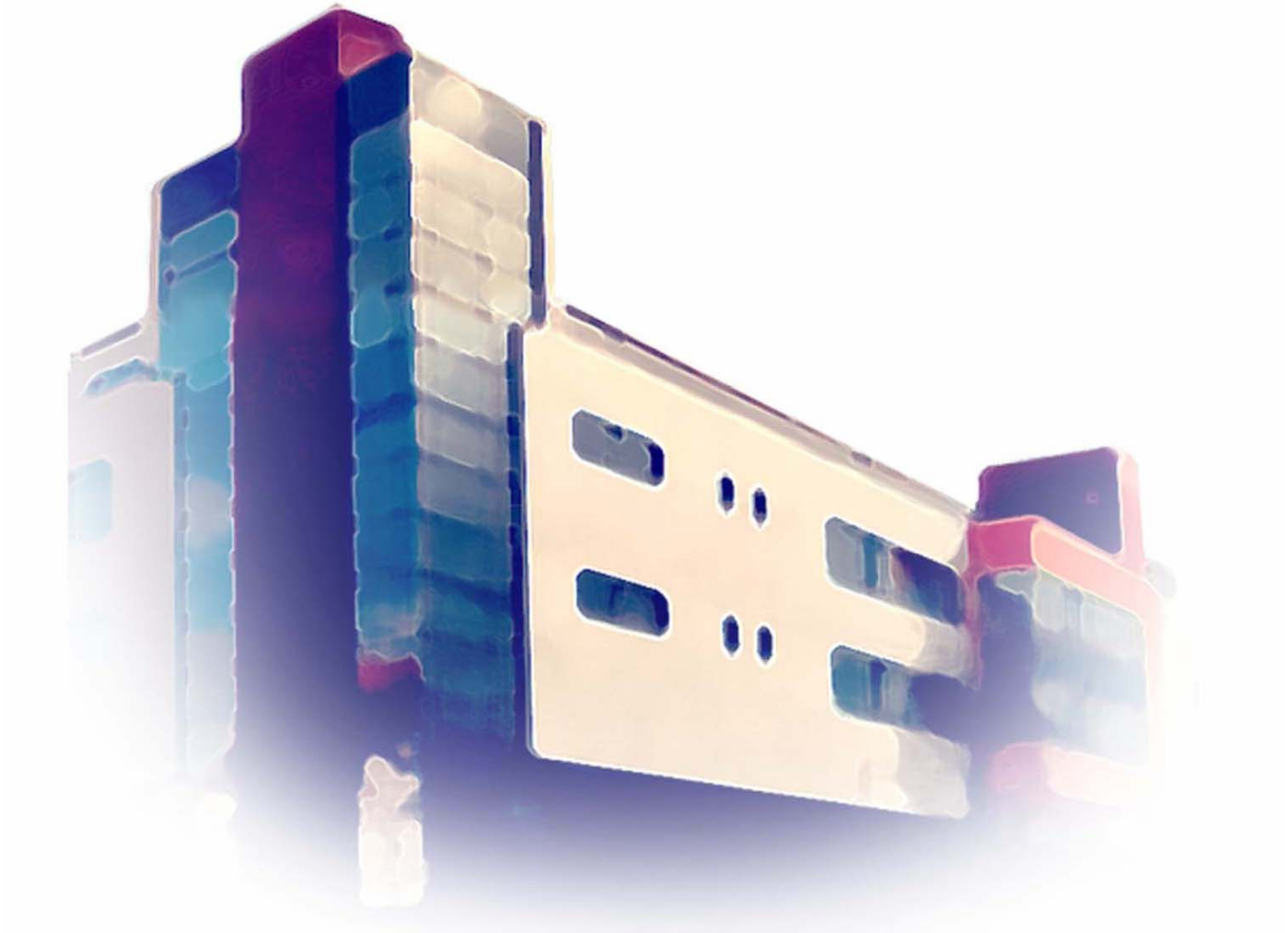
Christopher Lörken

Introducing Affordances into Robot Task Execution



PICS

Publications of the Institute of Cognitive Science

Volume 2-2007

ISSN: 1610-5389

Series title: PICS
Publications of the Institute of Cognitive Science

Volume: 2-2007

Place of publication: Osnabrück, Germany

Date: May 2007

Editors: Kai-Uwe Kühnberger
Peter König
Petra Ludewig

Cover design: Thorsten Hinrichs

UNIVERSITÄT OSNABRÜCK

Introducing Affordances into Robot Task Execution*

Christopher Lörken
Institute of Cognitive Science
University of Osnabrück†
christopher@loerken.net

November 29, 2006

Supervisors:
Dr. Hartmut Surmann
(Fraunhofer Institute for Intelligent Analysis and Information Systems)
Prof. Dr. Joachim Hertzberg
(University of Osnabrück)

*This research was funded by the European Commission's 6th Framework Programme IST Project MACS under contract/grant number FP6-004381. The Commission's support is gratefully acknowledged.
†The work has been conducted at the Fraunhofer IAIS.

Abstract

The concept of affordances as a function-centered view on perception might show to have great potential when being adapted in the field of autonomous robotic systems. Being embedded in the MACS project that aims at evaluating this potential and at proving the usefulness and applicability of affordances, this thesis will tackle the part of developing a robot task execution component as the component of the MACS hybrid robot control architecture that accesses the robot's actuators.

Following the principle of not only using affordances but basing the whole system on that concept, this thesis will provide a clear definition and interpretation of affordances as a relation between environmental features and the capabilities of an agent for action. Based on this definition a task execution component will be designed that defines the three different concepts of behaviors, actions, and tasks. The rationale behind these concepts is to provide with the concept of actions a clear-cut response point to the applied affordance-representation in order to ease its usage and to integrate affordances as a first-class citizen in the component.

An actual task execution component based on the defined concepts will be implemented and its applicability will be demonstrated.

Contents

1 Introduction

This chapter will provide an overview of the motivation and scientific contribution of this thesis in its first half and will conclude with an outline of the thesis's structure in the second section.

1.1 Motivation

In ecological psychology, J.J. Gibson (1979) has proposed the highly debated theory of affordances that can, put shortly, be interpreted as a function-centered view on perception. According to this theory, an agent, or porting the theory to robotics, a robot is assumed to perceive what it can do with a certain object; e.g. that it can lift a stone or throw it and that it can open a door by pushing it.

Perceiving the functionalities of objects based on their features instead of having to label, categorize, and recognize them in order to determine the one action that is knowingly connected to an object shows a great potential of this approach to nowadays robotic systems. While classical categorization-based approaches might be able to determine that a cup might be used to pour coffee into, they normally fail when the same system is being asked if the cup may as well serve as an appropriate paper weight. Extending these classical approaches by introducing affordances may very likely influence the overall system's robustness, flexibility, and power decisively.

While this thesis is not at all targeted at developing a whole robotic system trying to exploit affordance theory for robotics, it is instead part of a large-scale project that actually aims for this goal. In the context of the MACS project (see section 3.2), a whole hybrid robot control architecture is being developed that is based on using the concept of affordances as a first-class citizen and its underlying principle.

Being embedded in this project, the task faced in this thesis is to develop the architectural layer of a *task execution component*; the very component that actually realizes the higher-level targets specified in the system by accessing the robot's actuators.

The scientific contribution and core concept of this thesis is hereby, to define the theoretical background and the concrete design of such a task execution system that is, in accordance with the principles of the MACS approach, based on Gibson's affordance theory.

Since the layer of robot task execution operates on a representational rather low architectural level, the use of designing this component based on affordance theory aims more at supporting and facilitating the usage of affordances in the superordinate architectural components. Therefore, a definition and concept will be developed that aims at anchoring and grounding the concept of affordances on the task execution layer.

The component will therefore be defined in three separate layers that take care of performing basic navigational capabilities, purposeful and goal-oriented actions, and the execution of small tasks as they are appropriate for this component.

It will be elaborated that the task execution component's middle layer for performing purposeful actions defines the actual affordance relation, its executional anchor and response point.

The resulting system is due to its explicit and expressive structure regarded as being highly suitable to be used in the context of demonstrating the usage of affordances, as aimed at in the MACS project, but might as well server as the underlying execution system for other affordance-based architectures.

1.2 Structure Outline

This thesis is structured as follows:

Chapter 2 - Affordance Theory. As this whole thesis is based on introducing the concept of affordances to a robot task execution system, the chapter will elaborate the underlying definition of an affordance, highlighting as well the influence of multiple different fields of research and trying to straighten out some misunderstandings connected to the interpretation of affordances.

Chapter 3 - Using Affordances in Robotics. The first part of this chapter will outline the possible impact of affordance theory in robotics introducing as well some approaches that have already tried to use the concept in their systems. Hereby, it will describe the scientific background that lead to the MACS project that this thesis is embedded in will thus motivate this thesis. The second part will introduce the framework of the MACS project.

Chapter 4 - Concept and Design of an Affordance-based Task Execution Component. This chapter forms the core of the thesis as it describes the requirements of an affordance-based task execution component and elaborates according to these requirements the necessary concepts for its structure as well as, based on these concepts, the actual design that is applied in the context of the MACS project.

Chapter 5 - Implementation and Demonstration. This chapter will describe the implementation of the task execution component as it has been defined in chapter 4. It will furthermore demonstrate the functionality of the different components.

Chapter 6 - Summary and Outlook. This final chapter will provide a comprehensible summary of the whole thesis and will shortly introduce some interesting extensions that might extend the system in future work.

Appendix A - Controller of the Approach Action. This appendix explains the general algorithms used for the approach action that has already been defined prior to this thesis.

2 Affordance Theory

One of the fundamental question of psychological research nowadays and during the past century is how an animal's perception can inform the animal of the meanings and interaction possibilities of environmental objects. How does a cat know that it can jump on a table and how does a person know that it can open a door?

In the field of psychology two main opinions evolved. The classical one assumes, that objects that can be found in the environment have no inherent meaning. They are perceived and an animal or person reasons inferentially about their meaning to build an internal representation that allows to act or interact accordingly. The second view on the contrary assumes that the environment already does encode for some information, i.e. environmental objects are assumed to actually have an inherent meaning of some kind that is directly perceivable by the animal. Thereby, the animal does not have to deliberate extensively in order to comprehend what it can do or cannot do with an object; instead this information is supposed to be already contained in a meaning-laden environment to be perceived directly.

These two views can be verbalized as the *inferential* or *indirect theory of perception* and the *direct theory of perception* respectively (see Jones, 2003; Chemero, 2003).

One advocate of the latter form and actually one of its founders who elaborated a large part of the theoretical background of this view is cognitive psychologist James Jerome Gibson. During his lifetime Gibson has developed and established a direct perception approach, which he called the *theory of affordances*. This new view on perception became also known and popular under the name of the *ecological approach* or simply *ecological psychology*.

The actual term of an *affordance* has first been coined by J.J. Gibson as early as in 1966 though he has worked on that concept for even longer. See Jones (2003) for an essay on the emergence and evolution of the affordance concept as it was developed by Gibson. The probably most important and exhaustive definition on affordance theory is, however, provided by Gibson himself in Gibson (1979).

The interpretation of affordances that is used throughout this thesis and that is one of the cornerstones of the MACS project[1] is as well based on and inspired by this definition. Therefore, Gibson's sight on the topic will now be described in a little bit more detail. Afterwards some extended views on the topic, that have influenced the affordance interpretation followed here, will be introduced in section 2.2. It is considered to be of utmost importance to introduce the definition of affordances applied in this thesis in detail as the topic is not only handled controversially throughout the literature but, as

[1] The acronym MACS stands for *Multi-Sensory Autonomous Cognitive Systems Interacting with Dynamic Environments for Perceiving and Using Affordances*. It is the project that this thesis is based on and will be described in detail in section 3.2.

will be seen later in section 2.3, did already lead to noticeable misunderstandings that have to be avoided in this thesis.

Note that parts of the following introduction are inspired by the evaluations provided in Doherty et al. (2005) that is picked up as well in Rome et al. (2006a). The actual interpretations are strongly related to the different approaches that will be presented later in this chapter.

2.1 Gibson's View on Affordances

Gibson introduces the term of affordances as follows:

> "The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill. The verb to afford is found in the dictionary, but the noun affordance is not. I have made it up. I mean by it something that refers both to the environment and the animal in a way that no existing term does. It implies the complementarity of the animal and the environment." (Gibson, 1979, p. 127)

By this formulation, Gibson describes a reciprocal relation between the abilities of an animal (or agent) and the features of its environment. This relation is grounded on what the environment or objects within that environment afford, i.e. how the agent can interact with those objects or what it can do with them. A stone, for instance, affords being thrown or used as a paperweight. It might as well be used as a hammer or as a spear if it has sharp edges and is attached to a stick.

This way of looking at objects naturally implies that affordances are subjective to the specific agent that perceives the affordance. If, for instance, a chair has certain physical properties like being flat, horizontal and rigid it might be able to support a person and thus affords sitting in the eyes of that person. The chair, however, might be too weak or to small for another person who will thus lack the impression of a sitting affordance connected with that chair. Thus an affordance is not just an abstract physical property. Instead, it is always a subjective or relative unique relation between the agent and an object.

Furthermore, it is important to point out that connecting objects with affordances is not the same as ordering objects into categories like e.g. chairs or cups. Gibson describes this point like this:

> "If you know what can be done with a graspable object, what it can be used for, you can call it whatever you please. ... The theory of affordances rescues us from the philosophical muddle of assuming fixed classes of objects, each defined by its common features and then given a name. ... But this does not mean you cannot learn how to use things and perceive their uses. You do not have to classify and label things in order to perceive what they afford." (Gibson, 1979, p. 134)

By this, Gibson postulates that objects do not have to be clustered into fixed labelled categories according to their features. Instead he proclaims, that objects have a dynamic relationship, that is their affordance, with the agent. I.e. the agent learns which actions it actually can apply to those objects and furthermore learns to infer from this which actions it may presumably be able to apply. This relation is in a way dynamic as it directly depends on the state of the agent and the environment. As a change of one of those states could easily affect the affordance and might foil the agent's plans without the direct need of changing the physical properties of either the agent or the object. For instance, a stone that is lying behind a ditch and is therefore impossible to reach for the agent does not afford grasping what the same stone would afford if it was lying on the agent's side of the ditch. Thus, a stone of the right size and weight does not implicitly afford grasping what is a major distinction between simple object categories and such a relation concept as affordances. Gibson pronounces this by stating that "to perceive an affordance is not to classify an object" (Gibson, 1979, p. 134). If affordances somehow categorize objects they do not do this by classifying what an object is but rather by what one can do with that object, i.e. they provide a functional view instead of one that is based on explicitly identifying and categorizing objects.

Gibson goes even further by claiming that the existence of an affordance might as well be unconscious to the agent:

> "The observer may or may not perceive or attend to the affordance, according to his needs, but the affordance, being invariant, is always there to be perceived." (Gibson, 1979, p. 139)

Hereby, Gibson states that an affordance is an innate relation between an agent and an object that does always exist, if it exists. It is not dependent on the agent to pay attention to that particular relation. Instead the agent is able to filter out those affordances useful in a particular situation in order to avoid the impression of "drowning in affordances" (Rome, 2003, p. 4). Nevertheless, the agent must possess the capabilities to perceive the affordance and to act upon it. Thus, the actual affordance perception can certainly be understood as being "according to his needs", or in other words goal-directed.

This definition of affordances might actually seduce the reader to draw a connection between purposeful, goal-directed actions and affordances in a way that one would assume an affordance to be present if simply all preconditions for an action are fulfilled; be it on the agent or the environment side. Gibson himself struggles a bit when he tries to construe their character:

> "[A]n affordance is neither an objective property nor a subjective property; or it is both if you like. An affordance cuts across the dichotomy of subjective-objective and helps us to understand its inadequacy. It is equally a fact of the environment and a fact of behavior. It is both physical and psychical, yet neither. An affordance points both ways, to the environment and the observer." (Gibson, 1979, p. 129)

While this somehow bewildering definition can be interpreted as struggling with the characteristic of an affordance as an relation between the agent and the environment (see

as well section 2.2.1) it contains furthermore the definition of the affordance as being both physical and psychical, yet, assumably, neither explicitly. This actually depicts the difference between an affordance that could be understood as the mere validity of preconditions for an action and the actual concept of an affordance. Affordances are in a way more than just a set of fulfilled preconditions like, for instance, the chair affords sitting because its structure looks like it supports my weight and it is not taken. In a certain way, an affordance invites the agent to act. Take for instance the action of approaching a certain point in space: Under normal circumstances each point in an empty room would be approachable by a cat. Nevertheless, the cat cannot be assumed to perceive for all points affordances of being approachable. If, however, a mouse is trying to hide in a corner of the room, the location of this mouse would suddenly afford approaching. The affordance is hence a concept beyond the preconditions for actions as it includes some sort of offering or invitation for action that is meaningful in that specific environment and for that specific agent; the cat would go to the mouse rather than going just somewhere.

Note that this definition explicitly exclude meaningless actions of being afforded. Turning around on the spot is normally possible, but rather rarely afforded.

To finally summarize Gibson's approach to an affordance theory it is probably the best to put it in the words of his wife and fellow psychologist Eleanor J. Gibson who stated: an "affordance is a resource or support that the environment offers an animal; the animal in turn must possess the capabilities to perceive it and to use it." (Eleanor J. Gibson et al. (1999, p.4) in Wilson and Keil (1999)).

Thus the key concepts of Gibson's affordance theory can be pinpointed as:

- Affordance is a term used to reference to the meaningful possibility for action an environmental object or the environment itself offers to an agent.
- Affordances are relations between a particular agent and a particular object. Thus they are unique to that particular agent.
- The agent must be able to perceive an affordance and to use it though it may as well use it unattended or even unconsciously.
- Affordances are invariant and always there to be perceived.
- Perceiving affordances is not to categorize objects.
- Affordances are not mere preconditions for actions.

These key formulations of Gibson's affordance theory are the basis for the forthcoming section where some of the claims presented here will be discussed and substantiated in a bit more detail. This is deemed to be necessary since, as one might have noticed, this section already contains a fair amount of interpretation of the matter that goes beyond the actual formulations made by Gibson. And as one will see by reading the next section, the discussion on affordances is a very lively and still ongoing discussions, which raises the demand for a clear definition of how the affordance term will be used subsequently.

The following section will henceforth describe a few integral parts of some of the various interpretations and extensions made to Gibson's affordance theory that are felt to in some way influence the approach followed throughout this thesis.

2.2 Extending and Interpreting Gibson's View on Affordances

First of all, one should note that this section is far from being a complete, exhaustive survey on the diverse definitions and interpretations to the protruding concept of affordances as it was defined by Gibson. Here I will rather focus on some of that work that contributes to the basic understanding of how the term of affordances is used in this thesis, to disambiguate its use and to clarify it. This section furthermore aims at providing a brief overview on how the most important interpretation incitements emerged from the discussion and it moreover will shortly sketch the far reaching influence that Gibson's theory has on other sciences.

Nevertheless the start will be made by evaluating the work of fellow psychologist Anthony Chemero who had great influence on the interpretations anticipated above. Note that parts of this overview are influenced by Müller and Stratmann (2004).

2.2.1 Ecological Psychology

One of the probably most valuable views on Gibson's affordance theory has been framed by Chemero (2003). He outlined a complete and consistent interpretation of the concept while reconsidering the whole preceding line of argument that emerged in the field of ecological psychology after Gibson has formulated his beliefs.

As this work by Chemero has a very high impact on how the concept of affordances is interpreted throughout this thesis it will be introduced in a little bit more detail than some of the other views on the topic. The previous discussion that was the starting point for Chemero to create his theory will as well be briefly sketched as far as it has contributed to his considerations.

Chemero's interpretation of the affordance concept arose from the ongoing lively discussion mainly between Chemero, Reed, Stoffregen, Turvey and Michaels. The remainder of this paragraph that aims at shedding light onto the dispute carried out on the subject of affordances is mainly based on Chemero (2003). In the context of this article he refers mainly to Reed (1996); Stoffregen (2000); Turvey (1992); Michaels (2000); Heft (1989, 2001).

Reed and, at least previous to 2003, Stoffregen as well argued that affordances are to be understood as resources in the environment and thus properties of objects that exist prior to an agent that perceives that affordance. The difficulty for the agent or animal in their view was to select the appropriate affordance resource for action or in other words to evolutionary develop the right perceptual system to be able to react to those resources appropriate for that specific species. This can be described as a *selectionist view* of affordances since affordances are meant to create selection pressure on animals and by that regulate their behavior.

Meanwhile, Turvey and Michaels agreed with Reed and Stoffregen upon identifying affordances as properties of environmental objects but they furthermore claimed that the existence of an affordance depends on an agent that can perceive the affordance and act upon it. Turvey called this the concept of *dispositional properties* that need some *complement* or manifestation of circumstances in the environment to appear. Turvey (1992) gives the example of salt soluble in water; an example that was picked up by Michaels (2003) and Chemero (2003) as well. According to this example, it is necessary that salt has the dispositional property of being soluble in water whilst water must have the dispositional property of being able to dissolve salt. Only if these two constraints hold the property of having salt dissolved in water will emerge. Putting it in the words of Turvey one could formulate that the circumstance of having salt dissolved in water manifests due to the presence of both corresponding dispositional properties that complement each other. On the contrary to the view proposed by Reed and Stoffregen, this view can be described as *non-selectionist* since the affordances do not exist a priori in the environment and do not just have to be selected by an animal. Instead affordances are in need of an animal that will bring along the affordance only if it complements a dispositional property of the environment; for example: a banana affords to be edible only if there is an animal that can eat the banana. Hence, Chemero states that according to Turvey an affordance could not exert any selection pressure on an animal as the environmental properties do simply not have an affordance if the animal lacks the according propositional property. See as well Chemero's more recent work for a more detailed discussion in Turvey's approach (Chemero, 2006, to appear).

Before these two views are attended by Chemero he calls the attention to another open question: As different as these two views may be in terms of when an affordance comes into existence, they share the common understanding that affordances somehow represent animal-relative properties of the environment. But if they are environmental properties that are animal-relative, it raises the question what "animal-relative" actually means or in other words to what property of an animal the affordances are relative to.

Since Turvey (1992), endorsed by Michaels (2000), postulates that an affordance does only exist if both, the affordance as dispositional property of an environmental object, as well as the agent's ability to make use of that affordance, i.e. the agent's dispositional property, complement each other. Thus the affordance as property of the object would be relative to the property of the agent; that again is, it's ability.

On the other hand, Stoffregen (2000), endorsed by Heft (1989, 2001), would regard affordances as environmental properties relative to the agent's body size; a view that is based mainly on empirical findings by Warren (1984). Warren did experiments to investigate stair-climbing affordances in which he came to define a comprehensively valid so-called π-ratio relating a subject's body size (and some other closely related features) to the stair step height in order to be able to subjectively determine whether the stairs would afford climbing for a subject or not (see as well Chemero, 2003; Michaels, 2003). Thus the stone of the previous example would only be liftable for an agent of the necessary body size to lift that stone.

This summaries the initial position that showed the need to Chemero and inspired him

to formulate a new definition of the affordance concept. His aim was to clearly tackle those questions concerning, first, the circumstances under which affordances exist and, second, to what property of the agent they may be relative to. The solution which he came up with is quite elegant and shall therefore now be introduced:

He argues convincingly that affordances are to be understood not at all as a property of an object in the environment that is relative to some aspects of the agent but rather as the relation between an agent and the environment itself. More specifically he proclaims that affordances have a structure of affords-ϕ(feature, ability), i.e. affordances are relations between the abilities of organisms and features of the environment (see Chemero, 2003, p. 189). He gives the example of affords-*climbing*(riser-height, climbing-ability) that represents a functional relation between the agent's subjective climbing ability with the environmental features of riser height that in combination and dependent on their actual character either afford climbing or not. Quite obviously, such a relation is neither a property of the environment nor of the agent itself as it is dependent on both the environmental features and the abilities of the agent. Nevertheless, that relation is clearly perceivable because the process of perceiving affordances can be understood as "seeing that the situation affords a certain activity" (Chemero, 2003, p. 187).

