ABSTRACT

WOOD, ALEXANDER BURCHI. Effective Tool Use in a Habile Agent. (Under the direction of Robert St. Amant.)

Tool use is a hallmark of human intelligence, which has not fully been explored in the artificial intelligence research community. Research in cognitive neuroscience on primates suggests that not only do we maintain a mental representation of our body, but the body schema is modified to include a tool during intentional tool use (Iriki et al., 1996). We have developed a habile (tool-using) agent, based on the Sony Aibo platform, that can pick up a stick and use it as a tool to reach objects previously out of its range. The agent uses a recurrent neural network developed by Steinkühler and Cruse (1998) for maintaining an internal body schema used to find appropriate postures for reaching and grasping tools. We argue that analysis of activities of such tool using agents offers an informative way to evaluate intelligence.
EFFECTIVE TOOL USE IN A HABILE AGENT

by

ALEXANDER BURCHI WOOD

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

COMPUTER SCIENCE

Raleigh

2005

APPROVED BY:

________________________________    _________________________________
Chair of Advisory Committee
BIOGRAPHY

Alexander Wood received his BS in Information Sciences and Technology from The Pennsylvania State University in 2003. During his tenure as an undergraduate, his research was under the direction of Dr. Frank E. Ritter in the Applied Cognitive Sciences Lab. Interests in robotics emerged during an internship at Agent-Oriented Software located in Melbourne, Australia, where he implemented a strategic planner for the University of Melbourne’s Robocup small league team.

After the completion of his undergraduate degree, Alexander pursued a Masters of Science in North Carolina State University’s Department of Computer Science. His research placed an emphasis on Artificial Intelligence, where he completed his studies under the direction of Dr. Robert St. Amant.
ACKNOWLEDGEMENTS

This thesis would not have possible without the professional help and guidance of my advisor, Robert St. Amant; his enthusiasm and insightful feedback gave me the necessary momentum to maintain steady progress throughout my graduate studies. I would also like to thank Thomas Horton, for lending his MMC expertise at times of need, Lloyd Williams for his filming and directing abilities, and Brent Dennis, for the occasional trigonometry diversions.

Special thanks go out to my undergraduate research advisor, Dr. Frank E. Ritter, who introduced me to the cognitive science and artificial intelligence disciplines. His guidance has continued beyond my undergraduate degree, and I am in great appreciation of all of his assistance. I would also like to thank Erica and my family for believing in me and supporting me throughout my matriculation.
# TABLE OF CONTENTS

List of Tables........................................................................................................................................... v

List of Figures ........................................................................................................................................... vi

1. Introduction ........................................................................................................................................ 1

2. Related Works ..................................................................................................................................... 5

   2.1 Research on Animal Tool Use .................................................................................................... 5

   2.2 Body Schema and Cognitive Neuroscience ............................................................................... 8

   2.3 Behavioral Robotics .................................................................................................................. 10

3. Physical Tools .................................................................................................................................... 12

4. Effective Tool Use on a Robotic Platform ....................................................................................... 14

   4.1 The Reach Scenario .................................................................................................................. 15

   4.2 High-level Control Algorithm .................................................................................................. 18

   4.3 Computational Model of the Body Schema ............................................................................. 20

   4.4 Results ....................................................................................................................................... 30

5. Conclusion.......................................................................................................................................... 37

6. Future Work ....................................................................................................................................... 43

7. References ......................................................................................................................................... 45
List of Tables
Table 4.3.1 Angular constraints on the body segments......................................................... 26
Table 4.4.1 Simulated Error in Posture Calculation.............................................................. 35
List of Figures
Figure 1. Objects present in tool use studies performed by Povinelli (2000) ....................... 7
Figure 4.1.1 A typical layout of the Reach Scenario. ............................................................. 16
Figure 4.1.2 Raw and segmented images of the ball............................................................... 18
Figure 4.2.1. High-level state machine. ................................................................................ 19
Figure 4.2.2. Body Model of the Aibo. ................................................................................ 20
Figure 4.3.1. A four segment model of the robot body........................................................... 22
Figure 4.3.2. The nonlinear MMC network schematic. ....................................................... 25
Figure 4.3.3 Linear/Nonlinear Hybrid MMC net in an OpenGL simulation ......................... 28
Figure 4.4.1 Pickup using the MMC net. ............................................................................. 31
Figure 4.4.2 Reach using the MMC net. ............................................................................... 32
Figure 4.4.3 Convergence of Canis’s joint angles during the Reach task............................ 34
1. Introduction

The use of tools is an essential component of intelligent thought. It would be difficult to imagine the quality of human life without our ability to manufacture and use tools. At the start of your day, you rely on a tool to wake you up on time – the alarm clock. You also brush your teeth with a tool, eat your breakfast with a tool, and you may even use a tool to remember what you need to accomplish for the day. If you look around you right now while reading this paper, you will see that you are surrounded by various examples of tools. Lodged in the wall directly in front of me is a thumbtack used to mount my todo list to the wall; a case of a cognitive tool (my todo list) gaining improvement (held at a visually available location) from another tool. The reliance on physical and cognitive tools is common in just about all of our daily activities.

By using tools, humans are able to extend outside their physical and mental limitations hereby expanding their achievement potential. Tool use is of course not limited to humans, but is found in other species in the animal kingdom as well. In the following section, we look into influential studies in animal cognition on tool use.

Tool use has been studied across multiple domains, and for that reason, has been defined at various levels of complexity. The most widely accepted definition of tool use comes from Beck (1980), in the animal cognition discourse:

Thus 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.

However complicated this definition may seem, all of its conditions are required to distinguish tool use from various other non-tool using activities in the animal kingdom. For example, an elephant grasps a branch in its trunk and uses this to scratch its back is a typical
instance of tool use (Baber, 2003), whereas a horse rubbing against a tree does not signify tool use. Here we see the distinction between an unattached environmental object (elephant using the stick) and a stationary object (the tree) that can not be held or carried, but can still facilitate the scratching action.

This study focuses on a simple tool using scenario that uses a robotic dog, the Sony Aibo. Consider a variant of the classic monkey and banana problem: the agent wishes to retrieve a ball. The ball is visible, but located on a shelf, slightly out of the agent’s reach. What are the appropriate means of exploiting the environment surrounding the agent to achieve this goal of obtaining the ball? One solution may involve using the agent’s mouth to grasp a stick and thus extending its reach. An alternative solution may involve a moving a box up to the shelf for use as a step to orient the reach to the ball.

Both of the solutions above require the Aibo to understand the limitations on its body in relation to the goal, and how the internal body schema can be altered to overcome these limitations. In the first case, the solution was to extend the body in order to make the ball a reachable object. The latter solution manipulated the environment in such a way that the Aibo had gained ability to orient its body to bring the ball into reach.

