

Gallup, G. G. J. Chimpanzees: self-recognition. *Science*, 167(3914):86–7, Jan. 2 1970.

Gardiner, M. The superellipse: A curve between the ellipse and the rectangle. *Scientific American*, 213:222–234, 1965.

Gerkey, B., Vaughan, R. T., and Howard, A. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, pages 317–323, Coimbra, Portugal, June 2003.

Gibbon, J. Scalar expectancy theory and weber's law in animal timing. *Psychological Review*, 84(3):279–325, 1977.

Gibbon, J. Origins of scalar timing. *Learning and Motivation*, 22(1-2):3–38, February-May 1991.

Gibbon, J. and Church, R. M. Sources of variance in an information processing theory of timing. In Roitblat, H., Bever, T., and Terrace, H., editors, *Animal cognition*, pages 465–488. L. Erlbaum Associates, Hillsdale, NJ, 1984.

Gibbon, J., Malapani, C., Dale, C. L., and Gallistel, C. Toward a neurobiology of temporal cognition: advances and challenges. *Current Opin Neurobiology*, 7(2):170–84, 1997.

Gibbs, Jr., R. W. *Embodiment and Cognitive Science*. Cambridge University Press, 2006.

Gibson, E. J. *Principles of perceptual learning and development*. Appleton-Century-Crofts, New York, 1969.

Gibson, J. J. *The ecological approach to visual perception*. Houghton Mifflin, Boston, 1979.

Girdner, J. B. An experimental analysis of the behavioral effects of a perceptual consequence unrelated to organic drive states. *American Psychologist*, (8):354–355, 1953.

Glenberg, A. What memory is for. *Behavioral and Brain Sciences*, 20(1):1–55, 1997.

Glickman, S. E. and Sroges, R. W. Curiosity in zoo animals. *Behaviour*, 27:151–187, 1966.

Go, J., Browning, B., and Veloso, M. Accurate and flexible simulation for dynamic, vision-centric robot. Technical Report CMU-CS-04-180, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, November 2004.

Goldstein, M. H., King, A. P., and West, M. J. Social interaction shapes babbling: Testing parallels between birdsong and speech. *Proceedings of the National Academy of Sciences (PNAS)*, 100(13), June 24 2003.

Goodal, J. Tool-using and aimed throwing in a community of free living chimpanzees. *Nature*, 201:11264–1266, 1964.

Goodall, J. and van Lawick, H. Use of tools by egyptian vultures. *Nature*, 212:1468–1469, 1966.

Graziano, M. S., Taylor, C. S., and Moore, T. Complex movements evoked by microstimulation of precentral cortex. *Neuron*, 30 May 2002.

Graziano, M. S. A., Cooke, D. F., and Taylor, C. S. R. Coding the location of the arm by sight. *Science*, 290:1782–1786, December 2000.

Green, K., Eggert, D., Stark, L., and Bowyer, K. W. Generic recognition of articulated objects through reasoning about potential function. *Computer vision and Image understanding*, 62(2):177–194, September 1995.

Hall, K. and Schaller, G. B. Tool using behavior of the california sea otter. *Journal of Mammalogy*, 45:287–298, 1964.

Haralick, R. M. and Shapiro, L. G. *Computer and Robot Vision*, chapter 18 Object Models and Matching. Addison-Wesley, 1993.

Harnad, S. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

Harrisson, B. *Orang-utan.* Boubleday, Garden City , N.J., 1963.

Hayes, P. J. The second naive physics manifesto. In *Formal Theories of the Commonsense World.* Ablex, 1985.

Head, H. and Holmes, G. Sensory disturbances from cerebral lesions. *Brain*, 34:102–254, 1911.

Hervé, J.-Y., Sharma, R., and Cucka, P. Geometry of visual coordination. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, pages 732–737, 1991.

Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. The development of honda humanoid robot. In *Proceedings of International Conference of Robotics and Automation*, Leuven, Belgium, May 1998.