This definition yields to Chemero's answer to the first question of whether an affordance can exist without an agent or animal to perceive the affordance: According to Chemero, Heft (2001) has argued convincingly that the Gibsonian view on ecological psychology is closely related to *radical empiricism*. Whilst it would go beyond the scope of this thesis to deeply discuss this concept and its influence on the Gibsonian theory, the focus will lie here on one of the main claims of radical empiricism: "everything that is experienced is equally real" (Chemero, 2003, p. 186). Keeping in mind that Chemero has just defined affordances as being relations and relations as being clearly perceivable, the argument follows directly that to "the radical empiricist relations are perceivable, and anything perceivable is real" (Chemero, 2003, p. 187). Later, he subsumes this point by stating that:

> "Affordances do not disappear when there is no local animal to perceive and take advantage of them. They are perfectly real entities that can be objectively studied and are in no way figments of the imagination of the animal that perceives them." (Chemero, 2003, p. 193)

Affordances are thus real relations that are perceivable and do objectively exist.

The second issue of the question to what exactly of an agent, or animal, affordances are relative to is answered implicitly by Chemero's definition as he interprets affordances to be the relations themselves. He claims that affordances are neither properties of the environment nor of the agent so one can understand affordances more or less as features of whole situations (see Chemero, 2003, p. 185). Therefore, there is no need to define a complement to the environmental affordance property on the agent side, as the affordance does not need a complement to come about. The affordance is rather the relation between the agent's subjective abilities and the environmental features itself. It does not relate complements to each other as it was postulated by Turvey (1992) and

Michaels (2003). Hereby, Chemero explicitly disagrees with Turvey's interpretation of abilities being dispositions. According to Chemero, a disposition that is manifested by its complement deduces that this disposition will never fail. As Turvey interprets the agent's abilities as dispositional properties this would follow in the eyes of Chemero, that if they are complemented by the according environmental property, they as well would never fail. Staying with the example of lifting a stone that was introduced above it is obvious that even if the ideal circumstances are met for lifting that stone there is no guarantee that the agent will actually succeed in applying a lifting action. The agent may drop it in the attempt of lifting for what reason soever. Thus, the concept of defining an ability as a disposition is just wrong. Chemero regards abilities as being rather a *functional property* of an agent instead of a propositional property without the need of asserting whether they fail or not.

Regarding Stoffregen (2000), Warren (1984) and Heft (1989, 2001), Chemero answers that body size as the agent-relative point to affordances is "just an easily quantifiable stand-in for ability" (Chemero, 2003, p. 188). Meanwhile, Warren agrees on this interpretation and confirms the belief that body size is just one aspect that influences an agent's ability strongly; especially in the context of stair-climbing experiments.

This depicts Chemero's ideas concerning the topics of debate brought up by Turvey, Stoffregen, Michaels, Reed and Heft. To subsume the important parts of his interpretations of Gibson's affordance theory, Chemero claimed that affordances are relations between the agent's abilities and features of the environment, more formally they can be formulated as Affords-ϕ(feature, ability). Hereby, abilities are to be understood as functional properties belonging to a particular agent. Moreover, affordances are clearly perceivable and do always (and objectively) exist regardless whether they are currently perceived or not. That already shows the large impact of Chemero's accomplishments on the affordance interpretation followed throughout this thesis since this extension and interpretation of Gibson's affordance theory was already anticipated in the last section when the affordance concept was a priori introduced as a concept of relations between an agent's abilities that are afforded by the features of the environment.

One further influence from the field of ecological psychology should be mentioned here as it points out an important difference between the original wording of Gibson (1979) and the applied interpretation. This view is pointed out by Michaels (2003). Though I disagree with larger parts of her article in favor for Chemero (2003), see above, she calls attention to a vaguely formulated point by Gibson that concerns the connection between affordances and actions or, more or less, abstract concepts. Michaels goes into this distinction as she disagrees on some examples that Gibson gave to be examples for affordances (in Gibson, 1979). According to him, a cliff or snake might afford danger while air affords visual and auditory perception and certain substances afford nutrition (see Michaels, 2003, p. 137). Michaels points out that such abstract concepts as danger, perception or nutrition are not those information-guided, goal-directed actions that in fact affordances should afford. Though such situations could be easily translated to real affordances, as for instance a cliff could afford stepping back or an apple affords eat-

ing, it should be adherently clear that affordances are always to be understood as being ability- and, in this sense, action-related (see Michaels, 2003). This particular part of her affordance interpretation is being shared in this thesis.

2.2.2 Cognitive Science

From the field of cognitive science or cognitive neuroscience there is some work that supports the concept of affordances from clinical research. To name only one example, the work conducted in Humphreys (2001) shows some impressive neuroscientific findings in patients with brain disorders. These patients suffer from differently located lesions or other brain disorders that for instance impair them from associating names with tools, to use or group tools that are connected with each other but were for instance able to identify so-called *non-objects*. These objects were constructed tools or objects with some important part attached in wrong places what would make them impossible to use. Moreover, some of his patients were able to demonstrate the proper gestures when talking about a tool but were unable to actually use that tool. Humphreys directly relates these findings with Gibson's affordance theory that he regards as a theory that answers the question of "how categories of action are selected from visually presented objects" (Humphreys, 2001, p. 408), a process that he regards as rather complex.

Furthermore, when he developed his concept of affordances, Gibson was influenced by a theory on the area of cognitive science or more precisely cognitive psychology, namely the *Gestalt theory*. E.g. Koffka (1935), one of the founders of this theory, argued that the meaning of objects or things can often be perceived as readily as these things themselves. In an attempt to formulate some basic principles of perception, a set of so-called *Gestalt laws* emerged that determined the ways in which objects are perceived that are grounded to a good part on object forms and their alignments and groupings. Gibson himself related the affordance concept to a type of Gestalt law, the law of functional relevance. According to this law, an animal or perceiver of a scene groups those parts of its perception that are important for the ongoing action. That is in a way closely related to perceiving what one can do with an object or in other words what it affords (see Gibson, 1979; Jones, 2003; Rome, 2003).

Thus in the area of cognitive science, there was provided some evidence and incitements for Gibson's theory, where the principle of functional perception and the evidence found in the field of neuroscience are among the most important in terms of affordances. See as well Fitzpatrick et al. (2003) for a short summary of the neuroscientific correlates of affordances.

2.2.3 Artificial Intelligence

Gibson's affordance concept was of course picked up in the field of artificial intelligence as well. While the research that directly pertains robotics will be introduced in the first half of chapter 3, some other approaches of artificial intelligence will be highlighted here that picked up Gibson's affordance theory.

One of them that is definitely worth mentioning is the work of defining the concept of

lifeworlds that was carried out by Agre and Horswill (1997). Agre and Horswill shaped the term lifeworld by stating:

> "We will use the term lifeworld to mean an environment described in terms of the customary ways of structuring the activities that take place within it – the conventional uses of tools and materials, the "loop invariants" that are maintained within it by conventional activities, and so on." (Agre and Horswill, 1997, p. 114)

In other words, Agre and Horswill define a lifeworld as something that goes beyond a mere physical environment describing as well those "patterned ways in which a physical environment is functionally meaningful within some activity" (Agre and Horswill, 1997, p. 114). Agre and Horswill themselves relate this concept of encoding functional meanings in an environment to Gibson's affordances. They go on by describing and interpreting affordances as in a way *phenomenological*. By this they mean that the agent learns the phenomena, or outcomes, of its actions from its intuitive and subjective experience; the experience that it gathers when actually applying those actions. Therefore, this specific agent has acquired a phenomenological understanding of what actions that object as well as comparable objects might afford. This represents a clear function learning mechanism, a monitoring of own actions and an association of what actions can be used with what objects. Besides the interesting point that Agre and Horswill define the perception of affordances as being phenomenological, they explicitly point out, that the lifeworlds they have defined are highly subjective to the agents though they normally are physically equivalent. He gives the example that for instance a cat and a human share the same physical world but their lifeworlds are clearly different from each other. They may overlap in certain aspects but a table, for instance, affords different things for a human than for a cat. Thus, the lifeworlds of Agre and Horswill are a nice affordance-based world definition that is subjective in their functions and meanings to the phenomenological impressions of their inhabiting agents.

Artificial intelligence showed furthermore some important aspects concerning affordance-based robot control especially in the field of *active perception*. In this regard the work of Bajcsy (1988), Bogoni and Bajcsy (1995) and as well Arkin (1998) are amongst the most representative ones. Because of their close connection to the usage of affordances in robotics, they will, however, be introduced in section 3.1.5 to avoid unnecessary repetitions.

At this point it should suffice to briefly sketch the definition of affordances as it is understood by Arkin (1998). He explicitly pronounces that the information needed by an agent to act in an environment does already reside in that environment without the need of mental representations that somehow codify the agent's perception. Furthermore, Arkin points out that the actual environmental perception depends on the current intentions of an agent. He gives the example of a chair that can, dependent on the situation, be interpreted as a sitable surface, an obstacle blocking the way or as a throwable weapon if one is being attacked. This confirms the interpretation introduced in section 2.1 that states that affordance detection of objects is very different from ordering them

into object categories. A chair is not only perceived as a chair and semantically labelled, it is rather perceived dynamically under the aspects of what it affords in the current situation with respect to the goals and intentions of the agent (see Arkin, 1998, p. 244ff).

Arkin's definition of all necessary information residing within the environment also postulates to point out the close connection of affordance theory to the concepts of *situatedness* and *embeddedness*; terms that are frequently used synonymously. As these notions are defined e.g. in Smith (1999) and Seifert (1999), they first came to appearance through the work of Brooks (1991a), who will be mentioned in the context of his subsumption architecture later on in chapter 4. Put simply, a situated agent is seen in the context of its current situation what makes the actions the agent chooses to apply *situated* in that specific context. For example a foraging ant knows from its context how to interact with food and knows how to interact with a predator.

Based on this understanding, e.g. Seifert (1999) points out, that for a situated agent, the environment directs, structures and supports the cognitive processes of that agent. According to e.g. Vera and Simon (1993) this can be interpreted as a direct encoding of situational activity that yields to a more or less reactive action of the agent. In other words, the salient, functional features of an environment lead to an agent's sensation and perception and henceforth to its action (see e.g. Cornwell et al., 2003).

The close relation that this concept has to the Gibsonian theory should be quite obvious. A situated action is seen as residing in the functional properties of the environment and is directly perceivable as a potentiality for action by the one specific, situated agent; that actually fits pretty well to the affordance interpretation introduced so far. The close relation between these two concepts is prevalently approved, e.g. in Kirsh (1991), Cornwell et al. (2003), or more recently in Chemero (2006, 2007).

2.2.4 Design

One last to mention field of science that comes from a whole different direction but still makes great use of Gibson's affordance theory is the field of design. The affordance concept was introduced to this area of research by Donald A. Norman in Norman (2002). The relation of affordances to "the design of everyday things" is hereby quite straight forward. Norman tried to answer the simple question "When you first see something you have never seen before, how do you know what to do?" (Norman, 1999, p. 39). Unfortunately, Norman himself stated in this article that he has unleashed the term of affordances without explaining it sufficiently. This followed that it was heavily misinterpreted in design science. Norman just wanted to find a placeholder that states that the mere appearance of a device already does provide the functional meaningful cues that tell the user how to operate it (see as well Humphreys, 2001, who argued comparably in the field of neuroscience (see above)). Unfortunately Norman did not quite manage to communicate this interpretation unambiguously, since he argues that it was frequently misinterpreted. Therefore, Norman (1999) tries to sort out things by making a clear distinction between affordances and learned *conventions*. He argues that the most common misunderstanding is for instance stating that a scrollbar in a computer interface "affords

scrolling". This is wrong as it is at the most a *perceived affordance* but not a *real* one. According to Norman, a scrollbar does not afford scrolling but it is a convention that is arbitrarily and artificially learned by a user. Hence it is only a perceived affordance. A real affordance on the contrary would just be that the screen that displays something is touchable, whether it yields to an effect or not. This is a concept that is for example picked up by Hartson (2003), who himself defines *physical and cognitive affordances* as referring to the same distinction as in Norman's real and perceived affordances. Hartson even extends this interpretation of different affordance classes and introduces also *perceptual* and *functional* affordances that help the user in perception and allow purposeful actions respectively.

While it is an interesting understanding to define different classes of affordances, the influence of the design-inspired approaches should in the context of this thesis be restricted to once again highlight, that affordances are to be understood as being action related and thus have some physical features and effects. While Norman argues that the "designer cares more about what actions the user perceives to be possible than what is true" (Norman, 1999, p. 39), he argues that the term of a perceived affordance is well applicable in graphical user interface design. It is, however, not applicable in the context of this thesis.

This whole section aimed at showing the most important influences to the interpretation of the affordance concept from various directions of science. Furthermore, the far-reaching influence of Gibson's theory should have become clear as the use and lively discussions on this concept are indeed wide spread depicting the potential power and of course the still required clarification demands of the affordance approach.

As the disputatious character of the affordance debate should have become clear during this chapter it remains to straighten out some of the misunderstandings that arose from the various forms of interpretations to ensure that the concept is picked up correctly. The following last section of this chapter will therefore aim at achieving this clarification.

2.3 Clarifying some Misunderstandings

As has been stated in the very beginning of this chapter it is deemed important to deliver an explicit and clear-cut definition of the applied affordance concept that this thesis is based on. Due to this demand the need arises for extending this chapter a little bit further, trying to straighten out some of the misunderstandings faced especially when it comes to the understanding of affordances as a theory of *direct perception*.

This formulation, that has been proposed by Gibson himself as well as by Chemero (2003), postulates affordances to be directly perceivable without any high level deliberation and has hence caused some misunderstandings or at least lead to some restricted views on this topic.

This misunderstanding will therefore now be highlighted and straightened on the example of Murphy (1999).

In her article Robin Murphy describes three case studies of different robotic systems developed by her students and herself that, according to her, aimed at exploiting Gibson's

ecological approach to perception and henceforth the affordance concept for robotic action selection. These three systems were first a robot that was supposed to dock with a workstation, second a robot for path-following and third a robot that was supposed to pick up trash cans and deposit them into bins. Murphy explains in detail some criteria which are supposed to help the designers of a robotic system to judge whether an affordance based approach is useful in the context in which the robot is about to be applied. Interestingly, she argues that those systems are predestinated for using affordances that can be constrained to minimalist use of perception. She argues this way as she apparently beliefs that an affordance is synonymic to an easily perceivable feature of the environment due to the literally interpreted definition of being perceivable directly. Hereby she goes as far as stating that "the knowledge engineer could create an affordance and place it on the workstation" (Murphy, 1999, p. 107) to enable the robot to easily detect that affordance and henceforth its workstation. The essence of an affordance, she argues, is to be easily perceivable with as few computational costs as possible. Hence, Murphy restricts the affordance concept onto a level of atomic environment perception:

> "Affordances are attractive computationally because they don't require memory, inference or reasoning, or interpretation of a scene." (Murphy, 1999, p. 105)

According to her, this would satisfy the affordance concept proposed by Gibson and would be the direct value of this concept for robotics.

However, I take her to be mistaken in this point. An affordance that a knowledge engineer creates and just places somewhere would, in my understanding of this concept, not be an affordance as a possibility for action, it would be a mere landmark that marks a point of interest. Of course one could argue, that some features of the working station of the robot with that it is supposed to dock can be detected by the robot and they could afford dockability. However, that is not the way Murphy argues. Instead she goes on by stating, that in the second case of the path-following robot she "used two affordances, vision and sonar" (Murphy, 1999, p. 107). That robot was expected to follow a path that is marked with two white lines on each side of the track and that contains some blocking obstacles that had to be avoided. Again, it would match the interpretation elaborated during this chapter if one would interpret a perception of the white lines in a way that they would afford following the middle in between them. Nevertheless, Murphy focusses on the point of perceiving something directly without the need of deliberation or mental representation and she even describes the sensors that are assumed to perceive some environmental features of the environment as being affordances themselves. That is an interpretation that does not fit at all to the definition, Gibson provided and that she actually has repeated quite satisfactory in the beginning of her paper.

Nevertheless, this is a point of view that is not solely held by Murphy (1999) who just serves as an example in this context. This matter is actually quite closely related to the misunderstandings Norman (1999) criticized in the area of design introduced above.

This view can actually be understood if one regards the direct relation between the concepts of situatedness and affordances. Situatedness, as stated above, was first practically introduced by Brooks (1991a) in terms of purely reactive robotics (that will be

topic later on). Thus, it was assumed, that the mere existence of an environmental object that would afford some action in relation to an agent would be directly perceivable in a reactive manner.

Nevertheless, this point of directly perceiving affordances has been questioned by some researchers since it was formulated. For instance Vera and Simon (1993), who have argued for the similarity between the concepts of affordances and situatedness, state that to "acquire an internal representation of an affordance, a person must carry out a complex encoding of the sensory stimuli [...]" (Vera and Simon, 1993, p. 41). Others stand in for this point of view as well. For example Agre and Horswill (1997) argued that one could easily believe to see a direct perception of affordances if one observes the usage of everyday things by human subjects. They argue that the items of a human's everyday life are especially designed in a way that it becomes explicitly easy to perceive what they afford. E.g. handles are seen to be perceptibly suited for picking things up. But this does not per se call for an easy underlying perceptional process at work. This view is also endorsed by MacDorman (1999, 2000) who subsumes that perception, although it might look easy if it is only observed in living beings, is fairly complex. MacDorman gives the nice example of: "People who know more don't take longer to think!" (MacDorman, 1999, p. 20). He states that humans are capable of applying the three key aspects of thinking, namely its systematicity, productivity, and inferential coherence. These aspects are predestined to yield to the impression that a human being does not deliberate at all on its percepts. Especially if humans are very used to some action. MacDorman explicitly formulates:

> "It is not surprising that Gibson underestimated the computational complexity of vision, since he wrote before researchers had begun to explore it seriously. [...] Thus, the brain may need to process sensorimotor data extensively and to spend time learning what kinds of invariance are useful in recognizing affordances." (MacDorman, 1999, p. 1003)

Yet another misconception in the understanding of direct perception is straightened out by Duchon et al. (1998) as they explicitly argue that an agent does not merely respond to a directly perceived stimulus by applying the action that is afforded in that situation. It is not controlled by the environment. It can rather use the information provided by the affordances of a situation and reason about them in a goal-directed manner selecting those afforded actions that will lead to its goal.

Given these views on affordances, I strongly agree with Vera and Simon (1993), Agre and Horswill (1997), Duchon et al. (1998) and MacDorman (1999, 2000) as I do not think either that affordance perception is in that category direct as it is interpreted e.g. by Murphy (1999). As Gibson states, affordances can be perceived directly but in my opinion this does not exclude that there may be the need for perceptional processing if faced with a robotic system. While e.g. Humphreys (2001) shows various brain disorders that affect affordance representation and successful usage of affordances in various levels it is not to assume, that humans do no deliberation whatsoever on affordances. Thus, detecting colored blobs or determining distances to obstacles, for example, should not be regarded as something that violates the concept of direct perception just because

there is an extra algorithm for it applied in the robotic side of a system. Although the main consensus of Robin Murphy's statement is still the same in her well-known and more recent work of Murphy (2000), she herself actually restrains the interpretation of avoiding all deliberation of perceptual input partially by referring to some affordance related perception algorithms that "can be accused of some inference and interpretation" (Murphy, 2000, p. 90). She then restricts her postulation to a point that she assumes the affordance concept to involve a significantly lower level of sensor interpretation.

Still, it is to say that perception itself is a complex process if it has to be solved computationally and this even applies to such fundamental subtasks as detecting affordances. The declaration of Gibson that his affordance theory is a theory of direct perception should therefore be interpreted with regard to the arguments depicted in this section.

3 Using Affordances in Robotics

During the last chapter, the affordance concept as it is being interpreted and used in this thesis was introduced in detail. Having a concept like this defined now raises the question of what is the relation of this all to robotics? How can we use and exploit the affordance theory? What benefits arise? And eventually: Has something like this been tried before?

This chapter aims at answering exactly these questions by giving an overview on current state-of-the-art affordance robotics ordered by the benefits they are targeted at. In the following section, different views on the relation between the affordance concept and robotics will thus be shown that have an impact on different parts of robotics and might actually enhance them. Though there is not yet very much work done that exploits affordance theory in a way deep enough that I deem adequate, examples of those systems that are related will be briefly introduced within the corresponding sections. Please note, with regard to these examples, that I am not aware of any current systems that exploit affordance theory to a notable degree more than those presented in the next section.

Section 3.2 will then introduce the MACS project that aims at exploiting the Gibsonian theory Gibson in terms of a complete robot control architecture incorporating affordance-based learning, control and goal-directed task execution into an integrated system rather than dealing only with a subpart of the possible benefits.

3.1 Advantages and State-of-the-art

The relevance of the affordance theory as it was introduced and interpreted during chapter 2 lies especially in those fields of robotics that demand a high level of autonomy, complex adaptive perception, human-robot interaction and that are placed in cluttered and even dynamic environments that are unbound in their complexity.

The following subsections will focus each on a subject related to this list and explain how a robotic system may benefit from affordance theory in that particular case and what has already been achieved by other groups.

As one will see in the following, most of the introduced approaches make only little or implicit use of the affordance concept. On the one hand, this section should thus be considered to pick up some of the motivation that led to the MACS project described in the second part since this motivation is also applicable for this thesis. On the other hand the MACS project is, as on will see, based on an explicit affordance representation that as well forms the underlying concept and design basis for the task execution component developed in this thesis. Showing that the introduced approaches oftentimes use a rather implicit usage of the affordance concept should thus motivate and justify the need for

an explicitly affordance-related concept and design like the one that will be developed later on.

3.1.1 Flexibility and Adaptiveness

Affordance theory offers the potentiality to be able to react very reliably to any deficiency of the environment or the robotic system. As the whole concept aims at recognizing the functional relation of an object to the system and, therefore, to assess the actions that object affords, the object itself does not matter that much. If, for example, a robot is supposed to reach out for an object that lies on a table, it might need to extend its reach by some sort of appropriate tool. As the system should be able to recognize objects that afford the reach extension it is not necessary to find that one specific tool that is previously known to be a grasping device. Instead, the robot might use a stick, a book or a counter weight that allows it to shift its center of gravity in favor of reaching farther. Moreover, if it can detect affordances reliably, it should, for instance, be able to adapt to both, broken tools or inoperable parts of the robot itself at least as long as this is still possible. Henceforth, a robot that really exploits the affordance concept would be far more reliable and successful if it faces cluttered, dynamic environments that demand for a high degree of flexibility and adaptiveness. This flexibility outcrops especially if the robot's tasks subsume object manipulation or tool use as it is for instance a characteristic requirement in service robotics. Nevertheless, these criteria already apply if the target is only navigation in dynamic environments.

In the field of autonomous navigation one approach is especially related to affordances, namely *optical flow* that can be described as the perceived visual motion of objects, items, or features with respect, in this case, to the robot. It is based on the understanding, that the ecological stimulus for vision is not a mere sequence of static images but rather a constantly changing flow of information that encodes both the movement of the perceiving agent and the topography of the environment (Lee, 1980). It is closely related to object tracking as in flow field approaches the movement of feature points is often tracked over a sequence of images to yield a relative motion estimation for the agent itself or for moving objects (see e.g. the more recent work of Brox et al., 2006).

Gibson himself related his affordance concept to optical flow by describing the optical flow field that one perceives when flying a plane and how a pilot exploits its characteristics during landing (see Gibson, 1947, 1950; Gibson et al., 1999; Jones, 2003). Based on this argumentation, the relation between affordances and optical flow was also related to navigation and collision avoidance in studies on animals and humans (see e.g. Gibson et al., 1999, for a short overview).

In the area of autonomous robot navigation, there is, for example, the work of Duchon et al. (1998) who apply optical flow to achieve affordance-based navigation. They develop some basic control-laws, inspired by animal behavior, mimicking, for instance, obstacle avoidance, wandering, escaping and chasing behavior.