While both solutions are similar on some level of abstraction, in that they both involve manipulating the environment in order to modify their reach potential, they differ in their focus on navigation versus manipulation, respectively, with the latter solution being an unambiguous example of physical tool use. This problem of interest here is not how a plan is formed to achieve a goal but rather, once we see obstacles hindering the achievement of a goal (e.g., ball is out of reach), how do we overcome these obstacles by using objects in the environment. This study commits to the first approach to resolving the Aibo’s reach
limitations by grasping a tool with the agent’s mouth. Though this is a simple example of what we call “effective” tool use, it is illustrative of the requirements on the agent’s design and it is suggestive of more complex tasks that can be carried out in the future.

The actions central to tool use in this scenario, constraint evaluation, tool acquisition, tool application, rely on a computation model of the body schema. The body model uses the Mean Value of Multiple Computations (MMC) network to maintain the representation of the body as a series of four connected segments: the body, neck, head, and optional tool segment (more on this later). The Aibo uses internal simulation on the MMC net to mentally carry out the evaluation of its body constraints in the context of the reach scenario to recognize an opportunity to use a tool. This form of constraint evaluation is similar to how humans perform various forms of evaluation, by mentally visualizing the execution of task.

The MMC net is used for picking up the reaching tool, and using this tool to dislodge the ball. While it is possible to represent a habile agent’s capabilities conventionally (e.g. in-hand(hammer) → ¬grasping-ability ^ striking-ability), the literature indicates that for biological agents, the use of a tool fundamentally alters the agent’s evaluation of information from its sensors and of the capabilities of its actuators. When our agent holds a tool, the tool becomes a functional extension of its grasping manipulator. When the agent reaches for the ball, the workspace expands relative to the size of the tool and consequently the ball becomes reachable.

To maintain a feasible scope for this project, this particular study focuses on a selected set of components necessary for tool use. Our objective was to focus on how the representation of the body can guide constraint evaluation, tool acquisition, and tool application, in the context of the obtain-ball scenario. We do not use the body schema to
control all aspects of the Aibo; actions such as walking and visual searches using head
movements use motion packages provided with our robotic development framework,
Tekkotsu.

There are several components which are important for a tool using architecture,
which we do not emphasize in this study. High level action selection and motion planning,
such as navigation through the environment, is a task left for future development; the current
implementation uses a scripted sequence of actions to complete the obtain-ball task.
Furthermore, all objects and object dimensions in the environment are known \textit{a priori},
leaving tool selection an area of interest in later studies.

The next section provides a survey of research, primarily found in the fields of animal
cognition and cognitive neuroscience that is used to define criteria for a habile agent. Section
3 is dedicated to identifying functional capabilities necessary for structuring general tool use.
Section 4 covers implementation of reach scenario described above. Section 5 concludes
with a discussion on general tests of agent intelligence based on tool using ability.
2. Related Works

The impact of tool use on humans and other species in the animal kingdom has prompted research in several disciplines. This study is primarily influenced by three areas: animal cognition, cognitive neuroscience, and behavioral robotics. While studies in animal cognition have examined the cognitive and behavioral processes associated with tool use, cognitive neuroscience specializes on how the neural structure of the brain is affected during the use of tools. Not only has behavioral robotics assisted our research through the development of architectures and techniques necessary for tool use on the Aibo, but select studies have given us insight towards more general tool using algorithms.

2.1 Research on Animal Tool Use

Historically, there has been a presumed association between tool use and the sophistication of mental processing. The basis for this view was centered on the naïve assumption that human life was dependant on tool use whereas the less sophisticated species simply did not use tools as a result of their lower intelligence. Even Charles Darwin was forced to concede to the Duke of Argyll, who claimed that ‘the fashioning of an implement for a special purpose is absolutely particular to humans’ (as quoted by Wolpert, 2003). Through countless observations we know now that tool use is present at various levels of cognitive complexity.

Tool using behaviors exist in a broad spectrum of animals. The sea otter’s ability to balance an oyster on its chest, and crack the shell open with a rock is a well known example of tool use. Vultures are unable to break open the shells of some of its prey’s eggs, and thus throw stones at the eggs in order to overcome their limitations. Elephants hold a branch in
their trunk while scratching. The following three examples of tool use offer some useful insights we can apply to our habile agent design.

Certainly one of the simplest animals observed to use tools is the wasp. Baber (2003) explains how the mud wasp makes a burrow, lays their eggs, and subsequently packs the burrow with the aide of a pebble held in their mandible. Such activities illustrate that tool use can be present even when agents have limited manipulation, little or no capacity for learning, and simple processing capabilities. This observation offers some insight into general requirements of limited range of tool using behaviors into artificial agents.

Even with more sophisticated animals, we see imperfect strategies and cognitive limitations present during tool use. One compelling study was reported by Visalberghi and Limongelli (1996), in which they compared tool using behaviors among several types of primates: capuchin monkeys (Cebus apella), chimpanzees (Pan troglodytes), bonobos (Pan paniscus), an orangutan (Pongo pygmaeus), and children (Homo sapiens).

One test scenario involved a narrow transparent tube, baited in the center with a piece of food. This test was designed to evaluate the subject’s ability in tool selection, as the subject’s hand was too big to fit through the opening of the tube. Three sets of tools were provided, a bundle of reeds held together with tape, a stick resembling the letter ‘H’ and thus making it too wide to fit through the tube opening, and some additional sticks that were too short to reach the food at tube’s center. A successful strategy involves unraveling the bundle of reeds and using one of the reeds as a tool to push the food out of the tube.

An interesting observations from this particular task occurred with a capuchin subject. The subject in this case, managed to unravel the bundle of reeds, but then subsequently
attempted to push the bait out of the tube apparatus using the tape. This particular behavior shows signs of search strategy driven by poor heuristics.

Povinelli (2000) has carried out a similar study involving chimpanzees (*Pan troglodyte*). The experiments performed here evaluate the chimpanzees’ ability to reason about causal relationships and examine how grasping affordances affect tool use. As before, the experiment involved probing complex tools through an opening to obtain a food reward. The tool used in the experiment, shown in Figure 1, involved an easy to grip handle, small enough to poke through the hole, whereas the other end was too large to fit through the hole and more difficult to grasp.

![Diagram](image)

**Figure 1.** Objects present in tool use studies performed by Povinelli (2000). The object on the left illustrates food reward positioned at an unreachable depth in the tube apparatus. The object on the right depicts the tool of interest used for dislodging the food from the tube.

Solving the problem involves grasping the rod at the more cumbersome end and poking the narrow end of the rod at the fruit through the hole. The results showed that a large percentage of the subjects picked up the rod by the narrow end, rather than the more difficult-to-grasp end, and thus failed the task.

The failures exhibited by subjects in both Povinelli’s (2000) and Visalberghi’s (1996) experiments gives us some insight to some design criteria for our habile agent. The
implications from Povinelli’s complex tools test shows that tool orientation is governed by
grasping affordances rather than the interactions between the tool’s distal end and the food
bait. Section 4 discusses how our agent uses grasping affordances to orient a tool during the
tool acquisition phase.