Hodges, J. Naive mechanics - a computational model of device use and function in design improvisation. *IEEE Expert*, pages 14–27, February 1992.

Hodges, J. Naive mechanics: a taxonomy of representation primitives for representing and reasoning about problem solving with simple mechanical device. In *AAAI-93 Workshop on Reasoning about Function*, page 266, Washington D.C., July 1993.

Hodges, J. Functional and physical object characteristics and object recognition in improvisation. *Computer Vision and Image Understanding*, 62(2):147–163, 1995.

Huang, W., Krotkov, E. P., and Mason, M. T. Impulsive manipulation. In *IEEE International Conference on Robotics and Automation*, pages 120–125, 1995.

Hunt, J. M. Intrinsic motivation and its role in psychological development. In Levine, D., editor, *Nebraska symposium on motivation*, pages 189–282, Lincoln, 1965. University of Nebraska Press.

Inglis, I. R. Towards a cognitive theory of exploratory behavior. In Archer, J. and Birke, L., editors, *Exploration in Animals and humans*, pages 72–115. Van Nostrand Reinhold, Berkshire, England, 1983.

Iriki, A. *Personal communication.* (via e-mail), November 25 2003.

Iriki, A., Tanaka, M., and Iwamura, Y. Coding of modified body schema during tool use by macaque postcentral neurons. *Neuroreport*, 7:2325–2330, 1996.

Iriki, A., Tanaka, M., Obayashi, S., and Iwamura, Y. Self-images in the video monitor coded by monkey intraparietal neurons. *Neuroscience Research*, 40:163–173, 2001.

Ishibashi, H., Hihara, S., and Iriki, A. Acquisition and development of monkey tool-use: behavioral and kinematic analyses. *Canadian Journal of Physiology and Pharmacology*, 78:958–966, 2000.

Itakura, S. Mirror guided behavior in japanese monkeys (macaca fuscata fuscata). *Primates*, 28:149–161, 1987.

J., G. E. Exploratory behavior in the development of perceiving, acting, and the acquiring of knowledge. *Annual Review of Psychology*, 39:1–41, 1988.

Jones, S. S. Infants learn to imitate by being imitated. In *Proceedings of the Fifth International Conference on Development and Learning (ICDL)*, Bloomington, IN, May 31- June 3 2006.

Kang, S. *Robot instruction by human demonstration.* PhD thesis, Robotics Institute, Carnegie Mellon University, 1994.

Kang, S. and Ikeuchi, K. Toward automatic robot instruction from perception– temporal segmentation of tasks from human hand motion. *IEEE Transactions on Robotics and Automation*, 11(5):670–681, October 1995.

Kawamura, S. The process of sub-culture propagation among japanese macaques. *Primates*, (2):43–60, 1959.

Keeley, L. H. The functions of paleolithic flint tools. In Isaac, G. and Leakey, R., editors, *Human Ancestors: Readings from Scientific American*, pages 102–109. W. H. Freeman and Co., 1979. (Originally published in Nov. 1977).

Kimura, D. Neuromotor mechanisma in the evolution of human communication. In Steklis, H. D. and Raleigh, M. J., editors, *Neurobiology of Social Communication in Primates: An Evolutionary Perspective*, pages 197–219. Academic Press, New York, 1979.

Kise, K., Ashida, M., Dan, S., and Kitahashi, T. A function-based model for simple tools and its application to recognition of 3d shapes. Technical report, JSAI, Tech. Gr. SIG-FAI, 1993.

Kise, K., Ashida, M., Dan, S., and Kitahashi, T. A further consideration on the functional models. In *Information Modeling in Knowledge Bases V*, pages 186–190. IOS, Amsterdam, 1994.

Klüver, H. *Behavior Mechanisms in Monkeys.* Behavior Research Fund Monographs. University of Chicago Press, Chicago, IL, 1933.

Köhler, W. *The mentality of apes.* Harcourt, Brace, and Co., New York, 1931. Translated by Ella Winter.