Duchon et al. explicitly relate their approach to Gibson's affordance theory and even name it *ecological robotics*. They base their approach on interpreting the purpose of vision in animals as the generation of relations between the environment and the animal that

guarantee the animal's survival, which is most certainly the goal with the highest priority for that animal. Their approach uses the relative movement of environmental objects in order to align the robot in its orientation. If, for instance, their robot is driving down a corridor it will react with heading corrections to the optical flow field generated by the walls on either side in order to stay in the middle of that corridor actually mimicking behavior observed in bees. Duchon et al. (1998) explicitly point out that there is no need for modelling the environment if optical flow is used for navigational purposes and define this to nicely fit to the requirement of directly perceiving what the environment affords for navigation. Nevertheless, remember their statement of understanding affordances that allow goal-directed action selection (see section 2.3).

A similar and related approach is applied e.g. in Diaz et al. (2001) who do not use optical flow but extract local orientation information from laser data to build up a *potential field* that represents preferred orientations that are influenced, or in their eyes afforded, by the environmental constraints. For instance a door is regarded to afford passing through. That is why some distinct points in its proximity appear attractive to the robot. In other words, these points afford driving to and can be seen, in a way, as pulling the robot towards them. As a result, multiple of these points that Diaz et al. (2001) distribute around a door would create a potential field that virtually pulls the robot through that door. Nevertheless, their use of the affordance concept is unfortunately only vaguely defined and leaves much room for interpretation.

Although the overall system behavior they achieve can actually be interpreted from an affordance point-of-view, affordances are not explicitly regarded in their system's design. In the context of defining an affordance-based task execution component that is meant to allow to demonstrate the actual usage of an explicit affordance representation, this work of Duchon et al. (1998) and Diaz et al. (2001) actually motivates to focus on the affordance usage of purposeful actions instead of navigation (see chapter 4).

Beyond the just mentioned work on affordance-based navigation, it was Stoytchev (2005a) who conducted one of the first comprehensive and most interesting studies that is related to affordance-based object manipulation and tool-use. In his work, a robot, equipped with an arm, learns the observed outcomes of a variety of simple behaviors, which Stoytchev categorizes into the groups of *exploratory behaviors* and *binding behaviors*. Hereby, exploratory behaviors are meant to explore the environment by, for instance, sliding the arm or positioning the wrist. The binding behavior (Stoytchev defines only one), on the other hand, binds an object to the robot, i.e. it grasps for something. Stoytchev tests the quality of the learned representations on tasks in which the robot has to extend its reach using more or less appropriate tools on a table. Thereby, Stoytchev investigates how the robot reacts to broken tools.

Stoytchev formalizes the learned affordances in what he calls an *affordance table* that holds information of the applied behaviors, their parameters, correlated observations, and the number of trials and successful applications of the behaviors. Hereby, he explicitly emphasizes the point that the affordance representation should be *grounded* in the robot's perception and behaviors. This means, that he as well interprets tool affordances as the relation between perceivable features and the capabilities of the actual robot because if

they are grounded in the robot, the affordance relations become unique to that robot. Stoytchev (2005a) demonstrates a system that is very well able of performing affordance-based extension of reach or grasping tasks using differently shaped tools. Furthermore it is capable of adapting to sudden inappropriateness of formerly used tools, i.e. if they are broken (Stoytchev, 2004, 2005a,b).

As a summary it remains to say that using affordances in the right way allows for a much greater flexibility and adaptiveness of a robotic system. Robots show good performance in navigational tasks in dynamic and cluttered environments using, for instance, an affordance-based optical flow or potential field approach (Duchon et al., 1998; Diaz et al., 2001). Nevertheless, the implicity of the applied affordance representation used in these two groups motivates to focus the design of the task execution component rather on purposeful actions than on mere navigation.

Moreover, Stoytchev (2005a) showed that robots gain significantly in flexibility and adaptiveness with regard to object or tool selection and use when applying an affordance-based approach because that is able to substitute inappropriate tools as well as to adapt autonomously to tool insufficiencies.

3.1.2 Abstraction and Learning

Another very important aspect of the benefits of affordance theory to robotics is if the system can actually learn affordances itself. As it was already indicated in 3.1.1, being able to learn the affordances of objects to the robot bears an immense power when it comes to flexibility and abstraction in every-day environments.

Staying first with the example of Stoytchev (2005a), the robot he has introduced is able to monitor the outcomes of the behaviors it applies to the tools and other objects in its world. To acquire this ability, the robot has to run through what Stoytchev calls a stage of babbling. In this stage the robot tries out different behaviors on the objects presented in the environment and observes and learns in what outcomes the different combinations result in. This learning-by-doing behavior is for instance similar to behavior shown by playing infants (Cooper and Glasspool, 2001).

Stoytchev's robot is moreover able to predict the outcomes of behaviors that it has tried out before. This enables it to dynamically evaluate if an behavior or what behaviors at all will be successful in the current situation. It allows the robot to obtain a valid sequence of employable behaviors that lead to the desired effect and is hence a very powerful ability.

Similar to the door passing example of Diaz et al. (2001) (see section 3.1.1), Slocum et al. (2000) apply an evolutional learning approach to this domain. Though their affordance interpretation is seen as being restricted to depict a relative relation between the agent's body size and an environmental object (see section 2.2.1), Slocum et al. train continuous-time recurrent neural networks to be able to judge if a doorway is wide enough for an agent to pass through. Similar to Stoytchev (2005a), their system is able to predict some parts of their environment that is in this case not the direct outcome of actions but after all the dynamics of monitored objects that they have learned previously.

The system of Slocum et al. (2000) is hereby capable of catching an object that was only seen for a short time and then occluded from view. Their interpretation and explicit use of the affordance concept is, nevertheless, not consistent with the interpretation of affordances developed in this thesis as they restrict the power of the theory by relating solely to an agent's body size.

Another approach that fits better to the applied concept of affordance theory and that is related to affordance learning is followed by MacDorman (1999, 2000). MacDorman aims at exploiting the developmental abilities of a robotic system. He argues that if a robot is able to detect the consequences of its actions or, moreover, if it can generalize from other similar experiences, it has the ability to learn affordances. He specifies the learning process further by stating, that the robot has to learn the spatiotemporal correlations of the robot's motor signals, its current perception, originating from different sensory modalities, and its internal variables. MacDorman has developed a system capable of learning these correlations, that in his eyes correspond to affordances, by generating affordance representations. These representations are the result from the extraction of those environmental features that have changed since the action was last applied since those features that have changed in the meantime cannot be of any importance to the affordance of this action.

MacDorman demonstrates that his robot is able to project its learned spatiotemporal model into the future allowing it, similar to the successive work of Stoytchev (2005a), to make predictions about the consequences of its actions, without the need of applying them. MacDorman describes these learned correlations as a kind of embodied prediction about the future that can be evaluated for planning purposes (MacDorman, 1999, 2000).

The actual value of this ability becomes particularly clear if one assumes a situation where the robot has learned an abstraction of previously learned affordances and their outcomes that enables it to judge by a mere projection of an action into the future, rather than by trying it out, that it is an unwise idea to jump off that cliff.

Yet another approach dedicated to exploit the affordance concept for learning in robotics has been attempted in Fitzpatrick et al. (2003); Metta and Fitzpatrick (2003); Fitzpatrick and Metta (2003). Here, they train a humanoid robot to learn the effects of actions like pushing and pulling on objects by simply trying the actions out and observing their effects. The effects of these actions, i.e. the resulting movements of the hit objects, are tracked using an optical flow approach. The observed parameters are then transformed into what they call a *motion signature* (Fitzpatrick et al., 2003; Fitzpatrick and Metta, 2003). This signature is represented as a probability histogram depicting the likelihood of observing the associated object to roll or slide into a specific direction relative to the principal axis of the object. In other words the motion signature holds the learned and observed knowledge about where an object will probably roll if it is hit with regard to the relative direction from which it is hit. Thus the robot learns the effects of several actions on several objects and represents each as a motion signature. They interpret these signatures to represent the objects affordances.

In a subsequent step Fitzpatrick et al. associate the motion signatures with <object, action> pairs that is the corresponding object and the action that resulted in the observed movement. They argue that the knowledge acquired through this association

can for instance be used to choose, in a goal-directed manner, an action appropriate for achieving the result of moving a known object into a desired direction. Fitzpatrick et al. demonstrate furthermore, that their robot is able to mimic the effects of an action that is conducted by a human and solely observed by the robot. Of course it does not analyze and mimic the action itself but its effects, as the human might have to apply a completely different action that results in the same effects; for instance pushing rather than pulling for moving the object towards the robot.

Unfortunately, the approach of Fitzpatrick et al. is limited in a way that it is dependent on determining the identity of objects to reason about appropriate actions. This severely curtails the flexibility and abstractness of the affordance concept that has as a main idea to determine the abilities an object affords based on its features but not on the specific instance of that object.

This directly leads to the major drawback of this approach that is that the robot has no competence whatsoever in terms of abstraction. It is not able to infer from what it is observing to more general rules of functionality. I deem this, however, to be a crucial requirement when seriously exploiting affordance theory for robotics. It should be one of the theory's major benefits that the robot should not have to learn and remember each object it may encounter and what effects the robot's actions had on that object in the past (i.e. its motion signatures) in order to determine what it affords.

Fitzpatrick et al. describe their system as an initial step towards artificial cognition which explains these shortcomings and they themselves state that it is only the first stage on the way to developing "more complex behaviors which rely on the understanding of objects as physical entities with specific properties" (Fitzpatrick and Metta (2003, p. 16) and see as well Fitzpatrick et al. (2003, p. 3144)). Unfortunately, they neither make this statement more explicit nor take it up in more recent work so that it is left to their readers to interpret it.

As well as this approach serves as an example for learning the effects of actions by observation as rigorously it shows that the mere prediction of the outcomes of actions is not alone the most important point when it comes to the benefits of affordance theory to robotics. The learning of action related functionalities should instead bear the potentiality of generalizing over the agent's experiences as it is demonstrated e.g. in the work of MacDorman (1999, 2000).

3.1.3 Usability

Another important part in robotics in general is that they are normally meant to do something purposeful; and autonomous robots are no exception. In real world environments, the robot must possess the capabilities of flexibly interpreting both its goals and its surroundings. Recalling the aforementioned example of some sort of service robot, a normal task would be something like cleaning up a park. A semantically abstract formulated task description like "clean up the park" may imply such things as picking up rubbish on the park's alleys and lawn, or to empty garbage cans. It does, however, also imply that the flowers are to be left in their flowerbeds and that the decorative stone arrangements will remain at their place. Speaking in terms of usability, it is obvious,

that a user will most certainly prefer a robot that is able to handle abstract or even incomplete task descriptions like "clean up that park" instead of having to point the robot at every peace of rubbish that it is supposed to pick up.

A robot that is able to learn the affordances of objects may easily learn that small items on alleys that do not move afford picking up while flowers on the contrary afford to drive around them. Nevertheless, learning procedures as those described in 3.1.2 that learn affordances by autonomous interaction like playing might have problems with this level of abstraction as a robot that is able to pick up trash will probably be as well able to root out flowers. The learning can, however, be easily extended to include some supervision and reinforcement. Through this, a robot may generalize that an accumulation or formation of rocks on the lawn does not afford tidying up whereas a single rock on an alley would indeed afford that action. Exploiting the affordance concept always implies that the robot does not have to actually face every single item that it is supposed to handle. If it can infer the actions that the objects afford and generalize on them, then it can maintain tight and powerful knowledge representations.

Thus, applying the affordance concept to robotics depicts a very promising and above all very realistic approach to achieve a high level of autonomy as it is required for instance in the park cleaning example. Consequently, the usability of the robot becomes significantly more adequate if the user only has to refer to functional properties known to the robot. This is due to the close relation of the affordance concept to how people think as it is familiar to say for a human to "pick up the rubbish on the alleys" and as well for a robot who knows about affordances and about what kind of objects afford to be picked up. Thus, even if the robot is facing poor or incomplete task or goal descriptions by the user that do not point out each and every single sub-step, the robot should be able to interpret them and to adapt flexibly to the environmental constraints at hand.

Such a system can be used very easy even by inexperienced layman as the everyday user that simply does not know or care about affordances or the problems nowadays robots are facing.

Speaking of introducing affordances into robot task execution, this way of formulating actions for the robot to perform does again motivate the introduced design of the task execution component in chapter 4 that delivers such an easy to use response point.

3.1.4 Scalability

In this context, the term scalability subsumes the challenges of exporting a robotic control system to multiple kinds of environments, that differ in their dynamics and richness of objects contained therein, and to convey the system to other kinds or differently equipped robots.

The usefulness of a system that exploits the affordance concept becomes evidently clear if one looks first at the scalability of the system to other environments. If a robot already knows that round, hollowed, rigid objects afford to pour liquids in, it will know what to do with the glasses found in the environment of the kitchen, with the vases in the living room or with the buckets in the garage.

If using an affordance-based approach, the system designer does not have to model

every new object that the robot may encounter in a different environment. The robot should be able to infer from the appearance of that object how to interact with it and how to use it. Thus a system designer does not get overwhelmed by either modelling every possibility an environment may show or by extensively training the robot to learn every single object that it is supposed to be able to handle. Thus, there is simply no need to both model the objects by a designer or to remember and recall them by the robot. This is a huge benefit in terms of scalability of a system to other environments.

On the other side, the robot or its abilities can change. But a robot that can monitor the outcomes of the actions it applies, as those introduced above (see 3.1.1 and Stoytchev, 2004, 2005a,b), should be able to adapt to changes in the outcomes of this actions. This is comparable to the case of the broken tools that Stoytchev has investigated in his aforementioned study.

Another approach extending both, the environments faced as well as the systems that are supposed to handle the environments can be found in Cornwell et al. (2003) who aim at exploiting the affordance concept in multi-agent architectures. Since this approach is not directly related to the issues discussed here it should suffice to say that Cornwell et al. argue that an affordance-based multi-agent system offers, by encoding functionality in objects, easy extendability to different agents. An object might afford different actions to different agents distinguishing such an approach from those that need to define explicit actions for each agent on each object.

Nevertheless, when thinking this concept further it can as well be adapted to the task execution component that will be developed in this thesis. Since this component encodes actions for agents, the scalability considerations presented here also apply, when adding new actions to the system (see 4.3.2).

Summarizing this section on scalability, it should have become clear, that the affordance approach offers great benefits in terms of scalability as it implies only a manageable amount of work to apply a robot or agent control architecture in a diverse variety of environments and even in agents with varying capabilities. That is, because the affordance theory already represents a quite abstract interpretation of the environment that is, nevertheless, easy to use for an agent.

3.1.5 Perception

Classical perception approaches see the process of perception in a way of sensing the environment, interpreting and reasoning on an environmental model constructed of the perceived information, and eventually acting upon that representation and reasoning. Perception is hence regarded as a decoupled process that stands for its own in this *action-perception cycle* as it is defined e.g. by Neisser (1976).

With the ascent of interpreting agents as being situated in their environments (see section 2.2.3 and Brooks (1991a)), however, this view was extended to imply a direct feedback component (see Fig. 3.1). This direct feedback of the environment bypasses the higher cognitive levels of deliberation and representation allowing for actions that are able to directly react to changes in the environment. Brooks (1986) was one of the first who proposed this feedback component and build a complete *reactive* robot control

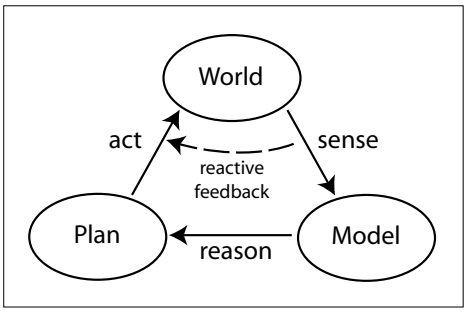


Figure 3.1: **The Action-Perception Cycle**. The figure is adapted from Arkin (1998, p. 246). The original action-perception cycle as it was defined by Neisser (1976) did not contain the direct environmental feedback depicted here by the dashed line.

architecture, known as the *subsumption architecture* (see section 4.3.2), that was able to fulfil tasks like navigating through an environment while robustly avoiding obstacles without the need of a deliberative component. Arkin therefore called this approach not only reactive but even *reflexive* in Arkin (1990).

As already mentioned in section 2.2.3 the approach of situated agents is regarded as being closely related to the Gibsonian theory of affordances because the direct feedback that might trigger an agent's reactive response in form of an action can easily be interpreted as an affordance for action. Brooks's approach is thus an early example of the tight coupling between directly perceiving environmental features and actions of the system. If, for instance, an object blocks the way in a corridor it affords avoiding and therefore, the agent will move around it. The actual benefit for a robotic agent is nicely demonstrated by Brooks, as such a system is only able to reliably and robustly react to dynamic environments without using a complex deliberation component. This is because the process of deliberation normally slows a system significantly down and hence it would not be able to react fast enough to sudden changes in the environment like, for instance, when a door is being closed directly in front of it.

Nevertheless, there is more to mention, when it comes to perception and affordances: An affordance describes the environment's opportunity for action and thus directly leads to a view that relates both the perceptional processes of an agent and its actions closely with each other, namely the two concepts of *action-oriented perception* and *active perception* that will again be picked up in chapter 4 especially when introducing the notion of tasks.

Action-oriented Perception

To start with an explanation of the first one, Arkin (1998) defines action-oriented perception as the tuning of the agent's perception to the needs of its motor activities. In other words, the intentions of the agent for action specify what should be perceived in the environment. This can be understood as a top-down approach to perception because

the actual perception is organized in an as-needed basis, rather than as the decoupled component as it was interpreted in the early work of Neisser (1976). This view is based on the assumption that it is not at all necessary to completely perceive a situation and to analyze all its facts in order to fulfil a certain action. If, for instance, a cat wants to pass through a cat flap it does not care, if there is a pair of shows standing next to the door. It is the cats intention to accomplish the action of passing through that cat flap. But therefore, it is sufficient for the cat to adjust its perception in order to segment the brighter area in the door that corresponds to the transparent cat flap. Thus the intentions of the cat influence its perceptional strategy to select those features of the environment that are of interest in that particular situation and with regard to the desired action.

Thinking this idea further shows the opportunities offered by action-oriented perception when being confronted with an affordance-based approach. If one, for instance, knows that blue objects of a certain size afford lifting, than the agent can adjust its perceptual processes to explicitly look for blue objects if it intends to lift an object. More generally speaking, if those features or environmental properties that support an affordance, the agent is looking for, are known, because they were for example learned from previous exploratory experience, the agent can influence its perception in a top-down manner, i.e. tuning it for effectively perceiving exactly those features. Henceforth, the agent would not have to bother about unimportant features in its surroundings. This shows the direct relation to one aspect introduced in section 2.1 when the goal-directed character of affordances was introduced. It was claimed that the affordance as such does not have to be perceived by the agent. An agent is rather able to filter out those affordances that are useful with respect to the intended actions the agent wants to accomplish. In contrast of the classical action-perception cycle the perceptual process is thus not only a stand-alone decoupled system that perceives and evaluates the whole environment. It can rather be understood as an action-related process that aims at extracting exactly those information from the agent's surroundings that are meaningful in its current context. Action-oriented perception therewith rescues the agent from the need of completely evaluating an environment and from detecting all affordances there are; it is the mechanism that keeps the agent from "drowning in affordances" (Rome, 2003, p. 4).

As a side note, it is to remark that the concept of action-oriented perception is often related to visual focus of attention approaches as the one that is used later on in section 5.4.1.

Active Perception

The second view that proposes such a tight coupling of actions and perception mentioned above is the concept of active perception. Action-oriented perception and active perception, in a way, complement each other: Whilst the first focusses on perceiving those information of the environment relevant for the desired action, the latter considers the question of how the physical abilities of the agent may support the search for these relevant features. In other words, active perception tries to somehow orient the agent or

its sensors in order to be able to perceive what it is looking for.

The approach of active perception was first introduced by Bajcsy (1988):

> "[...] it should be axiomatic that perception is not passive, but active. Perceptual activity is exploratory, probing, searching; percepts do not simply fall onto sensors as rain falls onto ground. We do not just see, we look." (Bajcsy, 1988, p. 996)

> "In conclusion we have defined active perception as a problem of an intelligent data acquisition process." (Bajcsy, 1988, p. 1004)

Active perception can thus be interpreted as an active exploration of the environment in favor of supporting the current perceptual needs of the system, i.e. active perception focusses on the needs of perception, rather than on the needs of actions.

This brings along that active perception and action-oriented perception complement one another very well. If, for instance, a robot wants to lift up an object and it already knows that blue objects of a certain size afford lifting, it will tune its perceptual modules to look out for blue blobs of the correct size. The robot may now not immediately perceive such an object and that is the point when an active perception strategy can decisively influence the robot's performance. If the robot is able to somehow change the way it is perceiving the environment, that is, if it is able, for instance, to drive around or maybe to move its cameras, it will have a much better chance of eventually detecting a blue colored blob of the right size. Active perception can thus be interpreted as an augmentation to action-oriented perception. This has as well been expressed by Arkin: "What the agent needs to know to accomplish its task still dictates perceptual requirements, but active perception provides the perceptual processes with the ability to control the motor system to make its task easier as well" (Arkin, 1998, p. 269).

The influence of affordance theory on active perception, however, yields moreover to a perception aspect that is far more interesting. The concept of active perception can be nicely exploited in order to perceive the functionality of objects.

In Bajcsy (1988), Ruzena Bajcsy motivates her newly introduced concept of active perception by outlining a task-driven approach. She sketches a system that is seen as being embodied in its environment. It is a participant in the current situation and has therefore the capability not only of perceiving that situation but as well of changing it actively. She emphasizes that through this interpretation the possibility arises to study different perceptual strategies that involve the modelling of a robot's sensors, the environment, the objects therein, and the purposeful interaction of all these parts; be it for the purpose of object manipulation, recognition or robot mobility tasks.

In Bogoni and Bajcsy (1995), Bogoni and Bajcsy then exploit this concept to nicely demonstrate the close relation between the active perception approach and the determination of object functionality. Although they do not mention Gibson's affordance theory explicitly it should be imminently clear that the functionality of an object to an agent is closely related to the character of an affordance.

Bogoni and Bajcsy (1995) develop a system that aims at determining the possible functionalities of several tools on the example of evaluating their appropriateness in a

piercing task. Their system applies piercing actions on different materials using differently formed tools. The actions and their effects are observed and represented in what Bogoni and Bajcsy call *force-shape maps* that encode the multidimensional information of the objects shape on the one hand and the force needed to pierce the different tool through the different materials on the other hand. Given that representation, the system has acquired a representation of the objects functionality, or how they call it has accomplished the functional recovery of an object and has therefore determined its *functional features*. Regarding the example of only the piercing action, Bogoni and Bajcsy's system is afterwards capable of choosing a tool most appropriate to that action, i.e. in this case the sharpest tool.

The important aspect of this approach, from the active perception point of view, is that the system of Bogoni and Bajcsy aims at perceiving the functional properties of an object. In the piercing example this is, however, not possible to the system, if the objects simply lie in front of it. It therefore has to interact with its environment. It has to try out the actions for that particular perception of a piercing functionality to be possible. In this case, the perceptual request would be to pierce something with an object in order to bring about the percept of the object's piercing functionality because the system cannot judge from the mere appearance of the object if it is suitable for that action. More generally speaking, Bogoni and Bajcsy formulate that the *intrinsic* properties of an object include not only its geometric shape and its material but as well the dynamics and kinematics of that object; properties that can only be perceived through interaction. The functional features of the object are then defined as being task-dependent. More specifically, the functionality of an object is determined as $Functionality(Object) = Structure + Context + Application$ (Bogoni and Bajcsy, 1995). Hereby, the structure of the object are its intrinsic properties, the context is defined as containing the agent, the environment, and a possible recipient for action, and the application holds the information on how to accomplish the desired action with the corresponding object. Thus they define a relation between environmental properties, the agent and some learned usage of a tool to be able to represent an object functionality. That is an approach that is at least affordance-like if not even closer related.