We also use these finding to guide our design of the information a tool can
encapsulate. An observation from these experiments suggests that tool-using agents are often
limited in their abilities to extract the properties of a tool, and the causal relationships
between the tool and other objects in the experimental environment. Recall that in
Visalberghi (1996), the Capuchin subject attempted to dislodge the food bait using the flimsy
tape. Although the tape was of appropriate length, the tape could not support the action of
dislodging the food due to its pliability. Under this premise, our current implementation only
maintains the dimensions of the tool, disregarding properties such as pliability or malleability.

2.2 Body Schema and Cognitive Neuroscience

Cruse et. al. (1998) makes the claim that the “world we see” is not a result of direct
mappings from sensory input. When we are looking at an object, we extrapolate a stable
image, regardless of the underlying rapid eye movements that are present. Observations such
as this indicate that a mental image is maintained and updated here and there through sensory
input (Simons & Levin, 1997).

Not only does the human brain maintain a mental model of the body’s surroundings,
but it also updates an image of the body as well. The notion of the human brain’s
representation of the body was first hypothesized by Head and Holmes in 1911, who studied
the perceptual mechanisms that humans use to perceive their own bodies (as cited by
Stoychev, 2003). This representation, commonly known throughout cognitive neuroscience
literature as the *body schema*, is a complex sense that combines information from bodily sensations of location and orientation (proprioception), tactile sensation, and visual input.

Indirect evidence of this body schema comes from clinical studies on patients with various neurological disorders or physical disabilities. Limb amputees often experience what is known as *phantom limbs*, the feeling of a nonexistent limb as a result of body schema’s original (complete) representation of the body remaining intact. By contrast, brain damaged patients might suffer from a denial of ownership of a body part suggesting an alteration in the body image.

A study by Graziano et al. (2000) also supports the notion of a body schema. In this study, they found neurons in Area 5 encode the position of the monkey’s arm while it was covered from view. Interestingly, these same neurons reacted to the movements of a visible, realistic false arm, when the real arm was stabilized.

As more recent studies have shown, this body model is not constrained to model our morphological boundaries, but can be extended to incorporate objects attached to our body, such as clothes, ornaments, and tools (Berlucchi & Aglioti 1997; Iriki et. al, 1996). For instance, when one becomes skilled with a hammer, the hammer feels like a part of the body. As far as the neurons are concerned, the hammer might as well be a part of our body. Baber (2003) notes how tools can be “forgotten” in use, suggesting that an experienced user of a tool concentrates on the task at hand and not the tool itself, whereas novice users may instead direct more of their attention on proper use of the tool.

Iriki et al. (1996) monitored the neuronal activity of Japanese Monkeys (*Macaca fuscata*) during a tool use scenario in which the monkeys were required to use a rake tool to reach for food pellets placed at the end of a table. The results of this study show that the
body schema is modified during tool use. One important qualifier in their findings is that their schema is only modified when the monkey intended to use the rake as a tool, not during passive grasping of the tool. In other words, when the rake is held in the hand after the tool using task is complete, the body schema is not expanded to include the tool; whereas during tool use, the body schema is subject to expansion.

Yet another intriguing disability which sheds some light on the body schema can be observed from patients with spatial neglect, a failure to attend to one side of space. As cited by Maravita et al. (2004), Berti and Frassinetti examined the effect of tool-use in a brain-damaged patient, P.P, who suffered from spatial neglect. In the case of P.P., the neglect selectively affected the space close to her body (near space).

When requested to show the midpoint of a drawn line, P.P. put her mark further towards the right from the objective midpoint, as typically observed in neglect. However, when lines were presented out of hand’s reach (far space), P.P’s bisections using a laser pointer were flawless. By contrast, when a long stick was used for the same far-line bisection, P.P. showed a rightward bias again (Maravita et al., 2004).

The conclusions drawn from this study show that when the body schema for the hand is expanded via intentional use of a tool, the far space was made reachable and thus coded by a neural network as near space. We use this concept of coding interactions of the body schema in near space in our habile agent.

2.3 Behavioral Robotics

The most closely related work comes from Stoytchev (2003), who describes an extended robot body schema for controlling robot actions based on self-organizing robot body schemas. This approach to modeling the body schema is comparable to the approach used here in that its foundation originates from cognitive neuroscience. Similar to the approach used in this study, Stoytchev extends the body schema during intentional tool use.
The primary difference between this and our work is that of scope: this study focuses on a more general conception of tool use, which in principle include navigation of the mobile platform as well as tool identification and acquisition.

Metta, Sandini, and Konczak (1999) have developed a learning model for visually-guided reaching using a robot arm. An apparent difference from our work is that the robotic representation does not expand to incorporate a tool, nor is a tool used in their studies. Along the same constraints as Stoytchev, Metta et al uses a robot arm that is fixed in location, and can only move in two dimensions. Their work is found to be very important for future work, as visual guided learning is an important component to learning tool use. The technique used for visually guided learning used in Metta’s research can not be directly applied to our work on the Aibo, as it requires the robot arm to be recognized in the visual sensory system. Since the entire body of the Aibo is treated conceptually as a robot arm with a camera at the end effector, we can not visually perceive the arm in the environment.

More recent research by Metta and Fitzpatrick (2003) extend their visually guided learning via direct manipulation with the environment. Through certain interactive gestures (poking, prodding, tapping, swiping, batting, etc.) using a robot arm, they were able to evoke object identification by tracking displacement of moveable objects upon contact. “Poking allows us to determine the boundary around a mass that moves together when disturbed, which is exactly what we need to know for manipulation” (Metta and Fitzpatrick, 2003). This form of learning through interaction offers value in our future work, in learning causal relationships, object identification, and affordance detection during tool use.
3. Physical Tools

The tool use research surveyed in Section 2 guides us to a broader understanding of tool use. The research of tool use over multiple disciplines lends a variety of perspectives, leading us to the need for a more comprehensive model of tool use, to support experimentation in Artificial Intelligence. In order to encapsulate tool-using behaviors into a habile agent, we must extract some general properties present in physical tools.

In physical environments, tools are persistent artifacts with specialized behaviors and properties that can be used overcome physical limitations of the tool user. The persistence of a tool is important, because they can be replicated, reused, and they can encapsulate information as well as behaviors (St. Amant and Horton, 2002). In the case of reaching for an object with a stick, we can see that in the context of problem, the stick can encapsulate length and whether it can be grasped.

Tools can be classified by their intended application. St. Amant and Horton (2002) have defined four categories that classify tools by their action:

- **Effective** tools produce a persistent effect on materials or the environment, e.g., hammers, screwdrivers, saws.
- **Instruments** provide information about materials or the environment, e.g., thermometers, yardsticks, and magnifying glasses
- **Constraining** tools stabilize materials or the environment. These tools are often used in conjunction with effective tools, such as a vise stabilizing a piece of wood for more efficient sawing.
• *Demarcating* tools impose informational cues on materials or the environment, e.g., a carpenter’s pencil.