Kotosaka, S. and Schaal, S. Synchronized robot drumming by neural oscillator. *Journal of the Robotics Society of Japan*, 19(1):116–123, 2001.

Krotkov, E. Perception of material properties by robotic probing: Preliminary investigations. In *International . Joint Conference of Artificial Intelligence*, pages 88–94, Montreal, August 1995.

Lack, D. Darwin's finches. *Scientific America*, (188):68–72, 1953.

Lewis, M. and Brooks-Gunn, J. *Social cognition and the acquisition of self.* Plenum Press, New York, 1979.

Li, L., Cox, B., Diftler, M., Shelton, S., and Barry. Development of a telepresence controlled ambidextrous robot for space application. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 58–63, 1996.

Lieberman, P. *Uniquely Human: The Evolution of Speech, Thought, and Selfless Behavior.* Harvard University Press, Cambridge, MA., 1991.

Littman, M. L., Sutton, R. S., and Singh, S. Predictive representaiton of state. In *Advances in Neural Information Processing Systems*, volume 14, 2002.

Lorenz, K. Innate bases of learning. In Pribram, K. H. and King, J., editors, *Learning as Self-Organization*. Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey, 1996.

Lowry, M. R. Reasoning between structure and function. In *In Proceedings of the Image Understanding workshop*, pages 260–264. Science Applications, Inc., 1982.

Lynch, K. M. Issues in nonprehensile manipulation. In Agarwal, P. K., Kavraki, L. E., and Mason, M. T., editors, *Robotics the Algorithmic perspective: The Third Workshop on the Algorithmic Foundations of Robotics*, pages 237–250. A. K. Peters, Natick, MA, 1998.

Lynch, K. M. and Mason, M. T. Dynamic nonprehensile manipulation: Controllability, planning, and experiments. *International Journal of robotics research*, 18(1):64–92, 1999.

MacKenzie, D. and Arkin, R. Behavior-based mobile manipulation for drum sampling. In *IEEE International Conference on Robotics and Automation*, pages 2389–2395, Minneapolis, MN, April 1996.

MacKenzie, D., Cameron, J., and Arkin, R. C. Multiagent mission specification and execution. *Autonomous Robots*, 4(1):29–57, January 1997.

Mah, C. D. and Mussa-Ivaldi, F. A. Evidence for a specific internal representation of motion-force relationships during object manipulation. *Biological Cybernetics*, 88(1): 60–72, 2003.

Mason, M. T. *Manipulator Grasping and Pushing Operations*. PhD thesis, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1982.

Mason, M. T. Mechanics and planning of manipulator pushing operations. *International Journal of Robotics Research*, 5(3):53–71, 1986.

Mason, M. T. *Mechanics of Robotic Manipulation*. MIT Press, 2001.

McGrew, W. *Chimpanzees material culture: Implications for human evolution*. Cambridge University press, Cambridge, 1992.

McGrew, W. C. Socialization and object manipulation of wild chimpanzees. In Chevalier-Skolnikoff, S. and Poirier, F., editors, *Primate bio-social development: Biological, social, and ecological determinants*, number 261-288. Garland, New York, 1977.

McGrew, W. C. Evolutionary implications of sex differences in chimpanzee predation and tool use. In Hamburg, D. and McCown, E., editors, *The Great Apes*. Benjamin/Cummings, Menlo Park, CA, 1978.

Melzack, R. Phantom limbs. *Scientific American*, (266):120–126, April 1992.

Menzel, E. W., Jr., E. S. S.-R., and Lawson, J. Chimpanzee (pan troglodytes) spatial problem solving with the use of mirrors and televised equivalents of mirrors. *Journal of Comparative Psychology*, 99:211–217, 1985.

Menzel, P. and D'Aluisio, F. *Robosapiens: Evolution of a New Species.* Material World/MIT Press, Cambridge, MA, 2000.