Bogoni and Bajcsy (1995) demonstrate that their system is well able to learn the functionalities of objects by means of active perception. It is furthermore able to represent the learned functionalities in form of force-shape maps and it is shown that it is able to use this representation in order to select tools appropriate for piercing. Bogoni and Bajcsy furthermore postulate explicitly that a higher-level functionality description could be reached by abstraction from the object's intrinsic features, its context and the application.

The system introduced by Bogoni and Bajcsy (1995) might be a bit overqualified from the active perception point of view as it also includes a close affordance relation in terms of functionality descriptions. An approach in which the active perception component can be isolated more easily is for instance proposed by Saffiotti and LeBlanc (2000).

Saffiotti and LeBlanc (2000) show on the example of the football playing toy dogs of the RoboCup Four-Legged league an approach that integrates active perception as a core component of their architecture. Their work is motivated by the insufficiencies of

the robot dogs because they have a very restricted field of view. Saffiotti and LeBlanc therefore propose a system that is inspired by the reactive approach of Brooks (1991a), the concept of affordances, which they refer in robotics to Arkin (1990), and the active perception approach of Bajcsy (1988). The result of integrating these concepts is a *perception-based behavior* that implements robot gaze-direction on an as-needed basis. Hereby, Saffiotti and LeBlanc introduce the notion of *anchors* that are collections of the available information about physical objects that can be used to be a target for active perception. They show that their robots have the ability of directing their gaze towards such an object, if that is judged to be important enough, in order to be able to perceive it and thus track and localize it. This is actually a nice example for the benefits of active perception as it shows that the robot dog would probably loose sight of for instance a ball if it does not actively adopts its own movement as well as to the movement of the ball.

Yet another approach that exploits the power of active perception in a nice way is that of Fitzpatrick et al. (2003), that was already a topic in section 3.1.2 in terms of affordance learning.

As aforementioned, Fitzpatrick et al. (and Metta and Fitzpatrick, 2003; Fitzpatrick and Metta, 2003) apply an optical flow based approach for learning the effects of actions like pushing or pulling on objects in the environment. While this already fulfils the specifications of active perception as e.g. in the work of Bogoni and Bajcsy (1995), their work exploits that concept in one further, very interesting aspect.

Fitzpatrick et al. uses the active perception approach to augment object segmentation by monitoring the effects of object manipulation by the robot. In their experimental setup, they provide objects like little boxes or toy cars to the Cog robot, a robot torso that is equipped with two arms. The objects the robot has to interact with are placed on a table and perceived by means of a stereo vision system. Fitzpatrick et al. describe the case that an object lying on the table cannot always be easily perceived as an object that is not a part of the table because it is, for instance, aligned with the tables borders in a way that the objects frames and the table frame cannot be distinguished by simply looking at it alone. Fitzpatrick et al. then demonstrate that their robot is able to help itself out of this misery by applying an action to that object and thus moving it a bit. The robot will observe the effect of its action that is the movement of the object. Based upon this observation it will the be able to decide which parts that are perceived as moving parts are connected with each other as they will all perform the same relative movement. This can easily be determined in optical flow approaches. All those parts that move equally therefore belong to the same object. Fitzpatrick et al. show that in doing so, the robot is able to reliably segment objects from the environment and moreover it can detect and recognize its own body parts that are of course visible as well.

The approach of Fitzpatrick et al. is thus a good example of how an active interaction with the environment can help the robot to recognize objects as such and to distinguish the robot's own body from other environmental structures. Hereby, they extend the use of an active perception system beyond the level introduced e.g. by Bogoni and Bajcsy (1995) and Saffiotti and LeBlanc (2000). Concerning the connection between such an

extension to the concept of active perception and that of affordances, Fitzpatrick et al.'s work suggests that unknown structures or objects in the environment may afford some exploratory activity of a robotic system as they convincingly show, how the robot's understanding of the world benefits from the information gathered through the additional perceptual quality achieved by its interaction with the environment.

As a final side note it remains to mention that the overall complexity of perception decreases drastically if one applies a combination of active perception and action-oriented perception. It has been shown that task-driven visual search, i.e. if one is looking only for features relevant to the currently intended action, the time complexity for that search increases only linearly with the complexity of the environment whilst the complexity of a data-driven search, i.e. perceiving everything in the environment, is NP-complete (cf. Arkin, 1998, p. 240).

This section should have provided a good overview of how affordances can be and are currently used in nowadays robotic systems. It showed furthermore that most of the current approaches apply rather implicit usage of the affordance concept that oftentimes expresses itself by giving only an interpretation of how the actual system can be related to affordance theory instead of actively and explicitly using and exploiting it.

The next section will thus introduce an approach that carries on the work that was started by those groups and aims at giving this explicit affordance relation that is as well the corner stone for the considerations that have led to the actual design of the task execution component that will eventually be introduced after the following section in chapter 4.

3.2 Towards an Integrated System - The MACS Project

During the last section of this chapter it should have become clear what potential Gibson's affordance theory and its current derivatives bear when it comes to robotics. Some notable approaches of other groups have been introduced that all aim at exploiting the affordance concept to augment their system's knowledge acquisition, representation, or reasoning, or of course its environmental interaction capabilities. They demonstrate the basic applicability of affordance theory and its uses to these different subsystems. Nevertheless, these approaches, though state-of-the-art, are on a level of fundamental research and are hence restricted to tackle the specific needs of these very subparts. They are either facing the tasks of flexible object manipulation, and even tool use, of learning and action prediction for execution planning and thus for selecting appropriate actions or tools, or of decreasing the perceptual load of the system by performing affordance-based action-oriented and active perception. Hereby, they apply an interpretation and representation of affordances that is specifically tailored to the characteristics and needs of the task at hand.

This, however, is exactly the point where further development is necessary. While the aforementioned approaches nicely show the potential and feasibility of affordance theory to robotics, it is the next significant step to integrate the affordance concept deeply into such a robot control architecture. For that it is not an attached methodology

that somehow augments the robot in performing certain tasks but is instead the key concept that the whole architecture is based on. This demands above all an explicit symbolic affordance representation that is used and can be used throughout the various components of the according architecture. Such a representation was, nevertheless, not provided by any of the approaches presented earlier.

The underlying assumption behind the benefits of an explicit versatile affordance representations and usage is that the resulting system should be able to diversely combine and extend the functionality of its subsystems yielding a powerful, abstract, and general affordance-based approach to robotics. In other words, the system would be able to integrate the different approaches into one highly functional system that truly uses and exploits the power of the affordance theory.

The MACS project, that this thesis is part of, aims at developing such an integrated solution. The remainder of this chapter will therefore introduce the project as far as it is appropriate in this context before chapter 4 will focus on the subpart of that system that is being developed in this thesis.

3.2.1 General Introduction

The *Multi-Sensory Autonomous Cognitive Systems Interacting with Dynamic Environments for Perceiving and Using Affordances* (MACS) project (MACS, 2004) aims at exploring and exploiting the Gibsonian affordance theory and its uses for a robotic system. It is targeted at developing a new paradigm of affordance-based approaches in order to establish this concept as a fundamental part of embodied cognitive systems.

To achieve this, the affordance concept has to form the underlying key principle for both the design and the implementation of a robotic architecture. The project will thus define a structure that bases each component of a hybrid robot control architecture (see section 3.2.2) on a clearly defined symbolic representation of affordances in a way that this project overcomes the aforementioned shortcomings of other current systems by taking affordances as "1st-class citizens" (Rome (2005, p. 6) and see as well Rome (2003)). Henceforth, affordances will not be a mere concept that is solely attached to a robot and restricted to one simple aspect of the theory's power. They are rather a comprehensive fundamental methodology that spreads through every component necessary to control an autonomous robot.

The objectives of the MACS project are thus defined as the development of an explicit, symbolic affordance representation and an according architecture that should be capable of controlling autonomous robots acting in a goal-directed manner in dynamic environments. The robot should be able to perceive affordances in its surroundings, to act upon them and to learn the affordances as a relation between its own capabilities for action and the corresponding supporting features of the environment (see section 2.1). Thereby, the affordance concept should be the driving force that determines the design and actual implementation of each part of this architecture, emphasizing its importance for the approach. It is assumed that providing a robot with an affordance-based function centered view on the environment would significantly increase the robot's performance

in terms of acting flexible and achieving its goals.

To demonstrate the validity and suitability of the MACS approach, a testing scenario has been designed that will be introduced in section 3.2.3. Nevertheless, the affordance representation used within the project as well as the architecture that has been designed around it will be introduced beforehand.

3.2.2 An Affordance-based Architecture

The demand put on the MACS project was to define an explicit, symbolic affordance representation that the whole architecture and all its various components are based on. Such a representation was eventually developed and accepted by all partners. At this point, the definitions which are of importance in the context of this thesis will be introduced. Note that these definitions are in a way simplified as they do not contain all the representational power aimed at in the MACS project but rather the degree suitable in this context. They are based on Doherty et al. (2005) and moreover on Rome et al. (2006a). Refer to the latter source for a complete list of the MACS definitions.

Def. 1 ((Agent) Affordance)
An affordance is a relation between an agent and its environment which affords a capability. The agent/environment relation affords a capability if the agent

1. *has the capacity to recognize that it is in such a relation between itself and its environment, and it*
2. *has the ability to act to bring along that capability.*

This definition is coherent with the affordance interpretation developed during the first chapter of this thesis. It states that the affordance is a perceivable relation between the subjective capabilities of an agent and the features of its surroundings.

An affordance can hereby be represented as:

Def. 2 (Affordance Representation)
An affordance representation or affordance triple is a data structure:

$$(\textit{outcome descriptor},\ \textit{cue descriptor},\ \textit{behavior descriptor}).$$

Hereby, an *outcome* or *cue descriptor* is specified as a list of attribute value pairs including information about both spatial and temporal development of these pairs. Attributes can hereby be features of the environment or even internal states of the robot while the values are not restricted to distinct values but can represent value ranges.

The cue descriptor holds that filtered or raw sensory information that supports the existence of the represented affordance whereas the outcome descriptor contains the data as it was perceived by the robot while previously executing the behavior referenced in the *behavior descriptor*.

This behavior descriptor on the other hand refers to a robot behavior and a set of parameters that were used with this robot behavior when the according cues and outcomes were monitored. You should not here, that the term *behavior* is, especially in the

context of this thesis, a bit delicate. At this point it should suffice to say that here a behavior is understood as some sort of interaction of the robot with its environment. This interaction has a perceivable effect, the outcome, and was applied after monitoring some constraints, the cues. The term behavior as such will nevertheless be defined a little bit different in chapter 4 and will be used according to that definition afterwards. Nevertheless it is used here to guarantee the consistence with the introduced MACS definitions as they are presented in Rome et al. (2006a).

To subsume this definition an affordance is represented by:

- The *cues* for an affordance, that support it. These are the perceivable features or attributes of the environment or the agent and their values or value ranges. They can as well be represented with respect to their spatial and temporal development giving the cue descriptor a connotation of the quality of a time series. Attribute value pairs stored in a cue descriptor can thus be, for instance, the relative distance to a test object, its color, or the different currents propagated to a robot's motors during the last couple of seconds.

- The *outcome* of any action or behavior executed upon the affordance. The outcome represents the changes of the agent and the environment as far as they can be perceived by the agent. Examples would be, that a blue colored blob is being perceived at a higher position, relatively to the agent, if it has applied a lifting action. Another example would be that the blue colored blob did not change its position in front of the agent while the agent itself changed its location after it has applied a pushing action.

- The behavior descriptor refers to the *behavior or action* the robot has applied when this representation was created. To stick with the last two examples this would be a lift or push action combined with the parameters like motor current or crane movement that were used for the particular action.

The affordance as such is thus represented by a behavior or action, the cues that are perceivable before applying that behavior or action and the outcome of this application. Therefore, the affordance representation describes the correlation of intrinsic and extrinsic features, i.e. features of the agent or the environment, with the applicability and assumed effect of an action or behavior. The approach is actually similar to the affordance representation of the *affordance tables* developed by Stoytchev (2005a) that was introduced in section 3.1.1 whereby his definition was on a bit more basic level that makes it harder to integrate it into a complete architecture.

This representation, on the other hand, especially bears the potential to learn and importantly as well to refine the cue and outcome descriptors associated with that affordance through repetition of the corresponding action or behavior. This can also be used to verify an affordance hypothesis that is represented in this way. In other words, this representation clearly supports that an agent can monitor its own actions in the environment as well as those features that made an action applicable as they were present before the action was applied and finally the resulting outcome of that action. It can

form affordance representations by combining these perceived cues and outcomes with the action and can in the following use that representation to reason about affordances.

This affordance representation thus fulfills the requirements of an explicit and symbolic affordance representation. It clearly defines a relation between the agent capabilities and its own and environmental features that make a specific action applicable or in other words afford that action. Henceforth, we overcome the limitation of classical robotic approaches to understand the world as labelled objects that have to be recognized and categorized in order to determine applicable actions by breaking the recognition process down to some basic features of environmental objects that can directly be perceived as affording some action to an agent. This representation thus explicitly supports the whole power of the affordance definition presented during chapter 2 as the perceptional process shifts from an object-centered perception to a function-centered perception allowing for reasoning and deliberation on this functional level.

The MACS project now aims at integrating this representation that was tailored to fit the concept of affordances deeply into a complete robot control architecture to evaluate and presumably demonstrate that an affordance-based robot architecture enables the robot to perform better than a non-affordance-based architecture.

Fig. 3.2 shows the hybrid robot control architecture diagram that was developed in Rome et al. (2006a). Note that this is not the location to argue about different possible architectures that would support the affordance concept. Hybrid control architectures, i.e. architectures that combine reactive and deliberative components, are most widely accepted to be state-of-the-art in this field and the reader is referred to the literature on robot control architectures, e.g. to a recent overview given in Makowski (2004) or regarding some more affordance related discussion to Rome et al. (2006b, to appear).

The different components of this architecture should now be briefly described.[1]

Perception - The perception module gathers data from both real and virtual sensors. Real sensors are those physical sensors that measure for instance distances and wheel encoder tics or that take camera pictures. Virtual sensors on the other side are sensors that correspond to internal states of the robot. These are for instance the activation values of reactive behaviors or the current values of a PID motor controller (this will be explained in more detail in the following chapters). The sensors yield raw data that can be preprocessed in the perception module by means of some multi-sensor feature extraction methods, e.g. for localizing environmental objects (see Rome et al., 2006c, for more details).

The perception module takes moreover care of, what is in the diagram called, the *entity structure generation module*. As the definitions provided above did not contain all parts of the MACS definitions it should suffice here to say that this module generates the appropriate data structures to support the affordance representation that has just been introduced (see Heintz et al., 2006).

[1] Most of the references provided in this section concern MACS internal technical documents. Some parts, however, are also topic in the MACS related publications. The full list of these publications can be found on the webpage (see MACS, 2004).

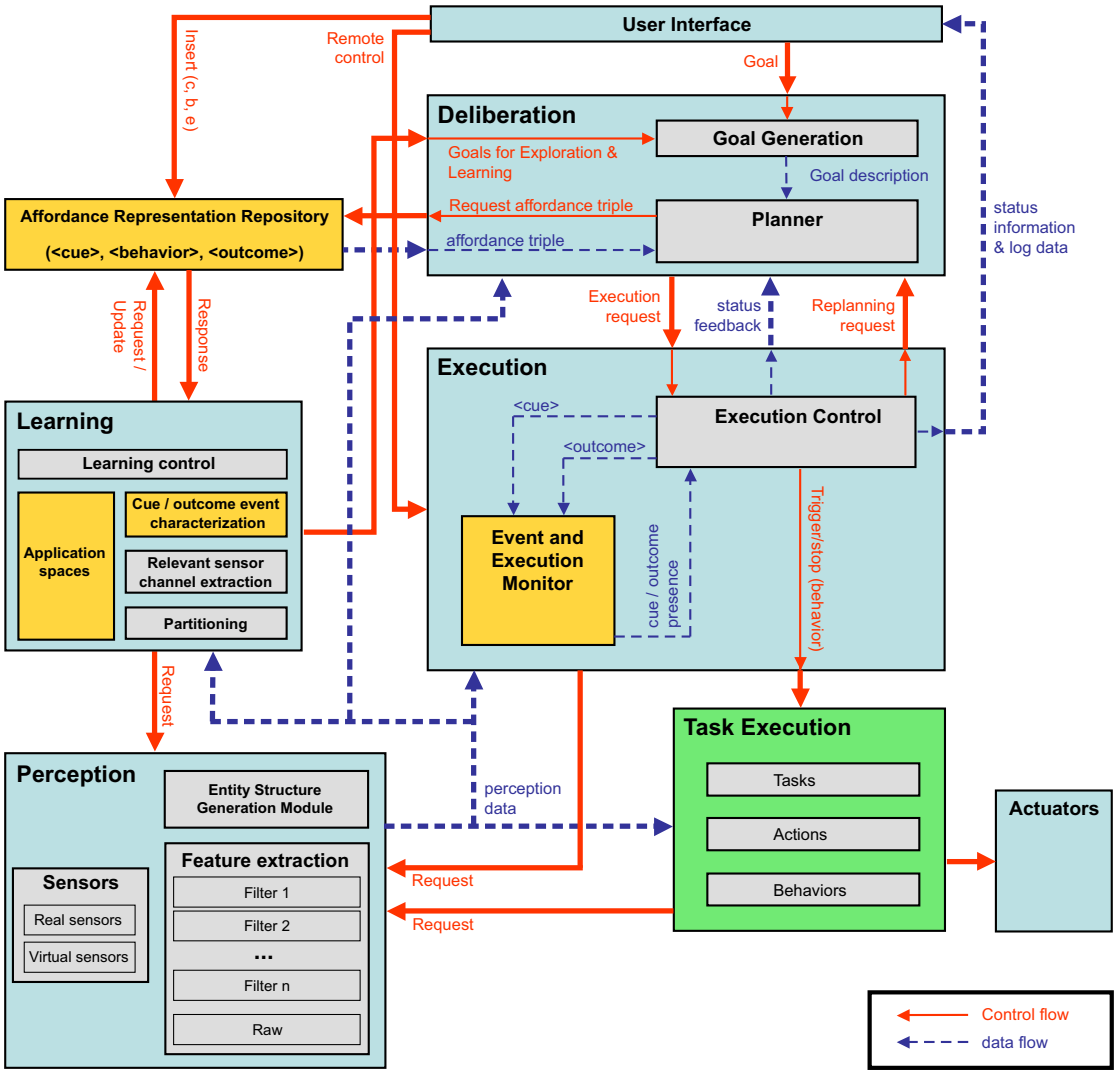


Figure 3.2: MACS hybrid robot control architecture. While the orange boxes represent architectural components that directly distinguish this architecture from non-affordance-based approaches, the green colored box depicts the task execution component that is being developed in this thesis. Adapted from Rome et al. (2006a)

Affordance Representation Repository - The repository holds the learned affordance representation triples.

Learning - The learning module takes all the perception data from the perception module and extracts the relevant data that holds the information about the mentioned cue and outcome descriptors as well as about the current application space of the robot, i.e. a sort of short term memory that stores perceived attribute value pairs and their development during the application of an action. Based on that stored data and the observed cues and outcomes, the learning module generates the according affordance representations and stores them in the affordance repository. The learning module is able to update previously learned affordance representations, to request certain perceptual data of the perception module in order to augment these refinements, and to request explicit disambiguation and exploration from the deliberation component. For an overview of affordance learning approaches refer to Dorffner et al. (2005b) and though the approach is not yet fully defined, first attempts concerning the MACS project can be found in Dorffner et al. (2005a).

User Interface - The user interface is meant to display some status information as well as to allow the user to either guide the robot manually through a specific task in order to trigger meaningful learning or to simply specify a goal that the robot tries to accomplish. The user interface has not yet been specified in more detail.

Deliberation - The deliberation module transforms the commands received from the user interface into a goal and determines an affordance-based mission plan specifying the operators that represent actions to apply in order to fulfil the different steps of the plan. It hands sub-plans to the execution module and should be able to replan if the status feedback from the execution module suggests that a step of the plan failed. For planning purposes, the deliberation module makes explicit use of the affordance repository in order to decide which actions might result in the desired outcome and which environmental features might afford those actions. It is furthermore the instance that should be able to react to goals instantiated by the learning module and not only by the user interface yielding a self controlled learning capability of the system.

Execution - The execution component takes care of controlling the underlying task execution component by triggering those actions specified as operators in the plan obtained from the deliberation component. Hereby, it uses the event and execution monitor submodule in order to ensure the successful execution of those actions. This submodule is the instance that monitors the cues and outcomes that are associated with the affordance representation that was part of the affordance-based plan. The monitor gives feedback to the execution control by confirming or negating the presence of these cues and outcomes. This directly corresponds to the existence of the desired affordance in the environment and to the successful application of an action resulting in the desired effect respectively. If, however, the event and execution monitor negates the existence of either the cue or the outcome,

the execution control has to either select an alternative affordance from the plan or, if none is provided, to ask for replanning.

Nevertheless, the actual assignment of the execution control is to trigger and parameterize the task execution module with those parameter sets specified in the third part of the affordance representation triple.

Task Execution - The task execution module is the component of the system that receives task or action execution requests from the execution control and implements them on the robot by triggering its actuators.

This module will be presented in detail in chapter 4 after the definitions and concepts used for its actual design were introduced and evaluated in the next chapter that is the key component of this thesis.

Note that the task execution module is referred to in Rome et al. (2006a) and some other MACS related documents with the somewhat weaker term of a *behavior system*. Nevertheless, the following chapters will make the change of the name clear in the context of this thesis.

Actuators - Last but not least, the actuators are the actual robot motors for controlling all possible movements, of the system as such or of subparts like pitchable laser scanner.

This architecture is characterized by its deep integration of the affordance concept in every single component of the architecture. While the perception module can be adjusted to support the detection of looked-for cues, the learning, deliberation and task execution modules work explicitly on the affordance representation presented above. The affordance concept as the integral part of each of these modules will also intensely influence the design that will be developed for the task execution module during the next two chapters.

But before I will come to this definition, the actual task that the robot will have to fulfil as it is specified in the MACS project needs to be introduced as it will help to understand the needed complexity of the task execution module.

3.2.3 Defining the Task

The architecture is defined in a way that it should be capable of handling a large amount of complexity. It bears the potential to be extendable in its different components in order to deal with, for instance, different sensors or even robots and is designed in a way that e.g. new feature extraction methods or more advanced planning modules can easily be implemented. But as the actual design and implementation of these specialized modules is each a research area on its own, the task that will serve as a proof of concept for the MACS project has been chosen with focussing on the demonstration of the parts of affordance perception, affordance-based reasoning and affordance-based object manipulation.

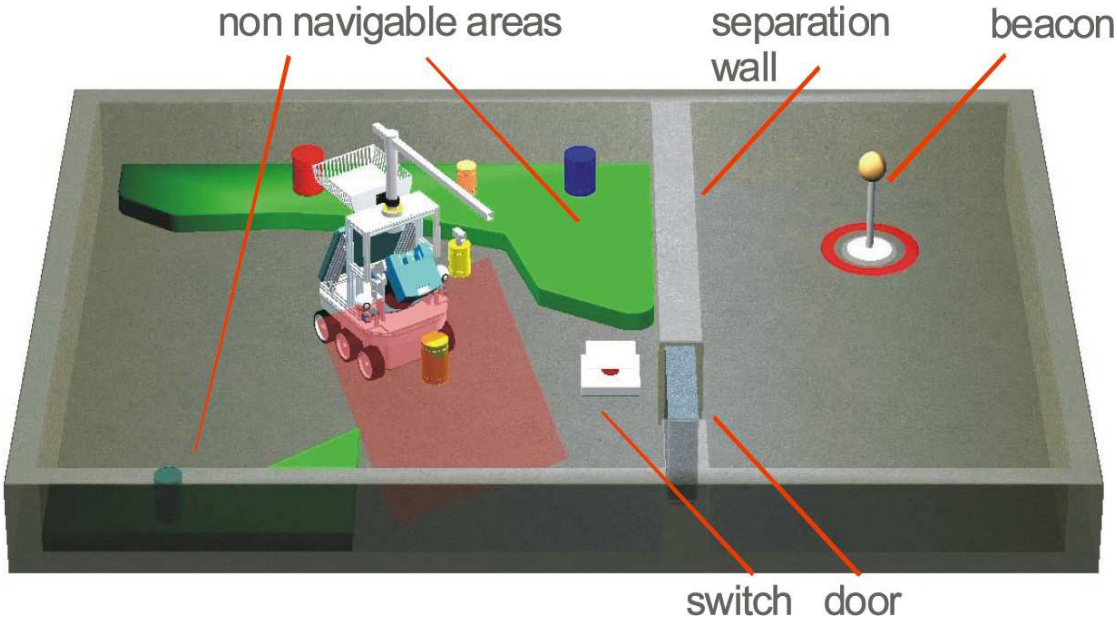


Figure 3.3: MACS final demonstrator scenario (from Breithaupt and Rome, 2005).