It is important to stress that this classification is not static, but can change depending on how one uses it. Tools can be used for purposes not intended by their designers. Consider the example of a yard stick; it is intended for measurement, thus classifying it as an instrument. However, when used to obtain something out of reach, this tool is now an effective tool.

In the yardstick example above, how does one know when to use a tool out of context? One important feature of tools is that they provide rich cues about their applicable use. Using a yardstick to increase the user’s reach is natural, because the stick can be grasped easily, it is lightweight, and it increases the reaching distance of the user.

These observations suggest that tool use is a fascinating area worth exploring from a variety of viewpoints. In our first implementation, we have developed a tool using agent that is capable of applying effective tools to overcome reaching limitations. This particular implementation is described in more detail in the following section.
4. Effective Tool Use on a Robotic Platform

The Aibo robotic platform serves as a good test-bed for habile agent development. Despite its limitations, the Aibo offers the advantages of navigation and manipulation capabilities integrated into a single, widely available platform, moderately priced at ($1,800 USD). For the purposes of this study the Aibo provides quality manipulators, CPU, networking, and sensors.

The most important requirement of our robotic platform is the presence of a robotic manipulator capable of gripping and holding objects. Under this premise, we committed to the Sony Aibo ERS-7 model, as this particular model is equipped with a mouth that can grasp and hold objects of sufficient size and weight.

Sony provides Open-R, a software development kit, enabling low level access to all sensors and manipulators. Functionally, Open-R is very powerful; however at the scope of the project, we required a development framework that provides a higher level of control over the robotic environment. Our development relied on Carnegie Mellon University’s Tekkotsu, an application framework for robotic platforms. Tekkotsu is an event driven, object oriented architecture that uses C++ to interact with various Aibo control objects. The framework contains many useful modules; most notably vision processing, wireless network communications, motion primitives, hierarchical state machine, and various low level behaviors.

The Aibo offers a range of input devices, including a color camera, infrared (IR) distance sensors, touch sensors, and stereo microphones. The color camera is capable of 416(H) x 320 (V) picture elements at frame rates of up to 30 frames per second. A touch sensor is essentially a button located on the bottom of each leg, and offers boolean feedback
as to whether or not the sensor is pressed against a surface. For distance feedback, the Aibo includes three infrared sensors: a near IR sensor for collecting distance readings from 50mm to 500mm, a far IR sensor for distance readings from 200mm to 1500mm, and a chest IR sensor, which points downward 30 degrees from the chest and thus useful for detecting table edges and objects on the ground. The near and far IR sensors are both located directly above the camera, and thus provide the necessary orientation for acquiring distance readings on the objects we see in the camera.

In this study, a basic tool using model was constructed which uses the Sony Aibo in conjunction with the Tekkotsu framework to carry out a goal of reaching a ball. The task is an evident example of effective tool use in that the use of tools is transparent to the tool user. For convenience, I will refer to the system we have developed for the Aibo as Canis Habilis, or simply Canis.

Our objective for this study was to build an agent that is capable of basic tool use in a given context. As our broader research goals are directed towards a more general model of tool using intelligence, we use Nillson’s (1995) terminology to refer to our tool using agents as Habile agents.

4.1 The Reach Scenario
Similar to studies done by Iriki et al. (1996), Povinelli (2000), Visalberghi and Limongelli (1996), and Graziano et al. (2000), the tool-using task used here involves the use of a physical object as an extension of a manipulator. The task Canis faces of obtaining a ball placed out of its reach is analogous to the classic monkey obtaining the banana study found in related neuroscience and anthropology literature. This is one of the simplest tool-
using tasks, but it is illustrative of the requirements on the agent’s design and suggestive of more complex tasks.

The current implementation uses primitive image processing algorithms and requires several visual assumptions about the environment and its available objects. The behavior of Canis as described in the remainder of this section therefore relies on functional placeholders in several areas of visual processing: objects are distinguished by color rather than shape; a long object is grasped at a particular place on its surface identifiable by a visual mark; dimensions of objects are known a priori. The impetus for relaxing these environmental constraints was to overcome various hardware limitations and provide more reliable state perception.

Figure 4.1.1 A typical layout of the Reach Scenario. The ball (pink) is raised out of reach of Canis, whereas the tool (orange) is low enough to grasp, and has an edge exposed.

Figure 4.1.1 shows the standard setup of the Reach Scenario. In this particular setup, the ball is raised roughly one foot off the ground, although the setup does not require any specific height. The orange object off to the left of Canis is the tool. The tool is raised off
the ground to expose a side of the tool for the pickup action. The large orange piece arranged on the front of the cup is necessary for two reasons: (1) It provides a distinct color sample that is easy for the Canis to detect. (2) It is facing Canis, providing better color readings and better infrared distance readings. When the tool lies on the ground without these enhancements, the glare from the fluorescent lights often makes the tool appear to white, thus preventing the tool from being recognized. The blue visual marking on the tool is the affordance. A closer look at the tool shows that the blue affordance is actually L-shaped, once again, for enhancing the color and distance sensory input of Canis.

Each object in the environment has a unique color to relax the complexities involved in object recognition. The framework must load color tables at startup, which is used to determine the classification of a specific pixel. Therefore, a sample of images must be collected of the various objects in the environment that embody the colors we wish to extract from the scene. These images are loaded into VisionTrain, a supplemental tool that comes with the Tekkotsu package, which allows us to assign a range of color intensities to a user-defined color index. The Tekkotsu will throw an event upon the detection of dense region of pixels corresponding to any user-defined color index. When these objects are identified an event is thrown if the object appears in the agent’s field of view. Figure 4.1.2 shows example results of the segmentation process on the pink ball.
The current incarnation of Canis recognizes four distinct colors: pink, blue, orange, and green. Colors orange and green are used to identify tools, whereas pink is used to detect the ball. As mentioned above, the blue color is reserved for affordances, as this is the indicator for the graspable region on the tool.

### 4.2 High-level Control Algorithm

Canis begins its task by visually searching the environment for the ball. Once the ball has been located and Canis finds that it is within reach, Canis pushes the ball with its nose. If the ball is visible but out of reach, Canis searches the environment for a tool, a stick of appropriate length. Canis approaches and picks up the stick with its mouth, and then performs another search for the ball. Once found, Canis moves into a position such that the can be reached with the stick. Canis then dislodges the ball from its position on the shelf.
For this task, the actions carried out by Canis are governed by a state machine, as shown in Figure 4.2.1. In future work this state machine will be replaced by a more general motion planning system. The state machine runs through the sequence of actions as shown to achieve the goal of reaching the ball.