Michael, O. Webots: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42, 2004. (see also http://www.cyberbotics.com/).

Michel, P., Gold, K., and Scassellati, B. Motion-based robotic self-recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, Japan, 2004.

Montgomery, K. Exploratory behavior as a function of "similarity" of stimulus situations. *Journal of Comparative and Physiological Psychology*, 46:129–133, 1953.

Montgomery, K. C. and Segall, M. Discriminative learning based upon the exploratory drive. *Journal of comparative and Psysiological Psychology*, 48:225–228, 1955.

Morasso, P. and Sanguineti, V. Self-organizing body-schema for motor planning. *Journal of Motor Behavior*, 27(1):52–66, 1995.

Mulgaonkar, P., Shapiro, L., and Haralick, R. Matching "sticks, plates and blobs" objects using geometric and relational constraints. *Image and Vision Computing*, 2(2): 85–98, 1984.

Nabeshima, C., Lungarella, M., and Kuniyoshi, Y. Timing-based model of body schema adaptation and its role in perception and tool use: A robot case study. In *Proc. of the 4-th Intl. Conference on Development and Learning (ICDL)*, pages 7– 12, 19-21 July 2005.

Newcombe, N. S. and Huttenlocher, J. *Making Space: The Development of Spatial Representation and Reasoning.* MIT Press, Cambridge, MA, 2000.

Noë, A. *Action in perception.* MIT Press, Cambridge, MA, 2004.

O'Regan, J. K. and Noë, A. A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, 24(5), 2001.

Owings, D., Borchert, M., and Virginia, R. The behavior of california ground squirrels. *Animal Behavior*, (25):221–230, 1977.

Owings, D. and Cross, R. Snake mobbing by california ground squirrels: Adaptive variation and ontogeny. *Behaviour*, (62):50–69, 1977.

Paillard, J. The hand and the tool: The functional architecture of human technical skills. In Berthelet, A. and Chavail, J., editors, *The Use of Tools by Human and Non-human Primates*, pages 36–46. Oxford University Press, New York, 1993.

Parker, C. The use of tools by apes. *Zoonooz*, 41:10–13, 1968.

Parker, S. and Gibson, K. Object manipulation, tool use and sensorimotor intelligence as feeding adaptations in cebus monkeys and great apes. *J. Hum. Evol*, 6:623–641, 1977.

Parker, S., Mitchell, R., and Boccia, M., editors. *Self-Awareness in Animals and Humans: Developmental Perspectives.* Cambridge University Press, Cambridge, 1994.

Parker, S. T. A developmental approach to the origins of self-recognition in great apes. *Human Evolution*, 6:435–449, 1991.

Pearl, J. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, 2000.

Pfeifer, R. and Scheier, C. *Understanding intelligence.* MIT Press, 1999.

Piaget, J. *The origins of intelligence in children.* International Universities Press, New York, 1952.

Pick, A. Störung der orientierung an eigen körper. *Psychol. Forsch*, 1:303–318, 1922.

Plotnik, J., de Waal, F., and Reiss, D. Self-recognition in an Asian elephant. *Proceedings of the National Academy of Sciences of the United States (PNAS)*, 103(45): 17053–17057, November 7 2006.

Poirier, F. E. and McKee, J. K. *Understanding Human Evolution.* Prentice Hall, Upper Saddle River, New Jersey, fourth edition, 1999.

Potts, R. Home bases and early hominids. *American Scientific*, 72:338–347, 1984.

Povinelli, D. J. and Cant, J. G. H. Arboreal clambering and the evolution of self-conception. *Quarterly review of Biology*, 70(Issue 4.):393–421, Dec. 1995.

Povinelli, D. J., Reaux, J. E., Theall, L. A., and Giambrone, S. *Folk Physics for Apes: The Chimpanzee's theory of how the world works.* Oxford University Press, Oxford, 2000.

Power, T. G. *Play and Exploration in Children and Animals.* Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey, 2000.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, Cambridge, New York, second edition edition, 1992.