The final demonstrator scenario that is about to be demonstrated at the end of the MACS project in 2007 is depicted as a concept art in figure 3.3. The robot that is shown in this picture is a Kurt3D robot that will be described in more detail in section 5.1. At this point it should suffice to say that it is wheel-driven robot that can turn on the spot, is equipped with some internal sensors, for instance, to measure its speed, a pitchable laser scanner for acquiring 3D distance data and two separately moveable cameras. See as well Surmann et al. (2003) for a description of the system and its capabilities. In this setup, the robot is furthermore equipped with a crane device with an electro magnet attached to it. While the architecture developed in the MACS project will be employed on different robot platforms as well, for instance a Pioneer robot that is used by the project partners of Linköping Universitet in Sweden, the focus of this thesis will lie on the implementation for the Kurt3D robot. This does neither diminish the expressiveness of the demonstration nor of the system's power because the robot's actual characteristics are not of very much importance when it comes to demonstrate the applicability and usefulness of an affordance-based architecture.

Now to describing the actual task that the robot is meant to accomplish and that is defined in Breithaupt et al. (2005); Breithaupt and Rome (2005): Figure 3.3 shows different test objects placed in a small scenario that the robot can interact with.[2] These test objects are different in their size, color, weight, and form as well as in their property

[2]Note that the environmental objects, the robot can interact with are in the following called test objects, to avoid any possible confusions with the object-oriented programming paradigm and to stick with the terminology used in the MACS project.

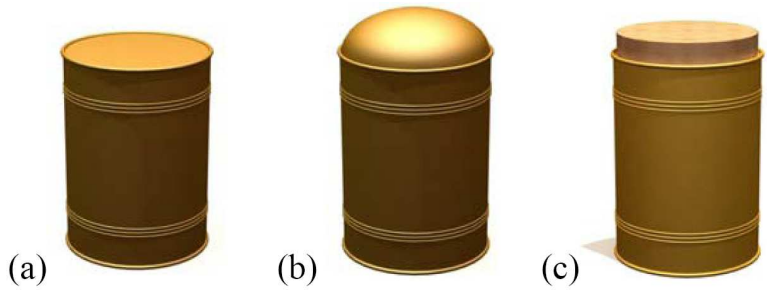


Figure 3.4: Different tops of standard test objects. The flat top (a) represents a test object liftable for the robot, objects with the spherical top (b) are hardly liftable and (c) shows an isolated top that is not magnetic and therefore not liftable by the robot (from Breithaupt and Rome, 2005).

of being either magnetic or not (this property is also encoded in their form, see figure 3.4). As the robot is equipped with a crane that uses an electro magnet to lift up objects, it can obviously only lift magnetic objects and of those only objects below a certain weight. While the crane is the main manipulator of the robot it can otherwise only drive around and hence push objects, that are light enough.

The test objects distinguish themselves in a meaningful way as, for instance, only blue cans with spherical tops are magnetic and thus liftable. They are designed in this way, to offer feature types to the robot that it can perceive with the sensors provided. Otherwise the robot would of course not be able to reason about these features when it comes to perception and learning of affordances.

Assume a scenario that holds blue and red cans. All blue cans have a flat top, are magnetic and lightweight and can therefore be lifted or pushed while all red cans have an isolating top, are hence not magnetic but still they are light enough to be pushed. A robot that explores this special environment and applies different kinds of actions, like pushing or lifting, on the different test objects should, given the affordance-based MACS architecture, be able to learn these relations between perceivable, environmental features and the actions it can apply on them. In other words, it learns the affordances that those test objects offer to this particular robot. Speaking in terms of the affordance representation introduced in the last section, the robot would learn, that the directly perceivable cue of a nearby blue object affords both the pushability and liftability of that object. The action stored in these two affordance representations would therefore be push or lift, with their corresponding parameters that made the application of these actions successful in the past. The outcome descriptor, as the third integral part of the affordance representation, would be the percept of a lifted or spatially moved object.

To think this concept further, the robot might have learned from exploration that blue features in the environment afford both pushing and lifting. If it is now confronted with the red pushable, but not liftable cans it has to refine its representation of the affordances which it has learned previously. The robot should adapt its affordance representation in a way, that the affordance of pushable contains in its cue descriptor the valid color value

range from red to blue, while the lift affordance's cue descriptor only contains the color range that covers the impression of blue cans. Another way would of course be to shift the representation of these affordance away from the color to the magnetic properties of the objects if they can be perceived, e.g. by the form of their tops.

A system that is capable of accomplishing this task, can indeed autonomously acquire a function-centered view on perception. It should be able to adapt flexibly to environmental changes by refining its affordance representations. It should be able to interact with unknown objects in an intuitive way abstracting from previously learned knowledge. And furthermore, given an appropriate deliberation module it can use its knowledge about affordances to develop plans that yield the desired outcomes.

Of course, the scenario of simply pushing and lifting colored cans is not very interesting and the demonstrator scenario developed within the framework of the MACS project pays tribute to this by defining some tasks that exploit the robot's understanding of affordances a bit further.

As one might already have noticed, the scenario depicted in figure 3.3 contains a door, a switch and a beacon. The task of the robot is to autonomously figure out, that a large enough weight placed on the switch opens the door and allows it to drive to the beacon. From an affordance point of view, this would for instance include affordance representations that code for the functions of carrying objects around, putting them on the switch and driving through opened doors. The robot has to learn the affordances provided as the meaningful relations between the percepts available in its surroundings and the actions it can apply on them. If it succeeds in generating these representations it will be able to succeed in figuring out how to open that door in order to reach that beacon.

Another task defined in the MACS project is a stacking task. Therefore, the different test objects have different diameters and different properties, e.g. beaked or flat tops which either afford placing another test object on them or not.

To summarize this section, the basic applicability of the approach can already be shown by forming representations of easy affordances like pushability and liftability. The scenario has nevertheless been increased in its complexity to show the real power of the approach by allowing the meaningful actions of carrying, dropping or stacking in order to be able to act in a clearly presentable goal-directed manner.

The demonstrator scenarios have been introduced in order to provide a general idea of what the actual system is meant to do and what problems it should be able to solve. As this thesis will describe the development of an affordance-based task execution system it is highly dependable on the specific needs and the application domain of the robot as it has to especially support the different actions, which the robot should be able to perform.

The next chapter will therefore define the affordance relation to robotic task execution and show how the concept can be integrated into that component in a supporting and meaningful way. Afterwards, chapter 4 will present the design for the execution control that is based on the defined concepts followed by the actual implementation in chapter 5.

4 Concept and Design of an Affordance-based Task Execution Component

During the last chapters, it should have become clear, what can be understood as an affordance and how this concept can possibly contribute to nowadays robotics. In the very last section the MACS project has been shortly introduced as a project that precisely tackles the question whether the affordance concept is suitable and appropriate as the underlying principle for a complete robot control architecture.

It is the core of this thesis, and actually of this chapter, to define a general concept for affordance-based task execution as a part of the MACS hybrid robot control architecture. Consequently, it will be designed in a way that it explicitly supports and augments the other components of such an architecture like e.g., learning, planning, and execution control[1].

Hereby, it is important to always see the work introduced here as being embedded in the MACS project. Of course, there are plenty of other approaches of robotic task execution and many of them perform very well. Even within the MACS project some of the other partners tried different approaches for task execution than the one presented in this chapter. Nevertheless, the approach that will now be introduced bears some advantages in expressiveness when compared to other execution components, especially when dealing with an affordance-based architecture.

This will become clear by explaining the component's actual design in section 4.2 and comparing it to other well-known approaches in section 4.3. But before the actual functionality of the developed task execution component will be elaborated, the following section will list the different requirements put on this component by both the physical constraints and demands of the robot platform and the demonstrator scenarios, and as well by the conceptual constraints of the architecture and the MACS project as such.

4.1 Requirements

The standard control flow in a hybrid architecture starts with some goal, that is either given a priori or specified by an operator. A deliberative component will then generate a plan, based on the robot's world knowledge, to achieve that goal. The plan is provided to an execution control component that splits it up into small tasks, sequences them

[1] Note that it is important here not to mix up the two decisively different components of the general *execution* module and the *task execution* module (see section 3.2.2).

in time and space, and provides them bit by bit to a task execution component for the actual execution.

Assume the MACS demonstrator scenario mentioned in the last section of accomplishing stacking tasks. In this example the control flow would start, for instance, with the goal of a blue can on top of a red switch. A plan for accomplishing that goal could hereby be a sequence of subtasks or actions like: Find a blue can, approach it, lift it up, find a red switch, approach it, put the can on the switch. This plan already requires a fair amount of functionality that a robot that is meant to execute the plan has to offer.

Generally speaking, a task execution component that should offer this or similar functionality can be interpreted as having to fulfil the following list of *physical requirements*:

1. The robot shall be able to move smoothly without invoking the impression of jerky movements.
2. It shall be able to react to perceived static obstacles and to adjust its movement accordingly.
3. It shall be able to react to perceived dynamic changes like suddenly appearing obstacles and again to adjust its movements accordingly.
4. It shall be able to perform goal-directed navigation.
5. It shall be able to perform purposeful manipulation actions.

The first three requirements can be grouped under the main target of the robot being able to move smoothly through its surroundings while avoiding both static and dynamic obstacles. One should note here that it is the topic of this thesis to develop a task execution component. Other requirements that are connected to other parts of the architecture, like the perception or deliberation module, will therefore not be discussed here and will only be regarded peripheral in chapter 5 as far as they influence the actual implementation of the component.

The latter two items of requirements 4 and 5, however, focus on the system's capability to perform purposeful actions. To stay with the example of the stacking scenario, the robot should be able to interact with the objects in its surroundings, e.g. to lift, carry, and stack them as well as to navigate to meaningful locations, preferably while avoiding obstacles that are blocking its way.

These requirements are very similar to those requirements of other, non affordance-based architectures. They mainly describe the demands put on the physical functionality and capabilities of the robot as they target smooth and secure navigation as well as the robot's ability to accomplish those manipulation operations it was built for.

Nevertheless, there arise as well some conceptual demands that should be considered when designing such a task execution component, especially when it comes to introduce affordances into robot task execution. In general, and in the context of an affordance-based architecture, the component should thus tackle the following *conceptual requirements*:

1. The component shall be able to deal with the actual level of complexity of the assigned tasks.
2. It shall provide the means of anchoring the affordance representation on the task execution level.
3. It shall provide an explicitly encapsuled response point for the execution-related part of an affordance.

The first requirement is again valid for all types of architectures. It simply states that the component should be interfaced in the right way with the rest of the architecture to accomplish those tasks that are assigned to it. It is mainly the question of defining which parts of the actual task execution are handled by the task execution component itself or have to be broken down and scheduled further by its superordinate component of the general execution control.

In the case of the MACS scenarios the level of complexity that can be handled directly within the task execution component will be as meaningful as, for instance, the accomplishment of a searching task. The task of searching for a blue can that shall be put on the red switch would therefore be regarded as ranging in the level of complexity handled within the task execution component. The reasons for this decision will be introduced later on in section 4.2.2.

The conceptual requirements 2 and 3, however, are specific to the requirements of an affordance-based robot control architecture in general and of the special target of the MACS project in particular. They demand that the affordance concept shall serve as the underlying principle of the design of this task execution component. In other words, the component shall provide a direct response point to the affordance representation in order to anchor or ground it in the architectural most basic layer of the task execution component. It is believed here that an explicit executional response point to affordances may serve as the necessary means to verify the usefulness of affordance theory in robotics as will be elaborated in the following.

Based on these specified requirements to both the physical and conceptual demands on the final design of an affordance-based task execution component, the following section will now introduce such a component that exactly addresses these requirements. While its actual implementation and performance will be topic in chapter 5, the last section here will compare the approach to other existing architectures.

4.2 The Task Execution Component

Figure 4.1 depicts the architectural component of a task execution module as it has been designed with respect to the requirements that have just been formulated. The following section will explain its structure and functionality while section 4.2.2 will explain in detail the concepts and definitions developed.

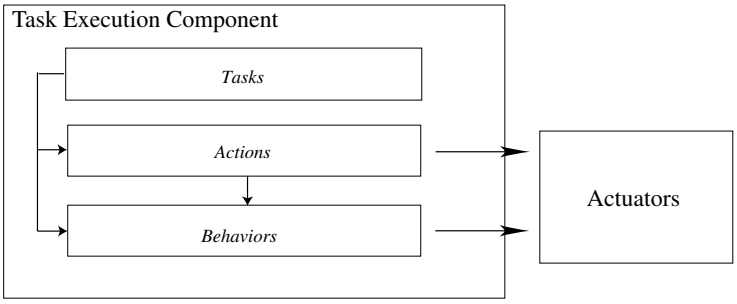


Figure 4.1: Task Execution Component. The component's elements are structured in a hierarchical manner where higher levels can make use of lower levels. The actuators may only be accessed by the task execution component's actions and behaviors.

4.2.1 Structure and Functionality

The developed task execution component is structured in a hierarchical manner. Hereby it defines the three concepts of *behaviors*, *actions*, and *tasks* that distinguish themselves by the level of complexity they are designed for.

The lowest level of the behaviors does hereby have the functionality of performing computationally cheap movement control. By applying a fast and tight coupling between action and perception, the component's behaviors will tackle the physical requirements 1 to 3, i.e. they will provide smooth movement capabilities to the robot and take care of performing obstacle avoidance. They are designed with a level of complexity which enables them, for instance, to brake in front of an obstacle or to turn around in a corner. They do explicitly not cover any higher levels of navigation and are restricted to influence the robot's movement but not its actuators.

On the layer above the behaviors is the concept of actions. Actions are those functional units that stand for the meaningful and goal-directed accomplishment of doings beyond the abilities of simple behaviors. In contrast to many other approaches (see section 4.3), actions are defined here to include all necessary capabilities to really execute that action. With respect to the aforementioned stacking scenario, a *lift*-action, for instance, would include the necessary control to move the robot's crane above the test object, lower the magnet, orient it correctly to lift it up, switch on the magnet and finally to lift the test object. In the following section, it will thus be argued that an action hereby becomes a representational powerful, clearly differentiated unit that already allows a somewhat high level of interpretation and thus eases its use in the superordinate architectural levels. It will furthermore be argued that actions, especially object manipulation actions, provide the underlying anchor point for the affordance representation as it is used in the MACS project by grounding the representation in this architectural layer of the task execution component. Actions therefore meet the demands of the physical requirements 4 and 5 and furthermore the conceptual requirements 2 and 3. Hereby, they become the most important part of the newly defined task execution component.

The third and final layer, the concept of tasks, addresses the conceptual requirement

1. As it has already been mentioned in the description of this point, tasks are designed to interface with the rest of the architecture in a way abstract enough to simplify the execution of meaningful small tasks like searching for objects or exploring an area. A task distinguishes itself from the higher levels of the execution control and deliberation as it does not include any complex scheduling or replanning mechanisms. With regard to the affordance concept, tasks have an additional functionality as they can nicely be used for active perception purposes because they form the connection between higher-level goal-oriented plans and the low-level actuator control.

While this short introduction should have established a basic understanding of the task execution component's trichotomy, all three layers of behaviors, actions, and tasks will now be defined and described in more detail before they are compared to other existing approaches.

4.2.2 Concept and Design

This section will introduce the formal definitions of the three layers of the task execution component and explain the rationale behind these concepts. Each of the definitions will furthermore provide the list of different instantiations that are or can be used to implement the necessary functionality of this component in the MACS architecture.

Behaviors With respect to the three elements of the task execution component, the notion of a *behavior* is probably the most ambiguously used one throughout the literature. Speaking of robot control, the term behavior can refer to arbitrary levels of abstraction or complexity. Nevertheless, the main consensus is that behaviors are relatively simple, independent, and parallel control procedures (cf. Matarić, 1999).

In the context of this thesis the notion of behaviors will be interpreted adapted from the literal meaning of the word as defined in Merriam-Webster (2006):

Def. 3 (Behavior)
A behavior is anything that an agent does that involves a direct response to stimulation. Hereby, behaviors

1. *work in an automatic fashion,*
2. *are not explicitly goal-directed,*
3. *may maintain an internal state.*

Thus, the term behavior, when used in the context of this thesis, will refer to a stimulus-driven response of the robot to its current environment. Behaviors are regarded as automatic processes influencing the robot's doings but do not have the quality of a goal-directed, purposeful action.

This definition is highly influenced by the work of Brooks (1986) who has developed one of the best-known approaches to reactive robotics - the subsumption architecture. In his work, Brooks defined several controller-like processes that were run in parallel in

order to steer the robot. One of his key ideas was to couple the robot's perception and action tightly to each other and to omit any kind of reasoning or representation (Brooks, 1991a,b,c), in other well-known words, they "use the world as its own model" (Brooks, 1991b, p. 139).

The processes Brooks has defined have a complexity like, for instance, staying in the middle of a corridor by monitoring the robot's distances to the walls, or simply to approach a light source. These processes distinguish themselves from the definition of behaviors provided in this context mainly by the fact that behaviors are allowed to have an internal state, which, strictly speaking, violates Brooks's underlying principle of avoiding any kind of representation. Nevertheless, behavior-based systems in general are based on the ideas and definitions of Brooks's subsumption architecture (cf. Matarić, 1999).

Recalling the requirements specified above, the behavioral layer of this task execution component is mainly meant to yield smooth robot motion and safe and robust obstacle avoidance capabilities. This aspect can already be reached by specifying only the three behaviors of *Brake*, *Steer*, and *Turn*. The key principle of a behavior system is that behaviors apply a computationally cheap, and therefore very fast, action-perception coupling and that they may be run in parallel in order to achieve coherent, *emergent* behavior. In this sense, the three just mentioned behaviors have the common attribute of running very fast on the current sensor input. For instance, a laser scanner mounted on the robot may deliver a new distance reading to nearby objects every 13ms. The different behaviors may then take this reading and compute each a quick response. The different behavior responses are then integrated, for instance, in a hierarchical priority manner as in Brooks's subsumption architecture, and yield a motor output that will influence or determine the following motion of the robot. For example, a brake behavior might slow the robot down, if an obstacle is sensed in driving direction. Hereby, the different behaviors normally run that fast that they are rather limited by the update time of the applied sensor than by the computational effort they produce.

The three behaviors defined here, that yield in their combination to a robust emergent wandering behavior with obstacle avoidance, have the following different characteristics:

Brake - The *Brake* behavior is the most important behavior when regarding the systems security since it takes care of slowing the robot down in the proximity of any objects or obstacles that the robot would otherwise hit.

Steer - The *Steer* behavior influences the robot's current heading in a way that it is repelled by obstacles. More precisely, it evaluates the current sensor reading by determining a direction with only few obstacles that is close to the actual driving direction and biases the robot to drive into that direction.

Turn - The *Turn* behavior turns the robot on the spot when its way is blocked, i.e. when the *Brake* behavior has stopped the robot.

This simple set of behaviors is sufficient for getting an already quite complex system behavior. A robot that concurrently executes all three behaviors will, given that the

behaviors receive the input of a robot set speed (see section 5.2), drive around in its surroundings by orienting itself into a free perceived driving direction (*Steer*), slowing down if obstacles block its way (*Brake*) and turn into a new direction if it did come to a halt because its way was blocked (*Turn*). The overall impression of the robot's movements would therefore be that of an emergent *roaming* behavior.

While the definition of behaviors provided here will be compared to other approaches found in the literature in section 4.3, their implementation will be topic in the next chapter.

Actions Actions represent the second and certainly most important layer within the task execution component since actions are those elements that directly embody the relation of this component to the concept of affordances.

The applied definition of actions is again close to the literal meaning of the word (see Merriam-Webster, 2006):

Def. 4 (Action)
Actions have the following characteristics:

1. *An action is the self-contained, goal-directed, and willed accomplishment of a thing usually over a period of time, in stages, or with the possibility of repetition.*
2. *Actions have preconditions and outcomes/effects.*
3. *Actions might be afforded by an agent-environment relation.*

According to this definition, actions are on a higher level than behaviors as they represent something that is done on purpose. They are consciously willed doings chosen and conducted by the robot with the intention of achieving a goal. Consequently, actions are not continuously running processes. They are instead explicitly triggered and end with a certain outcome or effect unless they fail. Actions are thus timely constrained, purposeful, repeatable, and self-contained doings. Hereby, the property of being self-contained implies that an action includes all necessary functionalities needed for its execution as has already been stated when introducing the requirements put upon this task execution component. An action is thus a functionally closed unit including, for instance, all control procedures for lifting up an object.

With regard to the definition point 2, that is concerning the outcome of actions, note that actions are not guaranteed to bring about the desired outcome. They may fail as the agent or robot may fail in applying them. For instance this can be the case, if the robot misjudged the validity of the action's preconditions since the perceptional processes that form the robot's belief of the current state of the world is only a subjective interpretation by the robot itself. It is by no means assured that this interpretation is correct and really reflects the current situation. Moreover, the robot may just fail in the execution of an action, for example, it may loose grip of an object while trying to lift it. The resulting outcome is hence not guaranteed to be a lifted object.

Concerning the third point of the definition, it follows that since actions are regarded as explicitly triggered and willed doings, it is to assume that actions are accomplished based

on some reasoning about the agent's possibilities in its current surroundings. This shows the direct relation between the concept of an action and the concept of an affordance. In fact, actions are explicitly what affordances afford.

In the definition of an affordance (see section 3.2.2) it was stated that an agent has to be able to act upon the existence and perception of an affordance. For instance, if a rock affords throwing, the goal-directed, repeatable, timely constrained action that is connected to the affordance is: *throw*.

Nevertheless, the third point states explicitly that an action only *might* be afforded. This is due to the fact that an affordance must not be mistaken to always be present when the action's preconditions are fulfilled. This has already been stated based on the example of a cat in an empty room (see section 2.1). That cat is definitely able to accomplish the action of approaching each and every location in that room. Nevertheless, the location that really affords approaching is only that, where the mouse tries to hide. This also pays tribute to the fact, that actions are only afforded if they are meaningful to the agent.

Moreover, speaking in terms of the MACS scenarios the robot is assumed to perform object manipulation tasks. These tasks include purposeful actions like pushing, lifting, or even stacking objects. Actions of this kind are much more likely to be subject to an affordance than general actions. This is because e.g. the affordances of liftable objects are presumably more meaningful and inviting to the robot than just the approachability of a random location.

Exactly this characteristic is one of the main reasons for defining this explicit layer of actions in the task execution component and shows the distinction drawn here between actions and behaviors.

When introducing the notions of behaviors above, it was stated that the behavioral layer is targeted solely at performing stimulus response movement control, mainly for the purpose of avoiding obstacles. Processes that control the robot's manipulator are explicitly not part of that behavioral layer since all necessary control and coordination for applying an action is interpreted as being an integral part of that particular action; in other words, an action is a self-contained independent unit. At this point actions are considered as coordinated, willed doings that are certainly on a higher level then the mere movement control of behaviors. It is no violation of this claim if the robot can react to its perception in a way that it can, for instance, orient a magnet spatially over a metal bucket in order to accomplish the action of lifting it up. Such a capability is instead classified as an inertial part of an action since the actual movements, which the robot conducts in order to fulfil the action, can certainly be interpreted as being the action's main ingredients. In other words, they are regarded as being purposeful parts of an action rather than some abstract simple and reactive behaviors running continuously and without deliberation.