![State Machine Diagram](image)

*Figure 4.2.1. High-level state machine. The state machine controls the sequence of actions the agent must operate over to accomplish the goal of reaching the ball.*

The two actions of importance in this sequence are pickup and reach. Both actions rely on the Mean value of Multiple Computations (MMC) net (discussed in Section 4.3), acting as an internal body schema, to converge to a desired posture. Once a solution is found, Canis carries out the resulting joint configuration. Canis’s body schema consists of four connected segments: body, neck, head, and an optional tool segment. Using this simplified model of the body allows the internal body to be controlled in a fashion similar to a robotic arm. This configuration is shown in Figure 4.2.2.
Actions, such as picking up the tool, offer little room for error in calculating the position of the body in order to successfully complete the goal. In order to produce controlled movements, we must know the joint values to achieve a desired position. The internal kinematics performed on the body model carried out with the MMC net. The computational model for the internal representation of Canis is discussed further in Section 4.3.

### 4.3 Computational Model of the Body Schema

During the intentional use of a tool, the representation of our body expands to incorporate the tool into our body schema (Iriki et al., 1996, Maravita & Iriki, 2004). Under this premise, the habile agent performing in the *obtain-ball* scenario requires an extendable body representation that models an abstraction of the agent’s body with or without the presence of a tool extension. The model used to represent this internal body schema comes from work done by Steinkühler & Cruse (1998) on the MMC net. The MMC net is a model for the control of multi-segmented manipulators with redundant degrees of freedom. An
MMC net offers an adaptive inverse and forward kinematics model, which can be easily extended to integrate additional body segments, without the need of recalculating the network. In this section, an extension on the traditional three-segment robot arm is presented for the inclusion of a held object.

The MMC net (Steinkühler U., & Cruse, H., 1998) is a recurrent neural network that converges to a stable state corresponding to the geometrically correct posture of the robotic body. The calculations involved in the MMC net rely on the notion that several geometric relationships can be used to calculate the same joint value.

The network calculates a complete set of state variables relative to the body model. Our particular network takes as input each vector shown in Figure 4.3.1, in addition to tilt, rotation about the vector’s x-axis, and pan, rotation about the segment’s y-axis, of segments L1, L2, L3, and L4. The final output is obtained by calculations of the mean value of these multiple computations. Subsequent iterations use the previously calculated output as its input each round.
Figure 4.3.1. A four segment model of the robot body. Each vector, L1, L2, L3, and L4, represents the body, neck, head, and tool respectively. The position of the end effector is represented by the vector R. Three additional vectors D1, D2, and D3 act as “muscles” which expand and contract during robot arm movement.

Working under the constraints of the Aibo, we can conceptionally think of the habile agent as a robot arm; the mouth being the grasping manipulator. Figure 4.3.1 shows the body segments oriented similar to the robot arm commonly found in MMC literature.

When a tool is grasped, the body schema for the habile agent must adapt to incorporate the held object during intentional tool use. A fourth segment was added to represent a tool held in the mouth of our Aibo. One could conceivably just extend the end effector of the head to include the length of the tool; however the tool may not be held at the same orientation every time it is picked up. Treating the tool as its own segment attached to the head yields more accurate modeling of the body schema. When a tool is not present, the length of this segment is set to zero, and thus, will not factor into the posture calculation of body schema. With the addition of the fourth segment, we must derive the multiple
equations required for network construction. Using the vectors presented in Figure 4.3.1, we can derive 15 equations that hold true:

\[
\begin{align*}
L_1 + L_2 + L_3 + L_4 - R &= 0 \\
L_1 + L_2 - D_1 &= 0 \\
L_2 + L_3 - D_2 &= 0 \\
L_3 + L_4 - D_3 &= 0 \\
L_1 + L_2 + D_3 - R &= 0 \\
D_1 + L_3 + L_4 - R &= 0 \\
L_1 + D_2 + L_4 - R &= 0 \\
D_1 + D_3 - R &= 0 \\
D_1 + D_2 + L_4 - L_2 - R &= 0 \\
L_1 + D_2 - L_3 - D_1 &= 0 \\
L_2 + D_3 - L_4 - D_2 &= 0 \\
D_1 - L_2 + D_2 - L_3 + D_3 - R &= 0 \\
D_1 + D_3 - L_4 - L_3 - L_2 - L_1 &= 0 \\
D_1 + D_3 - L_4 - D_2 - L_1 &= 0
\end{align*}
\]

Notice that each segment appears eight times in the 15 equations. We can independently solve for each vector, yielding the mean equation for each segment. This mean value of each vector can be computed by summing the eight equations containing the segment of interest and dividing by eight.

\[
L_{1m} = \frac{1}{8} \left( -L_2 - L_3 - L_4 + R - L_2 + D_1 - L_2 + D_3 + R - L_4 - D_2 + R + L_3 - D_2 - D_3 + R - L_3 + D_1 - D_2 + D_3 - L_2 - L_3 - L_4 + D_1 + D_3 - L_4 + D_1 - D_2 + D_3 \right) \\
\frac{1}{8}(4L_2 - 4D_2 + 4D_1 + 4D_3 - 4R)
\]

The remaining eight equations for the network are shown below. Each equation is calculated in a similar fashion to the calculation of \(L_{1m}\) above. The resulting mean value equations are the heart of the network. These equations are shown below in Figure 4.3.2 as the vertical lines in the network; each connection on an equation’s line represents the positive
(closed circle) or negative (open circle) inclusion of that segment into the mean value calculation.

\[
\begin{align*}
L_{1m} &= \frac{1}{8} \left( -4L_2 - 4L_4 + 4D_1 - 4D_2 + 4R \right) \\
L_{2m} &= \frac{1}{8} \left( -4L_1 - 4L_3 + 4D_1 + 4D_2 \right) \\
L_{3m} &= \frac{1}{8} \left( -4L_2 - 4L_4 + 4D_2 + 4D_3 \right) \\
L_{4m} &= \frac{1}{8} \left( -4L_1 - 4L_3 - 4D_2 + 4D_3 + 4R \right) \\
D_{1m} &= \frac{1}{8} \left( 4L_1 + 4L_2 - 4D_3 + 4R \right) \\
D_{2m} &= \frac{1}{8} \left( -4L_1 + 4L_2 + 4L_3 - 4L_4 + 4R \right) \\
D_{3m} &= \frac{1}{8} \left( 4L_3 + 4L_4 - 4D_1 + 4R \right) \\
R_{m} &= \frac{1}{8} \left( -4L_1 - 4L_4 - 4D_1 - 4D_2 - 4D_3 \right)
\end{align*}
\]

The MMC net comes in two classes, linear and nonlinear, and is scales effortlessly from 2D to 3D representation. The linear model requires fewer calculations and is easier to implement, but it fails to maintain segment lengths. In the linear model, the segments stretch as they converge to a solution. Conversely, the nonlinear net holds the segment lengths constant, and thus serves as a better model for real-world applications, including the control of Canis.
Figure 4.3.2. The nonlinear MMC network schematic. This is similar to the nonlinear network shown in Steinkühler U., & Cruse, H. (1998), but extended for a four segment representation. The connection weights are either 0.5 (closed circles), -0.5 (open circles), or zero (no circle). The square connections represent the dampening constant. The nonlinear subsystems are used for conversions between segment coordinate systems.