Ramachandran, V. and Blakeslee, S. *Phantoms in the Brain: Probing the Mysteries of the Human Mind.* William Morrow, New York, 1998.

Ramachandran, V. S. and Rogers-Ramachandran, D. Synaesthesia in phantom limbs induced with mirrors. *Proc. Royal Society London, Series B.*, 263(1369), Apr. 22 1996.

Reed, E. S. Applying the theory of action systems to the study of motor skills. In Maijer, O. and Roth, K., editors, *Complex movement behavior: The motor action controversy*, pages 45–86. Elsevier Sciences Publishers, B. V. North Holland, 1988.

Reiss, D. and Marino, L. Mirror self-recognition in the bottlenose dolphin: A case of cognitive convergence. *Proceedings of the National Academy of Sciences (PNAS)*, 98(10): 5937–5942, May 8 2001.

Riedman, M. L. *Sea Otters.* Monterey Bay Aquarim, Monterey, California, 1990.

Rivlin, E., Dickinson, S. J., and Rosenfeld, A. Recognition by functional parts. *Computer Vision and Image Understanding*, 62(2):164–176, 1995.

Rochat, P. Five levels of self-awareness as they unfold early in life. *Consciousness and Cognition*, 12:717–731, 2003.

Schaal, S. and Atkeson, C. G. Robot juggling: An implementation of memory-based learning. *Control Systems Magazine*, 14:57–71, 1994.

Schilder, P. *Das Köperschema*. Springer Verlag, Berlin, 1923.

Schultz, S. ?? *Princeton Weekly Bulletin*, 90(26), April 30 2001.

Shapiro, L. G. and others. Matching three-dimensional objects using relational paradigm. *Pattern Recognition*, 17:385–405, 1984.

Sharma, R., Hervé, J.-Y., and Cucka, P. Dynamic robot manipulation using visual tracking. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1844–1849, 1992.

Sharma, R. and Sutanto, H. Integrating configuration space and sensor space for vision-based robot motion planning. In Laumond, J.-P. and Overmars, M., editors, *1996 Workshop on the Algorithmic Foundations of Robotics:Algorithms for Robotic Motion and Manipulation*, pages 63–77, Wellesley, Massachusetts, 1996. A. K. Peters.

Shillito, D. J., Gallup, G. G. J., and Beck, B. B. Factors affecting mirror bahaviour in western lowland gorillas, *gorilla gorilla*. *Animal Behaviour*, 57:999–1004, 1999.

Singh, S., Littman, M., Sutton, R., and Stone, P. Learning predictive state representations. *Paper Draft (unpublished)*, 2002.

Skinner, B. F. *The Behavior of Organisms: An Experimental analysis*. D. Appleton Century, New York, 1938.

Smith, R. *The Open Dynamics Engine (ODE) User Guide*, 2003. http://ode.org/ (Date accessed: May 2004).

St. Amant, R. and Wood, A. B. Tool use for autonomous agents. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 184–189, 2005.

Stansfield, S. *Visually-Guided Haptic Object Recognition*. PhD thesis, University of Pennsylvania, Dept. of Computer and Information Science, 1987. Technical Report MS-CIS-87-92.

Stansfield, S. A. Representing generic objects for exploration and identification. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1090–1095, 1988.

Stansfield, S. A. Robotic grasping of unknown objects: A knowledge-based approach. *International Journal of Robotics Research*, 10(4):314– 326, August 1991.

Stark, L. and Bowyer, K. *Generic Object Recognition using Form and Function*, volume 10 of *Series in Machine Perception and Artificial Intelligence*. World Scientific, Singapore, 1996.

Stark, L., Bowyer, K., Woods, K., Hall, L., and Cook, D. Application of machine learning in function-based recognition. In Ikeuchi, K. and Veloso, M., editors, *Symbolic Visual Learning*, pages 141–169. Oxford University Press, New York, 1997.