Applying this definition of actions to the robot capabilities demanded in the MACS scenarios, the following list of actions can be identified:

Approach - As demanded in the list of physical requirements in section 4.1, the *Approach*

action is meant to perform goal-directed navigation. This action intuitively brings out the reasonable feature of behaviors that they can control the underlying execution layer of behaviors. An *Approach* action that can trigger and inhibit the specified behaviors may easily perform obstacle avoidance while approaching a destination but is also able to approach it closer as it would be allowed by the *Brake* behavior if the action can inhibit the behavior's output.

Push - The *Push* action should push an object that is lying in front of the robot forward but should break up its execution if the object is too heavy to be pushed.

Lift - The *Lift* action shall be able to access the robot's manipulator, in this case the crane, in order to lift a test object lying in front of the robot. Since actions are defined as being self-contained units, it should be able to perform all necessary steps of: orienting the crane above the object, lowering the magnet, making contact with the test object that should be lifted, and finally trying to lift it.

This action is a good example of an action that may easily fail since the test object might, for instance, be too heavy to be lifted, may not be magnetic, or simply out of reach.

Drop - The *Drop* action is meant to place lifted objects back on the floor.

Stack - The *Stack* action is similar to the *Drop* action but needs more powerful means to control the place where the lifted test object should be put.

Carry - The *Carry* action may use the *Approach* action to move to a target destination and combines this action with the functionality of holding a lifted object.

This action is an example of how one action might extend another action for accomplishing its assignment.

This set provides all necessary functionality required for executing the different actions faced in the MACS scenarios. But at this point it remains to highlight why the differentiation between the two elements of behaviors and actions is made that explicit in this context of developing an affordance-based task execution component.

The important point is that the introduced notion of actions explicitly determines an affordance-based concept that is addressable by other parts of the architecture.

In the context of defining this task execution component, the actual affordance representation triple introduced in section 3.2.2 should be reformulated by replacing the part of the *behavior descriptor* with the notion of an *action descriptor* yielding an affordance representation of:

$$(\textit{outcome descriptor},\ \textit{cue descriptor},\ \textit{action descriptor}).$$

Since an action is a sound self-contained procedure it can serve as the direct response point on the architectural task execution layer as it has been demanded in the list of conceptual requirements in section 4.1. It thus anchors the affordance concept on the

layer following the postulation of introducing affordances as first-class citizens to the architecture.

This leaves us with the question of why we do not only specify everything that is part of the task execution layer as being affordance related. After all, one could easily argue that behaviors, even in the form as they are defined here, have an affordance relation as well.

Of course, this is true. A behavior that takes care of driving around an obstacle in a corridor can indeed be interpreted as appreciating and reacting to the affordance of the obstacle, which in this case affords the reaction: *avoid*. In fact, some experiments, which show the basic applicability of affordance theory for mere navigation, have been conducted at the very beginning of the MACS project by Doherty et al. (2005). They have shown the affordance-based generation of traversability and standability maps, which reflect the robot's belief whether certain areas afford the actions of traversing or standing in them.

Nevertheless, the applied distinction becomes clear when regarding the overall target of the MACS project. The project aims at demonstrating not only that affordances *can* be used in a robotic control architecture, but in addition that it is supportive, meaningful, and advisable to integrate the affordance concept as a first-class citizen. But an approach in which a robot avoids an obstacle because the robot perceives the corresponding affordance cannot be differentiated objectively from standard potential field approaches that have been manifoldly demonstrated in the past. This is, in the scope of this thesis, interpreted as demanding a focus on what capabilities environmental objects afford for *interaction* rather than merely for *reaction*.

Hence it is believed that the task of purposeful object manipulation requires a different level of understanding by the robot. It thus qualifies better for demonstrating deliberation, understanding, and usage of affordances within an architecture. If a robot is able to perceive the functionalities afforded by objects, or in other words the concrete *actions* it can perform on them, and if it is able to evaluate on that perception and its knowledge, it will presumably as well be able to demonstrate some of those benefits, which have been extensively presented in section 3.1. Examples would include, for instance, the selection of objects according to their functionality rather than to a labelled category, i.e. something like putting a heavy object on a switch to open a door instead of explicitly searching for the one object that is labelled door-opening-weight.

To summarize the reasons for drawing the distinction between behaviors and actions from the affordance point of view it can be stated that simple and reactive behaviors can be related to affordances, but actions are where the concept becomes interesting.

Subsuming this paragraph, actions have been defined as purposeful, goal-oriented doings. Due to their self-contained nature, they provide an explicit response point to affordances, grounding them on the hardware-close level of the concrete task execution.

This leaves us to explain the third and last integral layer of the developed task execution component.

Tasks The third and final group of task execution elements, which holds the highest level within the component, is referred to as *tasks*.

Tasks are defined to tackle the demand of an affordance-based task execution component for a well designed integration and interfacing with its superordinate execution component (see the conceptual requirements of section 4.1).

The notion of tasks is defined again close to the literal meaning of the word (Merriam-Webster, 2006):

Def. 5 (Task)
A task is an assigned piece of work to be finished in a certain amount of time. Hereby, tasks

1. *are meaningfully capsuled modules, allowing the easy utilization for solving simple problems,*
2. *are able to perform low-level scheduling,*
3. *can react in a basic form to environmental constraints.*

As aforementioned, the execution component or scheduler of the architecture normally breaks down a plan into small pieces of work that are assigned to the appropriate elements in the task execution module. These pieces of work are seen, in this context, either as the explicit call of an action to be fulfilled or, on the other hand, of a small task for which it is advantageous to be accomplished within the actual task execution component.

Bonasso et al. (1997) formulate this claim by stating that it is beneficial to raise the level of abstraction from basic behaviors and actions to a higher-level that augments the mapping between plan operators and the actual instantiations of these operators. While they use this argument to motivate the architectural layer of a sequencer for mediating between a planning component and a task execution component, the argument does as well hold for the interface between an execution component and a task execution component as they are defined in the MACS architecture.

The concept of actions that has just been defined already aims at simplifying the work of the execution component since it introduces actions as being self-contained and thus as being able themselves of performing adjustments and movements that would arise in other systems from complex inter-behavior dynamics. This simplification shifts a lot of the complex work-load, an execution component would otherwise face, to the task execution layer. This step is deemed useful because actions can be implemented and designed by hand and tailored to the actual robot without too much effort.

The definition of tasks is a similar step and aims at an even slightly higher level of abstraction. Tasks are meant, as stated in the definition, as simple pieces of work that are timely constrained and are only capable of performing some low-level scheduling or plan adjustments arising mainly from the immediate sensory perception of the robot.

The list of tasks that is deemed appropriate in terms of the MACS task execution component contains the following:

Explore - The *Explore* task is a simple sequence that already shows the full complexity that can be handled within this layer of the task execution component. An *Explore* task will use the already specified set of simple behaviors in order to roam the environment. Hereby, the robot will identify interesting test objects and try to apply actions on them. Both, the cues to look for as well as the actions to apply should hereby be specified in advance since the task execution component is not the place to do deliberation of what action might be appropriate or interesting to try out.

Search - The *Search* task is the other task that has its own representation within the task execution component. This task is meant to perform active perception within the robot's surroundings (see section 3.1.5). Active perception is actually a task that is predestined for bridging the gap between a higher level execution control component and the actuator accessing task execution component since the purposeful search for objects is a fairly high level goal that can become arbitrarily complex while it is very useful to be able to directly access the robot's actuators when searching for the desired cues. This is especially true when facing a robot that has moveable sensors which can be actively aligned or, for instance, panned.

Again, the cues to look for should be specified in advance.

A task will therefore internally sequence the necessary actions and behaviors that are needed to accomplish its goal but it has to be noted that the scheduling of complex tasks, like for instance object delivery or moving a whole stack of objects to a different location, is taken charge of by the execution component's scheduler.

The definition of such tasks actually provides a nicely shaped class of abstraction that is very well to handle on the task execution layer. The benefit of being able to assign simple tasks to the execution is mainly, that the scheduling execution component does not have to work with the lowest levels of autonomous robot movements, i.e. the behaviors. The plan sequences established become thus better structured and from an implementation point of view easier to implement, debug, and use.

With the three notions of *behaviors*, *actions*, and *tasks*, this section has defined the concepts that an affordance-based task execution component can be built on. It has been stated that especially the concept of actions shows a clear-cut and explicit response point to the affordance concept that can be accessed directly by means of the affordance representation triples. The comparison to other robot architectures will show further that the self-contained nature of an action, enclosing all necessary functionality in one procedural unit, forms the core of this task execution component's affordance relation.

Besides the concept of actions, behaviors will assure smooth and collision-free navigation capabilities while tasks provide an easy to use interface to superordinate architectural components and might show to offer particulary beneficial aid in active perception tasks.

The following section will now compare the developed concepts to existing approaches of other well-known architectures before the next chapter will eventually describe the

implementation of those behaviors, actions, and tasks that have been defined during this chapter.

4.3 Comparison to other Approaches

During the last section the different concepts of *behaviors*, *actions*, and *tasks* have been introduced that form the structure of this task execution component. While it is already clear which functionality these different layers provide to the system the remainder of this chapter will now compare the elements of this newly developed component to other robot control approaches.

4.3.1 Behaviors

Taking the notion of behaviors as being reactive and situated processes on its own is, nothing very special. Designs like this have already been applied in the earliest work of reactive robots, e.g. by (Brooks, 1986) himself, though he even omitted internal states.

His idea was to completely avoid the processes of the computationally expensive modelling and planning needed in purely deliberative approaches and instead to design systems, which perform very low-level perception of the environment and react immediately to perceived stimuli. His theory was to design such low-level processes that will lead, in a bottom-up way of intelligence engineering, to real complex intelligent systems when they are combined in the right way. Brooks (1991a) motivated this view nicely with referring to the time span evolution needed to come up with reactive behaviors in animals compared to the short time for intelligence to emerge after the reactive key components where actually available.

As aforementioned, Brooks already defined several controller-like processes that were run in parallel like for instance slowing down if an obstacle occurs in front of the robot or staying in the middle of a corridor by keeping the same distance to both walls. These simple control processes are indeed very similar to the set of behaviors formulated here. In combination they explicitly showed their appropriateness for performing smooth and secure robot movement control, a matter of fact, which may best be described in the words of Erann Gat (meanwhile known as Ron Garret), who said that, after Brooks, robots were able of "zipping around the lab like R2D2" (Gat, 1998, p. 197).

However, the point to stress here, in the context of designing an affordance-based task execution, is that the behaviors defined above are not meant to be more than what has just been stated there. Their only assignment is to perform reactive actuator control aiming at smooth movements and the ability to react fast and robustly to dynamic changes. Hereby, their internal state should not significantly exceed the expressive power of an average filter over past outputs. The defined class of behaviors is meant for ensuring the safety of the system by avoiding both static and dynamic obstacles. Most importantly, behaviors are not explicitly goal-directed, since they represent action on a level too low and normally too fast to imply the amount of reasoning necessary for accomplishing goal-directed actions. Instead, they are control processes that may run in parallel and

continuously, or in other words *automatic*. Their influence on the overall actions of the robot is, nevertheless, limited to ensure collision-free navigation and nothing else.

This is indeed a major distinction to many other current systems. Behaviors are oftentimes used as the only task execution layer of a control architecture. This puts the demand on a behavior system that it has to be able to solve more complex, or higher-level tasks like, for instance, goal-directed object manipulation. Staying with this example, those architectures try to solve that problem by designing so-called *high-level* behaviors that coordinate sets of *low-level* behaviors in order to achieve the emergence of a goal-directed object manipulation. Following the main concept of designing behaviors as simple control procedures, low-level behaviors are on a level of performing movements like "extend arm" or "position wrist" (e.g. Stoytchev, 2005a). They are normally situated and reactive. High-level behaviors, on the other hand, are interpreted in a completely different way. Normally, they do not have direct access to the motor outputs of a robot but instead face the task of *behavior arbitration*, i.e. they select the low-level behaviors that will lead in combination to the desired goal while *inhibiting* the others. These kind of behaviors, that Murphy (2000) even calls *conscious behaviors*, require a fair amount of deliberation. Systems, which apply this form of behavioral layering, already fall into the paradigm of hybrid robot architectures since they exceed the point of, what Gat (1998) calls, the "capability ceiling" (p. 197) of reactive robotic systems.

In the context of this task execution component, however, behaviors will be viewed as being simple processes whose output is mainly determined by their currently perceived stimulus. They are not consciously goal-directed but take direct influence on the robot's movement adjusting it according to their nature. Everything that has the character of being purposeful, goal-directed or *willed* instead of automatic is assigned to the second element of the defined task execution component, namely to the concept of *actions*. The difference to other approaches that also use behaviors in their executional layer will thus become even clearer in the next section when discussing the concept of actions.

4.3.2 Actions and Tasks

The different approaches to task execution come in a wide variety. In deliberative approaches, a planning system generates a plan normally holding a sequence of operators that are believed to lead to a desired goal. They employ an execution component that instantiates these operators in the form of accomplishing the corresponding actions.

The relation of the task execution component developed in this thesis to deliberative architectures is quite straightforward since the presented definition of actions is held at a level abstract enough to allow to be directly related to such plan operators. Actions have meanings like *push*, *lift*, or *stack* rather than being primitive low-level movements that cannot be directly related to the application of a purposeful action.

This actually constitutes the most supportive feature of this task execution component from the view of an affordance-based hybrid architecture, which becomes clear when depicting the way most behavior-based approaches, whether they are reactive or hybrid systems, handle task and action execution.

Most of these approaches work on the basis that they combine different low-level

behaviors and run them in parallel in order to achieve more complex or goal-oriented emergent system behavior than offered by a simple and primitive behavior alone. The idea behind these approaches is to formalize a set of these behaviors that are understood as the fundamental building blocks of intelligence and that they only have to be combined in the right way to deliver real intelligent systems.

Nevertheless, the next few examples of other approaches will show, that the underlying concept, definition, and accomplishment of actions is usually handled in a less explicit and more blurred manner than defined in this thesis. This is interpreted here as a loss of expressiveness that serves actually nicely as the clear-cut anchoring point of affordances within the layer of task execution.

Subsumption Architecture In terms of Brooks's subsumption architecture (Brooks, 1986), being a placeholder for successful reactive control architectures in this context, the selection of behaviors, i.e. the behavior arbitration, as well as the dynamics between different behaviors, i.e. how they influence each other, are structured hierarchically. The higher-level behaviors of this architecture are able to suppress the output of lower levels according to their needs. Nevertheless, they do not simply ignore their capabilities but instead subsume their functionality, hence subsumption architecture.

To give an example, a robot that shall grasp an object might have on the lowest level of its architecture behaviors, that prevent its gripper from colliding with objects in its environment. Since such a behavior would impede the process of actually approaching an object close enough to pick it up, a grasp behavior would be localized on a higher level in the architecture and thus be able to suppress the low-level collision avoidance behavior in order to approach the target closer. Nevertheless, the subsumption architecture has been criticized for lacking a clearly defined methodology for adding higher level behaviors (see e.g. Gat, 1998). In particular, the ordering of behaviors into the different layers is normally quite arbitrary and leads to artificial structures since it is not always clear how to order the behaviors as there are cases when a lower-level behavior should actually influence a higher-level behavior. Gat (1998) refers furthermore to Hartley and Pipitone (1991) who have found that different behaviors are not independent of each other. This leads to a design and implementation process with a level of complexity that increases with the number of defined behaviors. It also implies that changes on some of the low-level behaviors cause necessary adjustments throughout the whole architecture.

In other words, subsumption architecture tries to reach complex system behavior by carefully designing a dynamic system of interconnected low-level behaviors. The concrete actions a subsumption-based system is able to fulfil emerge as a result from these overall inter-behavior dynamics. Nevertheless, as has been stated above, these system tend to get more and more interwoven and hard to maintain if new actions and new levels of complexity are added.

This way of combining multiple processes for achieving the capability of executing complex actions, is in fact handled very similar in many behavior-based hybrid approaches with the main difference being that the hybrid approaches use internal world models and support higher-level deliberation and behavior arbitration mechanisms.

The middle layer of these architectures, holding an execution component as in the MACS architecture or what is frequently referred to as a *sequencer*, is the place where the coordination of the different behaviors normally takes place.

Gat (1998) defines the task of a standard three-layered architecture sequencer as changing the primitive behaviors at strategic moments, that can, for instance, be events or time constraints, to coax the robot into performing useful tasks. His use of the verb "coax" already suggests that this changing between behaviors, the so-called *behavior blending*, is far from being trivial. In fact, hybrid approaches oftentimes employ increasingly complex dynamics for the selection and coordination of multiple low-level behaviors.

AuRA Arkin's AuRA architecture (Arkin and Balch, 1997), being one of the first published approaches to hybrid systems, uses a schema-based approach for robot movement control, whereby Arkin uses the terms of schemas and behaviors synonymously. They represent generic skeletons that are instantiated in the form of potential fields on environmental objects in order to perform movements. Arkin and Balch combine their repository of motor schemas with a so-called *pilot* or *plan sequencer* whose task it is to decompose actions into sets of multiple concurrently running motor schemas. Being a potential field approach, these schemas generate independent velocity vectors that are combined in order to determine the robot's next movement. The accomplishment of tasks is conducted in a purely reactive manner with the possibility to reinvoke higher level planning on failures (Arkin, 1998).

Although this architecture already falls under the hybrid robot control paradigm, the deliberation and sequencing mechanisms provided are mainly restricted to select and coordinate, in the form of a "boiling pot" (Arkin, 1990, p. 1245), low-level motor schemas, or behaviors, that are handling pure robot navigation (e.g. Arkin, 1989, 1990; Arkin et al., 1993; Arkin and Balch, 1997). The original interpretation of actions and behaviors, as it is followed in terms of the AuRA architecture, is hence to be seen as being somewhat more interwoven, than the definition applied in this thesis. An overall system behavior emerges from the combined instantiation of different motor schemas. They do not draw an explicit distinction between purposeful, goal-oriented, willed actions and lower level, automatic behaviors as they were defined here.

Nevertheless, in more recent work of Arkin's group, the action representation shifted away from the AuRA architecture to a meanwhile more standard view of differentiating between high- and low-level behaviors that are interpreted as, for instance, a high-level *soccer* behavior that is the composition of the low-level *find*, *approach*, and *kick* ball behaviors (Ulam and Arkin, 2006).

SSS Similar to Arkin, Connell (1992) proposes with his 3-layered SSS architecture another system that was mainly used efficiently for navigational purposes. Connell interprets actions of the robot as reactions to events that are architecturally integrated in so-called *contingency tables*. While these tables hold symbolic one-step plan representations for reacting in a goal-directed manner to events in the environment, the overall architecture was again mainly used in the field of robot navigation, lacking the needed

level for accomplishing manipulation tasks and thus lacking those concepts comparable to the action definition provided here (cf. e.g. Bonasso et al., 1997).

3T Another hybrid approach, that might well serve as the prototype system for three-layered architectures, is the 3T architecture of Bonasso et al. (1997). The aim of their architecture is the coordination of planned and thus willed activities with real-time behaviors in order to deal with dynamic environments. They propose the three layers of a high-level deliberation, a sequencer, and a dynamically reprogrammable set of *reactive skills*, i.e. their notion of behaviors, as the bottom layer.

Speaking in terms of robot task execution, Bonasso et al. use their sequencer, which is based on a Reactive Action Packages (RAP) approach, to accomplish, what they refer to as, sets of sequenced actions. Hereby, they do not really distinguish the terms of actions and skills. Instead, they understand skills as some low-level system abilities, both of actuator control and perception, that are on a level as, for instance: a skill to visually track a handle, a skill to move the robot's gripper towards a target, and a skill to close the gripper on contact. As their sequencer is working with RAPs, it is based on the idea to monitor environmental events and react on them by executing a corresponding, meaningful skill. Hereby, Bonasso et al. (1997) define the term of skills, or *situated skills* by referring to Slack (1992) as denoting a configuration of the robot's control system that will achieve or maintain a particular state in the world if it is placed in the right context. The actual accomplishment of goal-oriented task execution is thus organized as the ordered instantiation of multiple reactive skills that will reach or maintain a certain state in the world.

The combination of a deliberation layer with a RAP sequencer allowed Bonasso et al. (1997) to demonstrate a hybrid architecture that has capabilities beyond the mere scope of navigation. They demonstrated for instance a trash collecting robot showing more complex abilities like collecting the trash in a certain area and moving to the next one if the latter has been cleared. These robot abilities are achieved by structuring the RAPs in a multi-layered way with, for instance, robot self-localization or approaching of locations on a lower level, and changing areas or searching for objects on a higher level.

But since the 3T skills are comparable to the class of behaviors, as it has been defined in the context of this thesis, this approach again aims at reaching complex and purposeful overall system behavior by combining and organizing multiple low-level behaviors. A distinct and self-contained concept comparable to that of actions is not defined since actions are again being interpreted as being emergent properties of low-level processes.

Saphira Saphira is the further to mention robot control architecture as it shows some similarities to the approach followed here.

This frequently used architecture provides a complete robot framework focussing on the accomplishment of assigned tasks. The most prominent example is the robot Flakey that was used as an office delivery robot (Konolige et al., 1997). Saphira is a hybrid architecture as it as well follows the concept of introducing multiple layers of abstraction to mediate between high-level planning and low-level effector access.

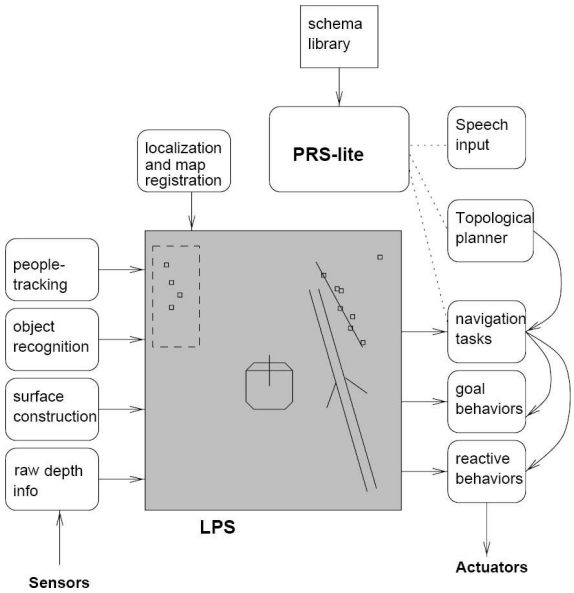


Figure 4.2: The Saphira robot framework. The three components of *navigation tasks*, *goal behaviors*, and *reactive behaviors* show a superficial conceptual similarity to the proposed approach. From Konolige and Myers (1998).

The actual architectural layout of Saphira is depicted in figure 4.2. The components that are of interest in the scope of this thesis are located at the bottom-right of the diagram and represent the task execution system of Saphira. These three are the *navigation tasks*, *goal behaviors*, and *reactive behaviors*. By this, Konolige and Myers (1998) introduce a distinction between several action-guiding behaviors that is, in a superficial way, quite similar to the approach proposed here since they define as well three differently complex layers that are used for robot task execution.

Their bottom layer of reactive or *basic behaviors* has a functionality mostly looking at occupancy information provided to the robot in its *local perceptual space* (LPS), which can as well be understood as the robot's current belief or interpretation of its context and surroundings. Basic behaviors are used for obstacle avoidance and compute a desirability function reflecting the behaviors current preference for those control values that are propagated to the robot's actuators. The behaviors have different priorities and are implemented and combined in their output by using fuzzy logic. Thus, basic behaviors are overall very similar to the definition of behaviors developed during this thesis as they mainly conduct reactive obstacle avoidance.