In the nonlinear model, the network takes the segment lengths (not the segments themselves) and their rotational components as inputs into the network. Additional subsystems are required for transforming into each segment’s coordinate system. These are shown schematically in Figure 4.3.2 as boxes P, T, Q.

There is a flexible mapping between the output of the nonlinear network and the robot platform. When the network relaxes to a posture, the Aibo’s joint configuration is modified by the resulting network output angles. In order to prevent conflicting or illegal joint configurations, the angular output of the network must be constrained each iteration. This step is shown in Figure 4.3.2 by the box labeled C.
The constraining of the angular output of the network ensures that the joint angles do not exceed their mechanical bounds. Due to the network’s adaptive nature, by constraining the joints after each iteration, the network will still converge to a geometrically correct posture, provided that one exists. In the current model, all changes to the angles $a_4$ and $b_4$ (tilt and pan angles for the tool segment) in Figure 4.3.2 are ignored, because Canis is unable to alter the tool’s orientation when held in the mouth. With this restriction in place, the other body segments will compensate for the constraint, and perform the necessary adjustments to attain the final joint configuration. The constraints in the current model are summarized in Table 4.3.1.

Table 4.3.1 Angular constraints on the body segments. N/A indicates that the joint may not be altered by the network.

<table>
<thead>
<tr>
<th>Segment</th>
<th>Identifier</th>
<th>Angular range (degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body Pan</td>
<td>A1</td>
<td>N/A</td>
</tr>
<tr>
<td>Body Tilt</td>
<td>B1</td>
<td>N/A</td>
</tr>
<tr>
<td>Neck Pan</td>
<td>A2</td>
<td>N/A</td>
</tr>
<tr>
<td>Neck Tilt</td>
<td>B2</td>
<td>-90 – 0</td>
</tr>
<tr>
<td>Head Pan</td>
<td>A3</td>
<td>-80 – 80</td>
</tr>
<tr>
<td>Head Tilt</td>
<td>B3</td>
<td>-15 – 45</td>
</tr>
</tbody>
</table>

The system is versatile, even when there is no geometrically plausible solution available. In cases where the ball is out of reach (with or without the presence of a tool), the segments converge to a posture that minimizes the distance of the end effector and the ball. Thus, the body model will still converge to an approximate solution.

One drawback to the nonlinear MMC network is that it must remain at a fixed location. It would be useful to use the body schema for testing if an object is reachable. In earlier implementations of Canis, the system took a more math intensive approach to determining the height of the ball. The previous method involved trigonometry calculations over the current distance readings and joint configuration to determine if the ball was
reachable. To support a body centered approach to evaluating the capabilities of the agent and its ability to reach the ball, we used a modified version of the MMC net that combines both linear and non-linear approaches.

Recall that in a linear MMC network, segment lengths are not preserved; as the network converges, the segments stretch to reach the target location. The idea of body centered reasoning for Canis involves translating the body schema in a two dimensional plane, and performing the reach task in the world it knows internally. Under this notion, the application of the linear network becomes clear – we can modify one of the segments into a linear component to allow for translation. As the network converges, this translational segment stretches to the target location. The effects of the linear-nonlinear hybrid can be seen in Figure 4.3.3. This representation allows Canis to “imagine” itself doing the task in order to properly evaluate if an object is reachable.
Figure 4.3.3 shows the effects of the linear/nonlinear MMC net in an OpenGL simulation environment. The figure shows the initial and final postures, calculated by the MMC network. In the reachable scenario, Figure 4.3.3 (a) and (b), the distance of the end effector and the target location, shown as a pink sphere, converges to zero. In Figure 4.3.3 (c) and (d), the joint constraints prevent the body schema from converging to the target location.

The application of the linear/nonlinear MMC network is a more biologically plausible assessment of whether physical actions are possible. Svennson and Ziemke (2004) argue that “many, if not all, higher-level cognitive processes are body-based in the sense that they make use of (partial) simulations or emulations of sensorimotor processes through the reactivation of neural circuitry that is also active in bodily perception and action.” Aside from the absence of sensory input or action execution, many argue that internal simulation of an action
is the same as overt actions (Hesslow, 2002; Jeannerod, 2001 as cited by Svennson & Ziemke 2004). Our system operates under this same notion, by using the modified MMC network we emulate the actual reach action internally to determine if a tool is needed to reach the ball.

The creation of a linear/nonlinear MMC network is quite simple. All that is required is the removal of the nonlinear subsystems (Q and P shown in Figure 4.3.2) used for L1. The up vector component must also be constrained to a constant value to maintain translation in a two dimensional plane.

The application of the linear/nonlinear hybrid is currently only used for determining if an object is reachable. The previous nonlinear MMC network is still used for pickup and reaching actions in its pure form, with all nonlinear subsystems intact. By using the same neural network control structures for both overt actions (pickup and reach) and action emulation, our habile agent uses the same mechanisms for actions and perceptions, as it uses for some of its cognitive tasks.

The MMC network serves as a good model for biological agents, as it is an approximate model of what we see in real world kinematics. The results of the MMC net are comparable to the results of Iriki et al. (1996), regarding the body schema adaptation to held tools; however we do not claim that MMC network is a model how biological brains maintain the body schema.

Direct application of the MMC network to robot manipulators differs from that of biological analogues. The end effector paths are different from paths of that yielded by human arm movements; they do not show straight paths in the workspace, nor do they exhibit a bell-shaped velocity profile (Steinkühler & Cruse, 1998). Additionally, our particular
MMC representation can only yield a three or four segment body schema representation, to reflect whether or not a tool is held. The network can not be extended to incorporate additional extensions.

There are properties of the body schema which the MMC net models quite nicely. Specifically, when picking up and using a tool, such as a rod to extend one’s reach, it is computationally inexpensive in the sense that it requires very little time to adapt to the use of a tool that amplifies an existing capability. Additionally, when one uses a tool such as a rod, it feels as if it is an extension of one’s arm; one carries out reaching tasks as if one simply had a longer arm. These conceptual properties of the body schema translate directly into the MMC net making it a plausible solution for our studies.

4.4 Results

The use of the MMC net has been successful in both tool acquisition and ball dislodgment. The current implementation does not incorporate a segment for the mouth of the agent (used to hold a tool). Therefore, the MMC net is used to position the agent’s mouth up to the affordance, and subsequently run through a scripted motion sequence to complete the grasping of the tool.
Figure 4.4.1 Pickup using the MMC net. Before (a) and after (b) the internal body model determines the appropriate posture for reaching.

The MMC net converges to a geometrically correct posture similar to that illustrated in Figure 4.4.1. This process requires roughly 60 iterations for the network to relax to a stable posture. Once this proper posture is achieved, the Aibo runs through a motion sequence that completes the pickup process by leaning in and closing its mouth.

The use of a motion sequence is necessary for overcoming sensory limitations in the IR distance sensors. With the closest distance readable being 50mm, we must stop the approach to the tool affordance when we are roughly 50mm away. For this reason, the most effective and consistent means of grasping a tool is to position the mouth directly in front of the tool, as shown in Figure 4.4.1.