Stark, L. and Bowyer, K. W. Achieving generalized object recognition through reasoning about association of function to structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(10):1097–1104, October 1991.

Stoytchev, A. Computational model for an extendable robot body schema. Technical Report GIT-CC-03-44, Georgia Institute of Technology, College of Computing, October 21 2003.

Stoytchev, A. Toward learning the binding affordances of objects: A behavior-grounded approach. In *Proceedings of AAAI Symposium on Developmental Robotics*, pages 17–22, Stanford University, Mar 21-23 2005.

Struhsaker, T. T. and Hunkeler, P. Evidence of tool-using by chimpanzees in the ivory coast. *Primates*, 15:212–219, 1971.

Sutton, M., Stark, L., and Bowyer, K. Gruff-3: Generalizing the domain of a function-based recognition system. *Pattern Recognition*, 27(12):1743–1766, December 1994.

Sutton, M., Stark, L., and Bowyer, K. W. We do dishes, but we don't do windows: function-based modeling and recognition of rigid objects. In *SPIE 1825: Intelligent Robots and Computer Vision XI: Algorithms, Techniques and Active Vision*, pages 132–142, Boston, MA, November 1992.

Sutton, M., Stark, L., and Bowyer, K. W. Function-based generic recognition for multiple object categories. In Jain, A. K. and Flynn, P., editors, *Three-dimensional Object Recognition systems*, pages 447–470. Elsevier Science Publishers, 1993.

Sutton, R. Verification. on-line essay (http://www.cs.ualberta.ca/˜sutton/IncIdeas/Verification.html) (Date accessed: September 2006), 2001a.

Sutton, R. Verification, the key to AI. on-line essay (http://www.cs.ualberta.ca/˜sutton/IncIdeas/KeytoAI.html) (Date accessed: September 2006), 2001b.

Tattersall, I., Delson, E., and Couvering, J. V., editors. *Encyclopedia of Human Evolution and Prehistory.* Garland Publishing, New York, 1988.

Tebbich, S., Taborsky, M., Fessl, B., and Blomqvist, D. Do woodpecker finches acquire tool-use by social learning? *Proceedings of the royal society of London Series B - Biological Sciences*, 268(1482):2189–2193, November 2001.

Tiemersma, D. *Body Schema and Body Image: An Interdisciplinary and Philosophical Study.* Swets and Zeitlinger B. V., Amsterdam, 1989.

Tomasello, M. and Call, J. *Primate Cognition.* Oxford University Press, New York, 1997.

Triesman, M. Temporal discrimination and the indifference interval: Implications for a model of the "internal clock". *Psychological Monographs*, 77(13), 1963.

Triesman, M. Noise and weber's law: The discrimination of brightness and other dimensions. *Psychological Review*, 71:314–330, 1964.

Tsai, R. Y. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, Miami Beach, FL, 1986.

Tsai, R. Y. "a versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, August 1987.

Tsukamoto, Y. Pinpointing of an upper limb prosthesis. *Journal of Prosthetics and Orthotics*, 12(1):5–6, 2000.

Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13 (4):376–80, 1991.

van Lawick-Goodall, J. Tool-using in primates and other vertebrates. In Lehrman, D., Hinde, R., and Shaw, E., editors, *Advances in the study of behavior*, volume 3, pages 195–249. Academic Press, New York, 1970.

Varela, F., Thompson, E., and Rosch, E. *The embodies mind: Cognitive Science and Human Experience.* MIT Press, Cambridge, MA, 1991.

Vauclair, J. Phylogenetic approach to object manipulation in human and ape infants. *Human Development*, 27:321–328, 1984.

Viana, L. M. and Jaulent, M. C. Object structure and action requirements: a compatibility model for functional recognition. *International Journal of Intelligent Systems*, (6): 331–336, 1991.

Visalberghi, E. Tool use in a south american monkey species: an overview of the characteristics and limits of tool use in cebus apella. In Berthelet, A. and Chavaillon, J., editors, *The use of Tools by Human and Non-human Primates*, pages 118–131. Clarendon Press, Oxford, 1993.