The second layer of Saphira's task execution system, the goal behaviors, or *goal-seeking behaviors*, is in a way comparable to what is defined here as actions. Konolige et al. (1997) state that these behaviors are meant to perform goal-directed actions by guiding the lower-level reactive behaviors. Hereby, goal-directed behaviors distinguish themselves from basic behaviors as they work on more complex data structures of Saphira's LPS. To be able to accomplish meaningful actions, they use, what is described as, *object*

hypotheses that can be seen as an already processed sensory perception of environmental objects.

The definition of a task within the developed task execution component can similarly be related to Saphira's abstraction layer of *navigation tasks*, though they are not restricted to mere navigation.

Saphira's integration between goal-directed and basic behaviors is nevertheless similar to the other hybrid approaches introduced above. Konolige et al. (1997) organize the control flow by utilizing a procedural reasoning system called PRS-Lite for accomplishing sequencing and deliberative reasoning, though they define another extended control language called COLBERT in later work (Konolige, 1997). The basic principle of the control flow is, nevertheless, that so-called *activity schemas* are *intended* into the system. An activity schema is the representation of a top-level goal like, for instance, directed navigation. It embodies the procedural knowledge of how an objective can be reached as a sequence of subgoals, perceptual verifications, primitive actions, and behaviors. Multiple intended activity schemas form the *intentions* of the robot that will eventually lead to the accomplishment of overall goal-directed behavior. Konolige et al. differentiate between different layers of intentions whereby higher levels are again composed of lower levels, which they can "intend" or "unintend" (Konolige et al., 1997, p. 223) or in other words activate or deactivate. Hereby, high-level intentions can have quite complex tasks to solve. An example would be a plan-and-execute intention for introducing functionalities like path following.

Though the Saphira architecture showed some impressive results and is a frequently applied robot architecture it has been criticized for not being a true three layer hybrid architecture since it does not define distinct components with explicit functionalities. Saphira does not perform classical AI planning but rather distributes deliberation and execution in the several layers of activity schemas. This complex structure had the consequence, that Saphira was only able to accomplish missions that were already available in the system's schema library (see e.g. Ross, 2003). The drawback of the Saphira architecture from the MACS point of view is, therefore, that this concept of activity schemas, intentions of different layers, and goal-directed and basic behaviors, is indeed a complete integrated framework for robot control. The different aspects of this architecture are tightly interwoven, observable for instance on the self-containing arbitrarily complex structure of the different intentions. This lack of clearly structured architectural components makes Saphira less appropriate for being used to demonstrate the applicability of the affordance concept to robotics as Saphira's distributed kind of planning and execution does not easily allow to integrate and point out the usefulness of such a basic and important concept.

Nevertheless, the structure of Saphira's behavior system provides the right ideas for distinguishing the actual task execution layers, though their internal structure is quite different from the definition applied here. Especially the differently complex intentions that organize the accomplishment of real actions remind rather of the dynamic combination of primitive behaviors, as proposed by some of the other approaches. They do not offer the same explicitness for selection and application that an action as a self-contained, independent procedure offers as it is defined in this thesis.

Norman and Shallice The overall concepts developed for this task execution component are also inspired by the work of Norman and Shallice (1986) who have defined a psychological model of human behavior that consists of two distinct systems (see as well Arkin, 1998; Glasspool, 2005). Norman and Shallice base their interpretations on observations of the characteristic errors conducted by several patients with different disorders. They developed the theory of two systems that model *automatic* and *willed* human behavior.

Automatic behavior is hereby closely related to reactive systems and is characterized as an automatic action execution that does not need awareness or attention and consists of multiple independent parallel activity threads. It refers to the well-learned and habitual behavior observable in human action execution and is thus coherent with the definition of behaviors provided above.

Willed behaviors, on the other hand, are interpreted by Arkin (1998) as the interface between conscious deliberation and the automatic behaviors. Though this interpretation as well as the interpretation of Glasspool (2005) suggest to see willed behavior as being incorporated within a scheduling layer of a control architecture, the concept can well be extended to be supported in the lower level of task execution. The definition of actions, as it has just been provided, is very well suitable for implementing willed behaviors in their quality of being attentive, motivated and goal-directed doings.

Summary With AuRA, SSS, 3T, and Saphira being examples of the most prominent hybrid approaches and subsumption architecture representing reactive robot control approaches, an impression has been provided of how other control architectures handle task execution. Most of these systems have the common strategy of defining primitive or low-level control processes and to achieve the capability of an action by combining these low-level processes.

The task execution component defined here, however, aims at providing a supportive structure that aids, augments, and primarily allows the usage and exploitation of affordances for robot control architectures. Therefore, a definition of *actions* has been developed that represents a layer within the actual architectural component of a task execution module. The definition of actions does hereby provide the explicit and self-contained response point to an affordance. This explicitness is exactly the strength of the definition. Where other approaches develop highly complex dynamic systems that aim at integrating and combining different low-level capabilities, the applied concept of an action is rather interpreted and understood as a simple unit that is on its own capable of accomplishing what it needs. If, for instance, a grasp action needs to position a grasper or a crane magnet relative to an environmental object, this process of self-alignment is here interpreted as being an integral part of that grasp action and not a low-level behavior, primitive behavior or skill.

In other words, the introduced definitions of actions and behaviors draw an explicit distinction between purposeful, goal-oriented, and willed actions and continuously running, automatic behaviors. A special focus lies on the actions property of being self-contained and thus including their internal parts and functionalities that are needed to accomplish

the actions. The overall task execution component therefore provides the possibility of directly referring to a decisive part connected to an affordance, namely that very action that is being afforded. This direct response point is one of the main distinctions and advancements compared with other architectural approaches that need to employ sets of concurrently running processes for action-execution. They would only be able to provide indirect relations to affordances arising from the different combinations and dynamics between several low-level processes. It is believed that the definition developed here will help in the process of demonstrating the applicability and usefulness of affordances as the underlying design principle in hybrid robot control architectures.

According to the introduced definition, actions become very well to handle in the context of a complete affordance-based architecture. They can be directly stored in the affordance representation triples and thus be addressed by the planner via operators, mapped to actions in the execution control, and eventually be instantiated in the task execution component.

In other words, the system can directly trigger and monitor the application of an action instead of instantiating and monitoring several parallel running primitive behaviors that somehow move or otherwise influence the robot.

One shortcoming of the applied definition remains to mention that addresses the ability of learning new actions: It is certainly an interesting approach to develop systems that are able to learn completely new sequences of movements and reason about them and what actions they assemble. See, for instance, the recent work of Remondini and Saffiotti (2006), with Saffiotti being one of the main contributors of the Saphira architecture in the past. They develop strategies to combine different sets of primitive behaviors during runtime and blending them, again by means of fuzzy logic, in order to achieve complex behavior. The aforementioned work of Stoytchev (2005a), for instance, also aims at developing an understanding of afforded actions as combinations of single primitive behaviors.

While that approach to task execution is both interesting and promising, it is not that appropriate for using it in the scope of the MACS project and thus in the scope of this thesis. The aim of the project is to demonstrate the robot's ability to make use of the affordance concept, to reason about it, and to develop an understanding of the functionalities that objects offer, but not to develop whole new actions. Nevertheless, such an approach is as well being applied by the MACS partner of the Austrian Research Institute for Artificial Intelligence (OFAI), who use a gripper arm with sets of behaviors that are similar to those of Stoytchev (2005a) that were mentioned above.

The concrete actions that are appropriate in the MACS context and that are listed in the beginning of this chapter, have nevertheless been defined prior to their implementation, since it is clear and previously known, what actions the robot can de facto perform with its manipulator. This is true not only for the MACS robot but for many other robots as well because robots are normally designed and built for a certain purpose that oftentimes implies a rather limited set of actions fitted to its actual manipulator. And even if the robot is being extended and new actions would have to be implemented the system would scale nicely as the affordances that emerge when adding a new action can

themselves again be learned, as has already been indicated in section 3.1.4.

While the learning of whole new motion sequences in order to perform actions is not directly supported, the actual action execution defined here can, given a well-designed implementation, still be influenced and adjusted by parameterizing that action. And since the proposed task execution component allows action parameters that can be learned and adapted, this drawback of being not able to learn abstract actions is rather regarded as a being negligible. In the direct context of demonstrating the applicability of affordance theory in robotics the benefits of a distinction between actions and behaviors do certainly prevail.

During this chapter the notions of *behaviors*, *actions*, and *tasks* have been developed as the underlying principles of the introduced task execution component addressing the different physical and conceptual requirements to the component. The concepts have been compared to other approaches and a special effort has been put on the definition of actions as self-contained units highlighting the concept as an explicit anchor and response point to affordances in the task execution layer.

The following chapter will now attend the actual implementation of those behaviors, actions, and tasks that have been listed above and will finally show a demonstration of the systems abilities.

5 Implementation and Demonstration

This chapter will describe the implementation of the task execution component as it has been developed during the last chapter. A full overview of the component and its different layers is provided in figure 5.1.

At this point it is again to remark, that this task execution component is the part of a complete robot control architecture that deals with accessing the different robot actuators. This brings along that other architectural components, especially those of deliberation and perception will only be mentioned peripheral in the second half of this chapter when demonstrating the system. Due to the same reason, the implementation will only cover a subset of those behaviors, actions, and tasks that were defined during the last section as, unfortunately, neither the perceptual system, nor the robot's manipulator, have been available in time to be included in this thesis. Figure 5.1 therefore highlights those modules that have been developed here. The integration of the other modules, which sadly include the most object manipulation actions, has to be left for future work.

To describe the realization of the task execution component in this chapter, its testbed of the Kurt3D robot will now be introduced, followed by the description of the implementation of the behaviors, actions, and finally tasks. Each of the three different layers will be accompanied with a short demonstration of the applicability of the concept, though this is due to the nature of the demonstrated features only possible in the form of an additional video when concerning actions and tasks.

5.1 The Robot Kurt3D

The Kurt3D robot, see figure 5.2, is based on the Kurt2 robot platform that has originally been developed at the Fraunhofer Institute for Autonomous Intelligent Systems[1] (Worst, 2003).

The Kurt2 platform is extended to a Kurt3D robot by the IAIS-3DLS laser scanner (see reference IAIS-3DLS, 2006). This scanner is a standard SICK LMS 200 2D laser range finder, a rotating mirror device, that is mounted on a tiltable frame. Controlled by a servo drive, the scanner can be pitched to scan the total area of $180°(h) \times 120°(v)$ with a horizontal or vertical angle resolution of up to $0.25°$ each.

The robot is furthermore equipped with two Logitech cameras that are again attached to servo drives allowing to pan and tilt them. See e.g. Nüchter et al. (2003); Surmann et al. (2005) for descriptions of the Kurt3D platform.

The main manipulator of the robot is a 3 DOF magnetic gripper crane arm. Unfortunately, further specifications of the crane arm are not available yet.

[1] Meanwhile the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS).

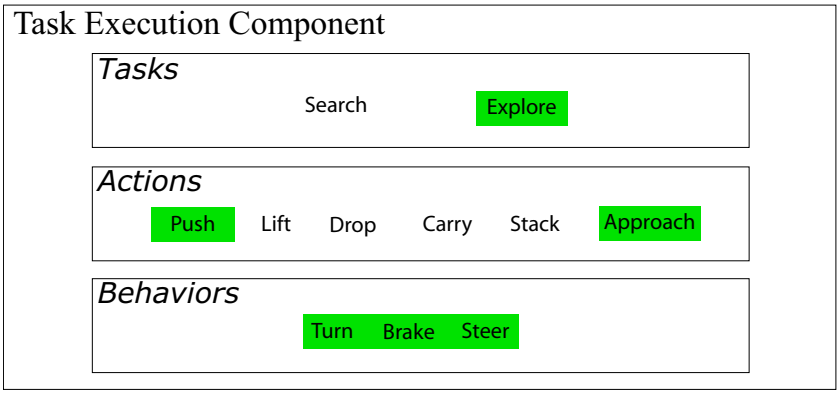


Figure 5.1: The task execution component with its three different layers. The colored modules are those that are presented in this chapter.

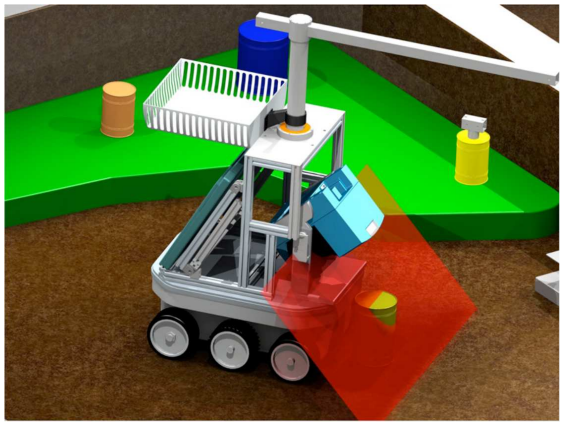


Figure 5.2: The robot Kurt3D as a 3D model including the crane manipulator.

The basic robot platform is approximately 45cm long and 35cm wide. Including the crane arm, its length increases by about 40cm and it reaches a height of 80cm. Its total weight amounts to 27kg. Kurt has six wheels and two motors each controlling one side of the robot. It can thus turn on the spot.

5.2 Behaviors

As has been defined in the last chapter, the behaviors main task is to ensure smooth, collision-free movement in static or dynamic environments. In order to be able to react quickly to environmental changes and to pay tribute to the robot self-motion, behaviors are normally characterized by a very tight coupling between the robot's perception and action.

Although the focus of this thesis does not lie on perception but on actuator control, the tight action-perception coupling of behaviors demands to at least shortly describe the used sensor and methodology that is used for performing obstacle avoidance.

5.2.1 Continuous 3D environment sensing

The robotic sensor that is used for that purpose here as well as in many comparable autonomous robot systems is the laser scanner. A laser scanner gives fast and highly precise distance measurements that can be used for localizing objects and performing obstacle avoidance. One drawback of a normal laser scanner is that it only scans a two-dimensional plane making the robot prone to obstacles below or above the actual scan height. If the robot's laser scanner is, for instance, mounted in a height of 30cm above ground, it will miss obstacles like curbstones or simply small objects like the cans in the MACS scenarios. On the other hand, obstacles that start at a certain height will as well be missed. Typical examples are table-tops as normally only the table-legs penetrate the scanning plane and are thus perceivable by the robot. A whole sequence of work has been conducted at the Fraunhofer lab to extend the standard 2D laser scanner to a device capable of measuring three-dimensional environment information. The basic idea is to pitch the scanner vertically and to generate consistent three-dimensional environment models out of the successive scans taken at the different angles (e.g. Surmann et al., 2005; Nüchter et al., 2003, 2005b,a,c). While they were able to generate highly sophisticated three-dimensional models of the environment, the underlying principle was that the robot stood still while performing a 3D scan. Autonomous robot movement between two successive scan points was restricted only to two-dimensional obstacle avoidance.

This is obviously a drawback when the robot is meant to perform fluent movements while still avoiding obstacles of different heights, at least in dynamic environments. We, and chiefly Dirk Holz, have therefore extended the existing system to continuously pitch the laser scanner in a fashion that reminds of a nodding motion (see Holz, 2006).

Henceforth, the environment is sensed continuously in three dimensions. Thereby it is being transformed into a two-dimensional egocentric map representation of the robot's immediate surroundings that holds information of nearby obstacles. This so-called *virtual obstacle map* that is built by breaking the three-dimensional environment

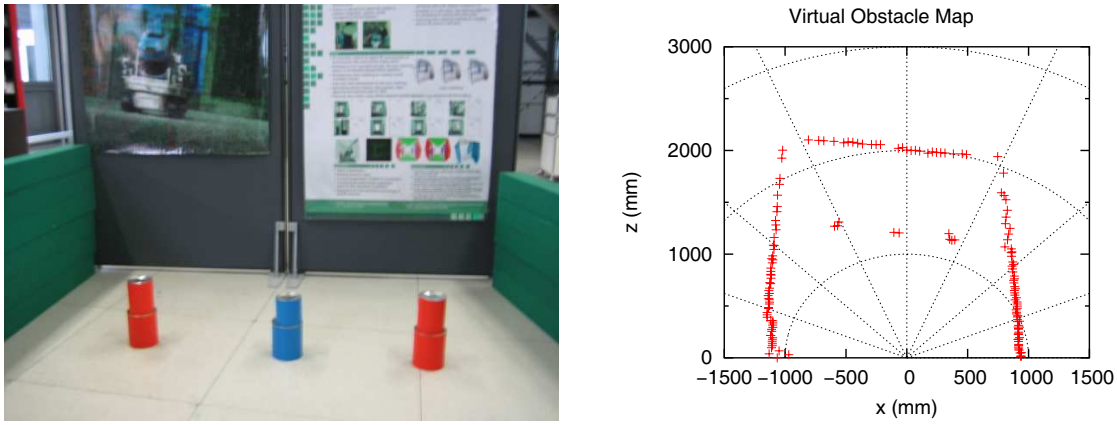


Figure 5.3: Scenario for demonstrating the 3D environment sensing. The setup depicted is similar to that found in the MACS scenarios (left). The different cans all have a height that lies well below the scan plane of a scan taken in a fixed horizontal position. The nodding pitch movement of the scanner nevertheless senses the obstacles and integrates them into an egocentric obstacle map (right).

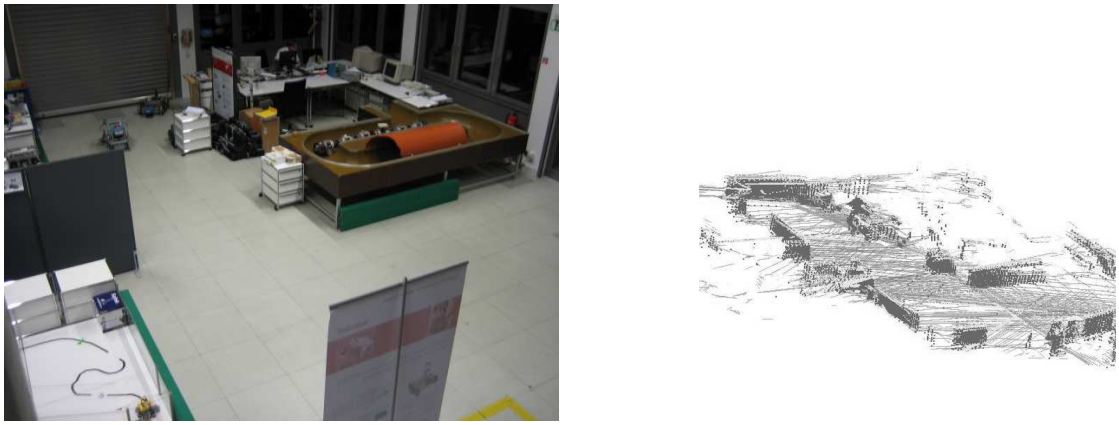


Figure 5.4: Scan area of interest. The picture shows a more complex testing area with the robot standing in front of the grey metal door (left). The generated model clearly shows only those environmental structures that are of the robot's height (right).

information down in one plane is depicted in figure 5.3. Note that in the different scanning directions only the closest readings are stored. For example, if an object on the floor has a height of 20cm, the scanner could easily look above it, measuring large distances in that direction. By taking only the minimal distances in each direction it is ensured that the resulting map will contain those objects that are closest to the robot, regardless of their actual height. It is not important for this transformation on which actual height the obstacle is sensed as the scanner readings that are integrated into the map are only taken from the area of interest, i.e. in this case those points that would interfere with the robot's bounds in width, length, and height.

For comparison, figure 5.4 depicts a complete three-dimensional model built from a more complex scenario. This figure nicely shows that the robot did only scan the vertical area that is of interest for it.

The map that can be used for the purpose of obstacle avoidance is nevertheless that of figure 5.3 (right). It contains in a representational slim 2D display those objects that are closest in the according scan direction. The objects that are inserted into the map are exactly those that would interfere with the robot's movement. They can thus be regarded as obstacles.

The three behaviors of *Brake*, *Turn*, and *Steer*, whose task it is to perform obstacle avoidance and that have been defined in section 4.2.2, are working based on this representation. Note that most of these behaviors are based on algorithms that have already been developed for the Kurt3D robot and are, for instance, explained in Lingemann et al. (2005). Applying them on the virtual obstacle maps, however, extends this fast and reliable methodology to be applicable in 3D environments. While the underlying algorithms are, nevertheless, very similar to those presented e.g. in Lingemann et al. (2005), the approach applied here extends them by embedding them into a behavior framework. Therefore, the principles of the underlying algorithms will now be briefly introduced while afterwards the integration of the three behaviors into a consistent system will be described.

5.2.2 Brake

The *Brake* behavior embodies the most important safety measure of the robot as it slows the robot down if an obstacle occurs in its driving direction. The approach applied here is based on a *virtual roadway* algorithm that has been proposed for the Kurt3D robot in Lingemann et al. (2005).

As already mentioned, the virtual obstacle map contains the information of the obstacles closest to the robot. As the map is egocentric, the robot is always assumed to be at the map's origin facing north. If the robot moves, its odometry-based pose estimation is used to transform the map accordingly. The applied approach for the *Brake* behavior thus only has to iterate over the map's array of distance readings to get the currently best estimation of free space into a specific direction. More specifically, the behavior determines the closest distance to an obstacle (dto) in front of it, i.e. with respect to its width. If this distance falls below the starting criterium of $dto < v_{current} + 1$, with

$v_{current}$ being the robots current speed, the robot starts to brake. If the robot is, for instance, driving at a speed of $1m/s$ it will start to slow down if an obstacle is sensed in a distance of less than two meters in its driving direction.

The strength with which the robot will brake scales with the actual distance to the obstacle. If this distance reading falls below a certain threshold of dto_{min}, the robot will stop immediately. Hereby. the *Brake* behavior computes a scaling factor that is applied to the robot's set speed, i.e. the preset speed for both motors of the robot coming from architectural higher components. Both speed values are scaled down symmetrically in order not to influence the robot's actual driving direction.

One feature that is shared between the different behaviors is that they have *activation values*. In the context of the MACS architecture, the concept of *virtual sensors* has been defined which provide proprioceptive information about the robot's internal state or even deliver preprocessed and thus not real but virtual sensor data to the other architectural components, especially to the learning module.

Activation values are chosen from a range of [0..1]. In the case of the *Brake* behavior, the activation scales proportional to the brake strength the robot applies and becomes 1 if the behavior stops the robot completely while it is of course 0 if the behavior does not need to influence the robot's speed.

The definition if a *Brake* behavior poses the question of what to do when the system has stopped and thus leads us directly to the implementation of the *Turn* behavior.

5.2.3 Turn

The *Turn* behavior aims at getting out of dead ends like e.g. in U-shaped corridors, or simply at changing the robot's direction when facing a very close obstacle, i.e. it turns the robot on the spot if it had to stop.

Different to the previously introduced *Brake* behavior, the *Turn* behavior does not start due to the perceptional information provided by the scanner or the virtual obstacle map but instead becomes active, if the *Brake* behavior reaches a full activation value of 1.

Once active, the *Turn* behavior results in turning the robot on the spot choosing a turn direction that corresponds to the preferred driving direction of the *Steer* behavior (see below).

While the *Brake* behavior is in its core a purely reactive process, the *Turn* behavior maintains an internal state. While a stopping criterium for the turn motion could simply be that the *Brake* activation is not 1 anymore, the *Turn* behavior extends this by applying a kind of *momentum* strategy. Instead of stopping to turn immediately, the behavior continues to turn for a short period of time even if the *Brake* behavior is not active any more. It furthermore remembers its current turning direction for some time after the actual turning has already ended. Both properties are used to avoid incoherent or oscillating movement that might occur otherwise.

On the one hand, by remembering its turning direction, the behavior ensures that the robot will not turn the other way, if it is being triggered shortly after the first time. If the robot is for instance facing a corner, the robot might simply start to switch continuously

from turning to its right to turning to its left, as the minimal braking distance might repeatedly be succeeded when facing the vertex of the corner, resulting in a *Brake* activation value below 1. Continue to turn in the same direction as before, however, will maintain the first turning direction and thus simply turn beyond that vertex even when intermediately deactivated during that turn.