The reach action is also carried out by internally computing the correct posture with the MMC net, using a variable tool length. Taking advantage of MMC net’s flexibility, the tool length could be changed at any given moment without the need for recalculating the network. A tool length of zero was used to indicate that no tool was present.

Similar to the pickup action, the MMC net’s joint configuration must be applied to Canis a small distance away from the actual location of the ball. Otherwise, the stand
holding the ball will be knocked down while Canis is approaching a reachable location. The solution to this, is to move the internal target location to be a few millimeters in front of the actual ball location, and reach at that point. Upon achieving the proper posture determined by the MMC net, a motion sequence is used to lean Canis into the actual location of the ball. The outcome of the MMC calculated joint configurations have shown to successfully dislodge the ball.

The MMC network takes roughly 100 iterations to stabilize to a desired posture. Although the four segment model is a better representation of the robot body, it requires a slightly more iterations in order to converge to the proper posture in comparison to a three segment representation. To compare convergence times of the three segment and four segment representations, we removed all angular constraints, and placed the ball at a reachable location in simulation. By extending the third segment in the smaller model to include the length of the tool, the representations are identical in length. Under these circumstances, the convergence of the three segment model to the desired state of a reachable location is roughly 50 iterations; half of what is required for the four segment model. We
hypothesize that the faster convergence can be attributed to fewer segment conflicts. As we decrease the number of segments in the model, our solution space becomes smaller, and thus we can reach a desired state faster.

The range of motion for the neck tilt is from 0 (neck is pointing upwards similar to Figure 4.4.2) to -90 degrees (neck is pointing outwards from the torso). The head nod can look up 45 degrees, or look down 15 degrees, while the pan can look left and right at +/- 90 degrees. As mentioned in Section 4.3, the MMC network must ensure that joints do not exceed their bounds. Figure 4.4.3 shows the convergence of these three angles during a reaching process similar to what is shown in Figure 4.4.2.
Figure 4.4.3 Convergence of Canis’s joint angles during the Reach task. Near iteration 10, the head tilt (b3) hits its maximum value, 45 degrees. In response to b3 reaching its boundary, the neck tilt (b2) changes its angle trajectory.

The plots in Figure 4.4.3 show the versatility of the network under angle constraints. As shown in Table 4.3.1, the maximum angle possible for Canis’s head nod, is 45 degrees. During convergence of the head nod (shown as yellow plots in Figure 4.4.3), the angle reaches 45 degrees around iteration 10 and remains at this point for the remainder of the reaching action. Corresponding to the head tilt reaching its maximum value, shortly after iteration 10 the neck tilt accommodates to the head nod constraint and changes its angular direction – tilting the neck upwards toward the ball.
The camera is located at the tip of the head. Under the robot arm abstraction, this places the camera at the end effector. Consequently, we must first perceive the object of interest, perform MMC calculations internally, and then apply the resulting posture to Canis. If we were to instead apply the angular calculations after each iteration, we run into the possibility of losing the relative target location due to camera displacement.

Although the MMC network is very robust, error can arise from hardware inaccuracies or flawed perception. On the internal model, we apply the MMC calculated configuration after once the network stabilizes to the calculated posture. The error on the internal MMC calculations is measured by the distance between the perceived target location and the end effector. In simulation, the average error over 20 random (reachable) target locations yielded by the MMC network is 1.0572mm.

To simulate error that may arise through hardware or perceptual limitations, we used Monte Carlo sampling of a Gaussian Distribution to add random error to each joint. In this process, we sampled 500 error values for a standard deviation of 1, 5, and 10 degrees and applied the error to each joint in isolation. The overall error of the MMC posture, with the addition of random posture errors, was calculated by measuring the distance between the end effector and the target location. The average error over 20 random (reachable) points across the three values of standard deviation is summarized in Table 4.4.1.

<table>
<thead>
<tr>
<th>Standard Deviation (in degrees)</th>
<th>Average Error (in millimeters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.0355</td>
</tr>
<tr>
<td>5</td>
<td>12.705</td>
</tr>
<tr>
<td>10</td>
<td>19.5981</td>
</tr>
</tbody>
</table>
It has been observed that on occasion, there is noticeable imprecision when transforming the MMC calculations from internal simulation to the physical joints. In practice, the primary cause of posture calculation errors result from inaccurate distance reading from the infrared sensors. When operating in a complex environment, there are many objects that may interfere with the infrared readings. For example, when trying to obtain a distance reading of the ball, it may in fact be reading the distance of the stand holding the ball. When the agent perceives the ball at a distance that is inaccurate, the target location may be miscalculated, thus resulting in an incorrect posture.

Another possible source of error may arise from imprecise translation from software joint values to the hardware joint settings, possibly arising from resistant joints or latency in joint updates. In any case, we can see from Table 4.4.1 that even slight changes in joint angles can result in significant displacement of the habile agent posture.
5. Conclusion

From a technical perspective, this study has expanded on kinematics research related to advances in MMC application and derivation. Deviating from the traditional three segment body model common the MMC net literature, we derived a four segment model in order to cast a better representation of Canis’s body structure. Furthermore, a new form of MMC networks is presented, the linear/nonlinear hybrid, which can be used to support body centered reasoning in the environment.

Conceptually, this study supports the basic idea that we can develop tool-using agents based on body schemas that are built into a robotic architecture. We have successfully constructed a habile agent to operate in an obtain ball context. The impetus for this study was to establish a model, which can evolve across implementations into a more general tool using agent.

The motivation for this work in animal cognition and cognitive neuroscience touches on research dating back to almost a century ago. Observations and experiments have shown that tool use can be found in a diverse spectrum of creatures in the animal kingdom, ranging from the hard-wired tool-using of insects, to elaborative, creative, and sometimes collaborative tool use of humans. Considering the wide range of tool using capabilities, it is evident that tool use is not architecture specific. From studies surveyed in Section 2, we can build a more generalized view of tool use by characterizing habile agents in terms of their capabilities and constraints on their behavior. Below we have listed some criteria that outline habile agent behaviors.

*The actions of habile agents are constrained by the relationships (geometrical, kinematic, dynamic, etc.) between their physical architecture and the tools they use.* In the
standard monkey-banana test scenario, the monkey must overcome some obstacle in order to obtain a food reward. The related studies covered in Section 2, require the primate subjects to use a tool to dislodge food from confined locations. The constrained relationship presented is that the food reward was either out of reach (Iriki, 1996), or their hands were too big to fit through the tube (Povinelli, 2000, Visalberghi, and Limogelli, 1996).