Visalberghi, E. and Limongelli, L. Lack of comprehension of cause-effect relations in tool-using capuchin monkeys (cebus apella). *Journal of Comparative Psychology*, 108: 15–22, 1994.

Visalberghi, E. and Trinca, L. Tool use in capuchin monkeys: distinguishing between performing and understanding. *Primates*, 30:511–21, 1989.

Wang, X., Merzenich, M., Sameshima, K., and Jenkins, W. Remodelling of hand representation in adult cortex determined by timing of tactile stimulation. *Nature*, 378: 71–75, Nov. 1995.

Washburn, S. L. Tools and human evolution. In Isaac, G. and Leakey, R., editors, *Human Ancestors: Readings from Scientific American.* W. H. Freeman and Co., 1979. (Originally published in Sep. 1960).

WATSON, J. S. Contingency perception in early social development. In FIELD, T. M. and FOX, N. A., editors, *Social Perception in Infants*, pages 157–176. Ablex Pub. Corp., Norwood, N. J., 1985.

WATSON, J. S. Detection of self: The perfect algorithm. In PARKER, S., MITCHELL, R., and BOCCIA, M., editors, *Self-Awareness in Animals and Humans: Developmental Perspectives*, pages 131–148. Cambridge University Press, Cambridge, 1994.

WEIR, A., CHAPPELL, J., and KACELNIK, A. Shaping of hooks in new caledonian crows. *Science*, 297:981, 9 August 2002.

WENAR, C. Executive competence in toddlers: A prospective, observational study. *Genetic Psychology Monographs*, 93:189–285, 1976.

WENG, J., MCCLELLAND, J., PENTLAND, A., SPORNS, O., STOCKMAN, I., SUR, M., and THELEN, E. Autonomous mental development by robots and animals. *Science*, 291 (5504):599–600, Jan. 26 2001.

WESTERGAARD, G. C. and SUOMI, S. J. Mirror inspection varies with age and tool-using ability in tufted capuchin monkeys (cebus apella). *Human Evolution*, 10(3):217–223, 1995.

WHITE, R. W. Motivation reconsidered: The concept of competence. *Psychological Review*, 66:297–333, 1959.

WILLSON, R. *Tsai Camera Calibration Software*, 1995. (this document and the source code can be downloaded from http://www-2.cs.cmu.edu/~rgw/TsaiCode.html) (Date accessed: January 2005).

WILSON, E. *Sociobiology: The new synthesis.* Belknap, Cambridge, MA, 1975.

WOLFF, P. H. The developmental psychologies of jean piaget and psychoanalysis. *Psychological Issues*, 2(1):3–181, 1960. [Monograph 5].

YERKES, R. M. *Chimpanzees: A Laboratory Colony.* Yale University Press, New Haven, 1943.

YERKES, R. M. and YERKES, A. W. *The Great Apes: A study of anthropoid life.* Yale University Press, New Haven, 1929.

YOSHIKAWA, Y., HOSODA, K., and ASADA, M. Cross-anchoring for binding tactile and visual sensations via unique association through self-perception. In *Proceedings of the Third International Conference on Development and Learning (ICDL)*, 2004.

YOSHIKAWA, Y., KAWANISHI, H., ASADA, M., and HOSODA, K. Body scheme acquisition by cross map learning among tactile, image, and proprioceptive spaces. In *Proceedings of the Second International Workshop on Epigenetic Robotics*, 2002.

ZLATEV, J. and BALKENIUS, C. Introduction: Why epigenetic robotics? In *Proceedings of the First International Workshop on Epigenetic Robotics*, September 16-18 2001.

ZUMEL, N. B. and ERDMANN, M. A. Nonprehensile two palm manipulation with non-equilibrium transitions between stable states. In *IEEE International Conference on Robotics and Automation*, pages 3317–3323, 1996.