The morphological structure of the tool-user will constrain the actions available during tool use. The configuration of the grasp directly affects which tools an agent can hold. In the case of Canis, the only grasping manipulator available is the mouth. Consequently, the location of the camera and the infrared distance sensors limit the type of tools Canis can use effectively; the tools should not obstruct the vision and distance sensors when held in the mouth. The sticks Canis uses as tools in recent experiments do not disrupt the necessary sensory readings. The visual system is partially blocked by the tool, but the obscured view is not enough to disrupt tool use. Also, being a quadruped, an opportune time to use tools is when objects are too high. Since Canis is unable to stand up on two legs, a long stick is necessary to overcome the height disadvantage, and indeed, even on two legs you would like to reach higher.

During tool use, habile agents apply strategies that range from scripted sequences to search-based problem solving. Multiple observations and studies in animal cognition have supported these varied levels of sophistication concerning action selection. Richard Byrne (2004) hypothesized that non-primate tool use functions are generally limited in a highly specific way. The distinction Byrne suggests between primate tool use and tool use found in simpler organisms is that “tool use in primates is learnt from experience, while tool use in other animals is innate, coded on species genes” (page 32). The tool using behaviors defined
in Canis’s current implementation relies on a high-level state machine for control over the tool using behavior. The scripted behavior used in Canis is along the similar levels of complexity as the tool-using found in insects, but serves as a good illustration of tool-using capabilities possible on the Aibo.

An internal representation of the agent’s unaided action capabilities adapts (in some cases in a hard-coded fashion, in others more flexibly) to the agent’s acquisition and application of a tool. The internal representation was bolstered by research from Iriki et al. (1996), where the brain activity of macaques was monitored during tool use. Their findings support the notion of an extendable representation of the body, which encodes tools into our body schema during tool use. Our use of the MMC net supports adaptation to the acquisition and application of a tool with appropriate properties. Taking advantage of the MMC net’s flexibility, the tool length could be changed at any given moment without the need for recalculation the network.

Much of our research has been guided by research across various disciplines studying tool use. Just as the underlying principles we surveyed in Section 2 aided in our design and implementation of Canis, the evaluation methods used by animal cognition researchers provide strong guidance for evaluating habile agent systems. Therefore, the evaluation methods used in animal cognition laboratory studies should serve as a model for habile agent assessment over the wide range of tool-using capabilities.

This line of thought guides us into the development of an evaluation framework for tool using intelligence. Ron Endicott, one of our informal collaborators, has suggested that the tool use provides an opportunity for evaluating the general intelligence of artificial agents. The foundation of this test is an analogy to Turing’s imitation game that Endicott calls the
“tooling test.” Each test passed in the tooling test indicates a greater degree of intelligence in the habile agent subject. The test involves the use of physical tools directed towards the accomplishment of a specific task. We can imagine a sequence of increasingly challenging tasks, in which various obstacles are set in a way that can overcome tasks.

*Transparent Tool Use.* The task used in this study is an example of transparent tool use; the tool-using behavior of Canis is directly observable to the tool user. The studies surveyed in Section 2.1 are also examples of transparent tool use. This particular type of test has helped researchers to better understand the extent to which animals can reason about spatial relationships, kinematics and dynamics, understanding casual relationships, and so forth, in comparison with humans.

*Tool Construction and modification.* Although the process of tool construction is present across the animal kingdom, it is not common among simple tool using animals. New Caledonian crows (*Corvus moneduloides*) have been known to make and use complex tools (Hunt, 2000). In captive, these crows were observed bending pliant wires into hooks that, when held in their beaks could pull up on a bucket containing food from a vertical tube (Weir et al, 2002). Canis does not make it this far in the tooling tests, but it remains a task for future work.

*Non-transparent tool use.* Sometimes the tool-user does not have the ability to observe the tool using operation or its effect. For example, consider the task of picking a lock, where a correct procedure may be found over an extended period
of time through a blind search. What we gain from posing such problems is the opportunity to observe the agent’s exploratory strategies.

*Learning tool use by observation.* An interesting question is whether the agent can extract useful tool using information from seeing an object manipulated by someone else. Research in cognitive neuroscience suggests that primates have special circuitry in our brain that affects how we see others, called mirror neurons. In primates, it has been found that these neurons fire during the execution of specific tasks. Interestingly enough, these mirror neurons also fire when primates observe the same task being executed by another individual (Gallese and Goldman, 1998). Passing the test of learning by observation will require habile agents to extract casual relationships and other useful tool-using observations from other tool-using entities.

*Collaborative tool use.* Equally important for the evaluation of intelligence is the ability to communicate. This particular test involves a collaborator, which is unable to complete a task due to the lack of some physical tool or aide. The habile agent should be capable of recognizing the actions and intentions of the collaborator and contributing appropriately to attaining the goal.

The tooling test is not an intelligent/non-intelligent, binary evaluation of tool use. Instead these tests follow the same principles that are followed in animal cognition research; we can say that a tool-using agent is more or less intelligent, based on the sophistication of
their actions. Furthermore, the tooling tests are effective in that they have explicit success criterion: was the task achieved?

In this study, we have taken a task developed for animal cognition, and adapted it to robotics research. The evaluation framework for this experiment also borrows from animal cognition research, by using a success criterion in conjunction with observing behaviors during the execution of a task. Following this methodology, we have developed a foundation for further development of habile agents, with the overall goal of achieving habile agents with more general tool using capabilities.
6. Future Work

The current implementation of Canis illustrates a basic model of transparent tool use. To control the scope of this initial implementation, several environmental assumptions were required, which we plan to phase out over future implementations. The immediate changes discussed in this section address refinements towards the transparent tool use model.

The vision processing Canis uses to extract information out of the environment has several weaknesses. Most important, the images are segmented into only four separate color layers: pink, blue, green, and orange. The system must be very strict when classifying the pixels into these four color channels, in order to reduce the amount of noise detected in the environment. This process of segmentation requires all objects to be one of the four colors for it to be recognized. Therefore, the process of segmenting the visual input for Canis is not very adaptable in regards to object identification.

Additionally, the tool lengths are known a priori – Canis knows beforehand that the orange tool extends 130mm from the mouth when held, or that the green tool extends 50mm from the mouth. To escape the context specific tool identification the system faces now, we plan to integrate existing symmetry analysis algorithms we have developed into the Aibo vision processing system.

Recognizing when and how a tool can be applied is an essential requirement for general tool use. The concept of symmetry is valuable for analyzing potential interactions between the agent and the various objects in the environment. Symmetrical relationships are particularly visible in common physical tools: screw driver head shares a symmetrical relationship with the head of the screw, the handle of a hammer shares symmetrical relationship with a hand, etc. By recognizing symmetric relationships in the environment,
habile agents will be able to search for and select the appropriate tool for the given problem domain.

Aside from various object recognition drawbacks, the state machine control does not offer adaptable behaviors. A state machine control structure restricts the agent to context specific tool using action sequences. This scripted control is present in some tool using animals. For instance, although the wasp grasps a pebble in its mandible for burrow construction, it is unable to use the pebble in other applicable situations that would benefit from tool use. A more general motion planning system is required to control the high level navigation of the habile agent, in order to establish more generalized tool using behaviors.
7. References