On the other hand, it is for very similar reasons meaningful to continue turning for a short while after the actual *Brake* activation fell below its maximal value again. For instance, the robot might have approached a wall in an obtuse angle. If it will only turn far enough until it can drive a bit further, the overall impression would result in very fast and thus jerky switches between turning and driving forward. But continuing to turn a bit longer than necessary will omit this impression as the robot will in most occasions and in the example of facing a wall be rather able to drive along that wall instead of facing it immediately again. The same method applies when the brake activation only fell due to sensor noise below the necessary level.

Of course, such internal values and especially their usage must be carefully designed. If the robot is for instance driving very slow the time in which it will continue to turn might not be long enough to get out of a local minimum. On the other hand, if it goes very fast, it might miss the direction in which it could commence driving and thus turn endlessly on the spot. Stopping criteria values that adapt to the current robot movement and maybe to the characteristics of the environment may form one solution to this problem. If the robot for instance perceives its surroundings as being rather narrow, it might be useful to turn only shortly in order not to miss any way out of its current situation. Another approach might as well decide to determine the change in the robot's heading rather than the turn time as a stopping criterium, e.g. turn for at least 20°. This would decrease the need to adapt to the robot's speed but is instead subject to odometric errors and is again dependent on the kind of environment it faces.

The approach followed in this thesis, nevertheless does not use one of these strategies but applies simple fixed values as the MACS demonstration scenarios are restricted in a way that does not justify the effort of more complex approaches.

Speaking of an internal state of a behavior, e.g. Gat (1998) states that it is important that the internal state values have to decay over time. Since a behavior's internal state might not reflect the state of the environment truthfully, the robot might again get caught in an unwanted state. For instance, it might always choose to turn right, regardless of the current surroundings. Nevertheless, in the applied approach such a decay of the behavior's internal state is already included when stating that the robot only remembers its last turning direction for a short period of time.

In contrast to the *Brake* behavior, *Turn* completely overrides the robot's set speeds in order to be able to turn on the spot. It hence does not have a scaling activation value but instead a binary. *Turn* is instead active (1), if it is actually turning the robot that is the case if *Brake* has an activation value of 1. Otherwise the activation value of *Turn* will be 0.

Summarizing the design and functionality provided by the *Turn* behavior, it will turn the robot on the spot if the robot's brakes and thus the *Brake* behavior are fully activated.

Its activation values are either 1 or 0, if turning or not turning and it overrides the robot's set speeds. In addition to the functionality of the *Brake* behavior, it maintains an internal state that will decay over time and allows to omit oscillating movement. Although the stopping criteria might be chosen to adapt to both, the robots surroundings and its actual movements, the MACS scenarios do not demand such a capability that is why this approach uses the more simple approach of fixed values for determining the behavior's stopping criteria.

5.2.4 Steer

While the simultaneous execution of both the *Brake* and the *Turn* behavior would lead in a robot that is able to drive around its environment in a zig-zag fashion, driving straight from one wall to another, the *Steer* behavior will provide the means of driving not only straight but into a direction that promises larger navigational space.

Applying again algorithms similar to those explained e.g. in Lingemann et al. (2005), the *Steer* behavior determines a preferred orientation into such an area of supposedly free space.

The actually preferred orientation is computed by processing the raw data of the current 2D laser scan, or in our case, the current egocentric virtual obstacle map (see figure 5.3). The algorithm simply takes the distance readings in each direction and weighs them, first according to their actual distance, and second according to their direction. As the target of this behavior is to deliver an orientation that reflects a driving direction into free space, those distance readings that correspond to far away obstacles will be preferred over those that correspond to close obstacles. In the case of the relatively small MACS scenarios, the arena has a size of roughly $3m \times 4m$, distance values are preferred that are farther away than $30cm$.

The resulting orientation value is furthermore weighted according to its relative direction to the robot with preferring those directions that are close to the robot's current driving direction.

Applying both weightings to the 180 distance readings of the virtual obstacle map yields a preferred direction that is assumed to provide free space for the robot. See Lingemann et al. (2005) for an interpretation of this computation by means of fuzzy logic.

The *Steer* behavior takes this orientation and influences the robot's current driving direction into that direction. Taking the example of an empty room and a robot that is actually driving along a wall, the *Steer* behavior will compute a preferred driving direction that yields the robot into the room and away from the wall as the measured distances to possible objects are much closer in the direction of the nearby wall. Similarly, this behavior will steer the robot into the direction of an open door if it drives straight at a wall, as the distances measured through the door's opening suggest free space in that direction.

It is important here to note that in contrast to the other two behaviors, *Steer* does not have an explicit starting or stopping criterium. Instead it is meant to be always active. Thus, it does not override the robot's set speeds as the *Turn* behavior, but

only changes them to reflect a curvature in the robot's trajectory that will influence its heading and steer it in the computed preferred direction. In other words, the *Steer* behavior generates scaling factors that are applied to the current set speed for achieving this change in orientation.

Similar to the *Turn* behavior, the *Steer* behavior maintains an internal state that simply is a median filter over the last few preferred heading directions obtained by applying the just introduced algorithm. The demand for a decaying internal state value is fulfilled due to a limited history size of the median filter.

The activation value of *Steer* is again related to how much it actually influences the current driving direction of the robot. With a maximally difference of $\pm 90°$ of the preferred heading to the actual heading of the robot, a smaller difference results in a smaller influence of the robot's *Steer* behavior and hence in smaller activation values.

Summarizing the design and functionality of the *Steer* behavior, it will influence the robot's current driving direction into a direction that is assumed to yield large amounts of free space or far away obstacles and that is preferably close to the robot's actually desired heading. The *Steer* behavior applies an internal state median filter of limited size and shows activation values that correspond to the actual change in heading originated by this behavior.

While all three specified behaviors and their provided functionality have just been explained it remains to depict how to integrate the different behaviors with each other in order to achieve a robust and consistent overall system behavior.

5.2.5 Behavior Dynamics

During the motivation and comparison for the concept of actions (section 4.3.2) it should have become clear that the coordination and integration of several behaviors into a consistent system is far from being trivial. However, the explicit distinction between the concepts of behaviors and actions as it is applied in this thesis simplifies this process notably. Since the mere number of behaviors that remain as independent running processes for performing smooth movements and robust obstacle avoidance have been reduced to simply those three behaviors that have just been introduced, the task of coordinating them becomes quite easy.

The setup that is sufficient for integrating these behaviors applies a slim hierarchical ordering that is inspired by Brooks's subsumption architecture (Brooks, 1986). For a more sophisticated method that combines multiple and more complex behaviors for the Kurt3D robot, one might refer to Hartanto et al. (2004) who applies a concept based on dual dynamics (Jaeger and Christaller, 1998).

The hierarchy and design that is applied while combining the introduced behaviors is depicted in figure 5.5. Those behaviors that are located on a higher level within that diagram are able to adjust the output of lower level behaviors or to subsume their output completely. In the concrete implementation followed here, the behavior system gets an input either from the user or from some higher level system component that

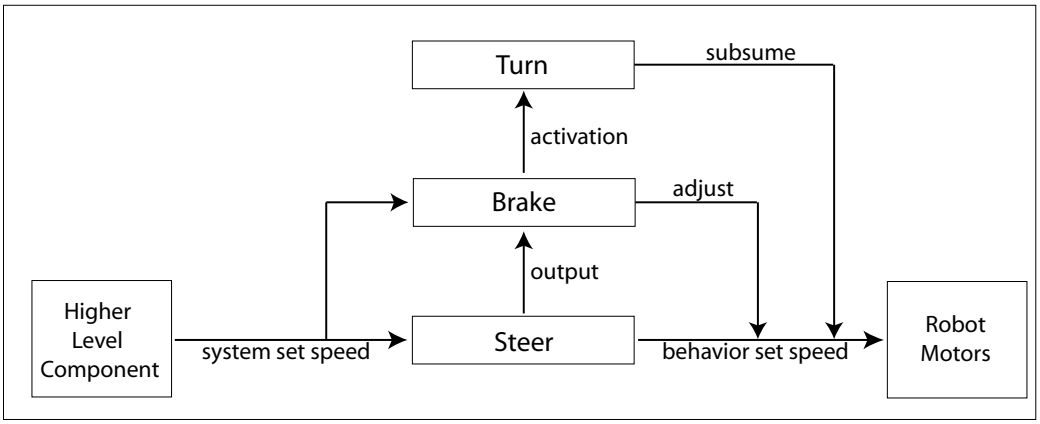


Figure 5.5: Behavior system hierarchy.

is a set speed for both sides of the robot (remember the two independent motors of Kurt3D). The behaviors are executed in a bottom up fashion. In the first step, the *Steer* behavior influences the robot's desired driving direction, which can explicitly be any desired direction, into a presumably free direction in order to avoid collisions early on. Afterwards the *Brake* behavior determines if it has to slow the robot down, which it will do, if necessary, without changing the robot's heading. Note, that the *Brake* behavior will work on the higher level set speed if the *Steer* behavior has been deactivated and does not produce any output. The *Brake* behavior will only adjust the set speed or the output of the *Steer* behavior without subsuming it completely. The *Turn* behavior, in contrast, becomes active if the *Brake* behaviors activation reaches a certain value, in this case 1.0, and completely replaces the set speed of the system. In this setup this means that the robot will immediately start to turn on the spot if it had to brake because of a close obstacle. Hereby, the *Turn* behavior does of course not need the set speed information of other behaviors since it replaces the set speeds completely instead of just influencing them like the other two behaviors.

This simplistic structured behavior system coordinates the three specified behaviors in a way sufficient to deal with those requirements faced in the MACS scenarios. Some demonstrations showing the basic capabilities of this behavior system will now shortly be introduced.

5.2.6 Demonstrating the Behavior System

The basic applicability of the overall concept and the behavior interaction framework is depicted in figure 5.6 and will of course as well be observable when demonstrating actions and tasks later on.

The demonstration's setup consisted of an arena similar to that depicted in figure 5.3 (left) extended with an opening in the back. The robot's set speed was set to $0.3m/s$ for both wheels. The actual experiment was conducted twice. Figure 5.6 (left) shows the trajectory that resulted from applying the three behaviors on normal 2D laser

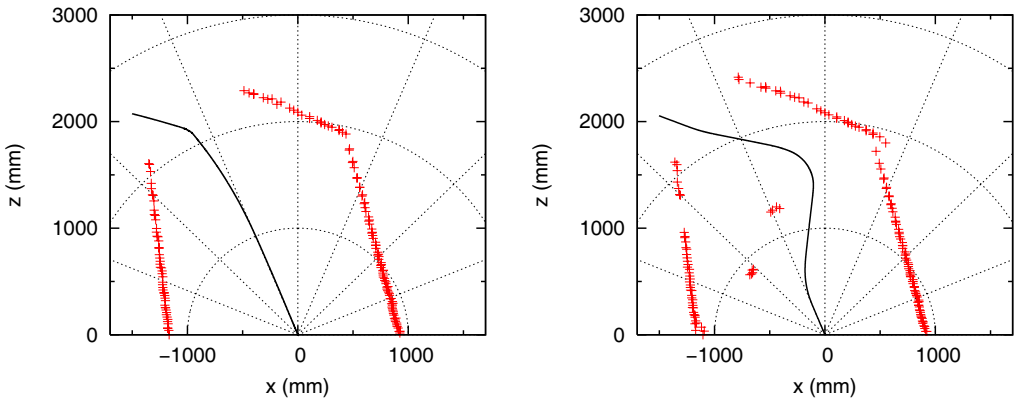


Figure 5.6: Demonstration of the behavior system. Depicted are the autonomously achieved trajectories for both, driving with the scanner held in a fixed horizontal position (left) and with the continuously pitching scanner (right).

data, whilst the right part of figure 5.6 depicts the resulting trajectory of the robot based on the virtual obstacle maps achieved with the pitching scanner. Both resulting trajectories avoid all perceived obstacles and the robot steers out of the area at the large opening. This nicely shows the successful interaction of the three behaviors and as well the applicability of the underlying algorithms to both standard 2D laser range data as well as the virtual obstacle maps.

Among the supplementary videos there is as well one that shows the autonomous driving capabilities of the robot applying all three different behaviors.[2]

One should note here that these demonstrations have not only been conducted to demonstrate the behavior system and the autonomous driving capabilities of the robot but as well to demonstrate the usefulness of virtual obstacle maps. In this context they are as well used in Holz (2006).

The general combination of the *Brake*, *Turn*, and *Steer* behaviors yields an emergent system behavior that can be described as *roaming* the environment. This will be demonstrated in combination with the *Explore* task in section 5.4.2.

To summarize this section, a set of behaviors has been presented for performing autonomous navigation while avoiding obstacles based on continuously gathered 3D data.

Due to the definition of the concept of actions that takes care of object manipulation, the introduced simple behavior system provides the necessary means for combining the three behaviors *Steer*, *Brake*, and *Turn*. The system is furthermore able to run with, for instance, the *Steer* behavior switched off completely.

While this system is sufficient for driving around in an environment, the following section will now introduce the current state of implementation of the next layer of the defined task execution component that handles purposeful actions.

[2]Note that an online version of all videos can be found on http://www.loerken.net/ma/videos.

5.3 Actions

During the last section a set of behaviors has been introduced that provide the necessary means for driving around in an environment. As has been defined in section 4.2.2, actions introduce purposeful goals into robot task execution. As the robot's manipulator was unfortunately not available in time, there are only two actions described and implemented in the context of this thesis. The first one is the *Approach* action as the action which performs not mere reactive movement but rather approaches a given robot pose while applying a nicely shaped trajectory and taking into account the kinematics of the actual system. The second is the only object manipulation action the system can perform without a manipulator, namely the *Push* action.

Actions are meant in this context to be executed sequential rather than, like the behaviors, in parallel. Hence, there is no need for specifying a framework similar to that for coordinating the behavior system.

This leads to the further main difference between actions and behaviors as actions cannot specify a meaningful activation value since they are either active or not, but not only to a certain degree. To be able to provide some feedback to the system, that can for instance, be used for learning, but as well might be usable for replanning, actions may provide a measure of their *grade of completion* when applicable. The idea behind this is that many actions can be described as a sequence of atomic movements that executed in a row will produce a certain action. A lift action, for instance, has to pivot the crane above the corresponding test object, move the magnet to the appropriate position along the extension arm, lower the magnet, make contact with the object, switch on the magnet, and eventually trying to lift the object. As the sequence of steps does not change between several applications of the action, of course unless the robot has to redo a certain step, the system can determine a grade of completion depicting of how many of the necessary steps it has already fulfilled. Again this determined feedback is primarily used as a virtual sensor of the system.

Due to actions being goal-oriented, they may actually themselves provide a feedback to the system whether their execution succeeded or not. Regarding the example of lifting an object, the action might return a success after all steps have successfully been executed. If the test object was too heavy to lift up the magnet while switched on, or if the robot already failed to place the magnet on that object, for instance, if it was out of reach or inappropriately shaped, the action might return a fail value. Especially when combined with the notion of a grade of completion, the system could then reason about what part of an action failed and adjust its successive actions accordingly.

Taking these basic differences into account, the following will now introduce the two actually implemented actions of *Approach* and *Push*.

5.3.1 Approach

The *Approach* action applies a controller that is capable of driving trajectories suitable for robots with differential drive to reach a target position with a certain target heading. The controller is an implementation of Indiveri (1999) and is internally called the *Gio-*

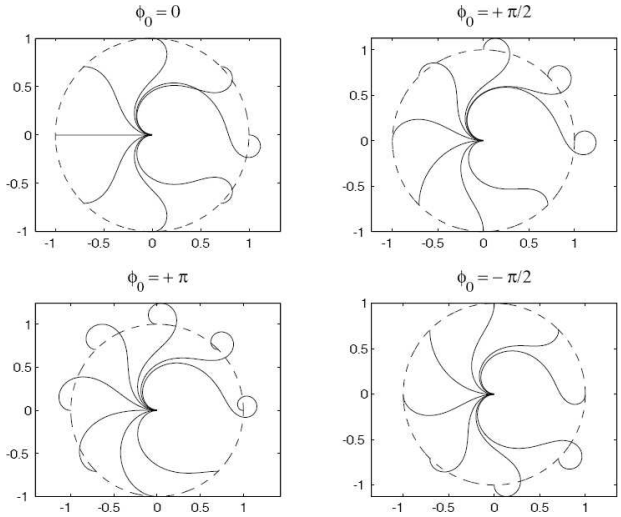


Figure 5.7: Trajectories that result from applying the Giovanni Controller. The trajectories are starting on the unit circle with different start headings of the robot (ϕ_0). They are all ending up in the origin facing rightwards. From Indiveri (1999).

vanni Controller. Since an implementation of this algorithm for the Kurt3D robot is as well presented e.g. in Lingemann et al. (2005) the details will only shortly be presented in appendix A.

Here it should suffice to say that the *Approach* action is based on a controller-like approach that moves the robot step-wise to a desired target coordinate by computing the currently best fitting set values for the robot's speeds according to a simplified kinematic model of the system.

The resulting trajectories have nicely shaped curvatures that are depicted in figure 5.7. Regarding this figure, note that the controller is designed in a way that it will reach the target pose $(0, 0, 0)$, i.e. the origin with a northward heading. Of course, the implemented action can easily transform global world coordinates to bypass this constraint.

The part that distinguishes this *Approach* action from the prior implementations of the controller is that, being embedded as an action in the developed task execution system, the specified behaviors for obstacle avoidance can simply be left running while the action is being executed. The applied controller approach always computes the currently fitting set speeds based solely on the robot's pose relative to its target without considering the robot's surroundings. This set speed can then serve as input to the behavior system which will take care of the necessary obstacle avoidance. The action is in this matter equivalent to the input o the higher level component depicted in figure 5.5. Note however, that this combination does not save a developer from creating a more sophisticated path-following system if navigational tasks of a higher complexity like e.g. in office delivery setups are requested. In the MACS context, however, this approach

proves sufficient.

Furthermore, the *Approach* action is most of the times used to closely approach a test object in the scenario. For this reason the actual action has the capability of deactivating the behaviors when it is close to the target, as they might otherwise prevent the system from eventually reaching its target location. Using this parameterizable feature, one should however keep in mind that it always might endanger the system if deactivating the obstacle avoidance security features provided by the behavioral layer.

Unlike an object manipulation action, the *Approach* action does not apply a sequence of steps for reaching its goal. It thus does not implement the possibility of returning a grade of completion of the action it executes. Nevertheless, the approach action can determine a fail or success state if its execution is coupled with a certain time frame during which the target pose has to be reached. If the robot does not reach that location within the specified time it can be considered as a failed execution.

Summarizing the characteristics of the *Approach* action, it is implemented in a controller-like fashion driving the robot on trajectories appropriate for a comparable kinematic model to its goal location. In combination with the behaviors specified above, the action can perform obstacle avoidance while being executed. It can return a success or fail state if executed with specifying a limited time frame but does not define a meaningful grade of completion.

5.3.2 Push

The push action is probably the most simplistic action the robot can apply. Since *Approach* is an action on its own, the other specified actions, including the *Push* action, can assume that the robot has already approached the location that is appropriate for its actual application. The task of approaching the location is left to the higher-levels of the system that have access to higher-level world information like e.g. maps.

Therefore, the *Push* action only drives the robot forward a bit, pushing everything that is standing in front of it. On the one hand, since this is actually a one step action, the grade of completion is again not bearing much information. On the other hand, the push action can again determine a failure or success state. The actual robot is equipped with a motor controller that adjusts the electric current for the motors in order to reach a desired set speed. If this current, that is also provided as a virtual sensor to the system, is exceeding a certain threshold, i.e. the robot needs too much current to reach a desired speed, it can be assumed, that the robot is standing in front of a test object that does not afford pushing. In this case, the implemented push action will simply reverse the robot a bit and return a failure notification stating that the action was not accomplished successfully.

Note at this point that the successful application of an action is oftentimes a perceptually challenging task to confirm and that the actions defined here are not meant to bear this full load. It is only reasonable to return such a success or fail value if it can easily be determined from the view of an action. Judging, for instance, whether the action succeeded because the robot was standing on ice and only perceived itself to be

moving while its wheels were actually only spinning on the spot, lies far out of scope of the defined concept of actions.

Summarizing the characteristics of the *Push* action, it is implemented as a method to simply try to push everything in front of the robot. The action might easily return a feedback of success or fail that is related to the needed motor current for accomplishing the push action. On the case of failure the action reverses the robot a bit to detach it from the corresponding test object. The meaning of the grade of completion is somewhat restricted as *Push* is only a one step action.

5.3.3 Demonstrating Actions

A demonstration of the defined *Approach* and *Push* actions can be found with the supplementary videos. The pushing video demonstrates the approaching of a target location and in one case the successful accomplishment of pushing a lightweight object and in the other case a failed pushing action that is correctly recognized by the system as can be seen by the robot moving backwards shortly after trying to push.

5.4 Tasks

As specified in the last chapter, the defined task execution component defines an intermediate layer between the actual scheduling and accomplishment of a whole plan and the accomplishment of a subsequence of such a plan, namely a task. Tasks are hereby specified to encapsule meaningful portions of work that do not require much deliberation for their execution.

While it has shortly been argued that this concept may bear high potential concerning active perception tasks (see section 4.2.1), the specified search task could, unfortunately not have been implemented as the needed perceptional models were not ready in time.

Nevertheless a basic implementation of the *Explore* task has been established and will now be introduced.

5.4.1 Explore

The *Explore* task provides the functionality of roaming an environment and applying actions to objets that are deemed interesting. Being on the executional level of tasks, *Explore* can thus be described as a sequence of accomplishing the following steps:

1. Roam the environment for a while.
2. Look for interesting objects and localize one of them.
3. Approach the object.
4. Try out an action on that object to learn what it affords.

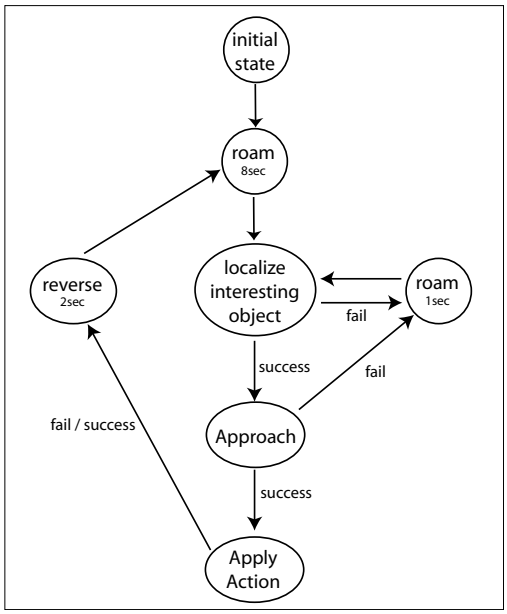


Figure 5.8: Explore task automaton. The system starts in the initial state. All arrows that are not labelled depict state transitions that are executed automatically after the time span specified in the originating node was exceeded.

5. Reverse a bit to ensure that the object is within the pitching scanner's field of view.

6. Continue with step 1.

As can be seen by this list, the actual assignment to the *Explore* task is to schedule a sequence of simple movement requests and action applications. It distinguishes itself in complexity from the superordinate architectural components as there is no high-level goal to be fulfilled by this simple task. It may instead be run continuously.

The applied implementation of this task can be depicted similar to a simple finite state machine extended by the possibility to introduce durations for states.

An automaton visualization for accomplishing the *Explore* task is depicted in figure 5.8. Note that the nodes represent actions or movement that is partially limited by the amount of time specified in the nodes but not states in the normal sense of state automata. The unlabelled arcs depict transitions from executing one action to executing the next one which are applied due to the time limit that is connected to the first action. The idea behind the *Explore* task is to allow parametrization of the task's perceptual filters that should be used for localizing an interesting object as well as to specify beforehand the actual action to be executed as the decision of what objects to probe for what actions is a higher-level process that might even be triggered by the system's leaning component.

The *Explore* task applies an object localization approach based on Vocus (Frintrop, 2005) that is an attention system looking for salient features in images in either a top-down or bottom-up way. The *Explore* behavior can thus be extended to support searching for objects specified in a top-down manner coming close to the functionality of a *Search* task.

The original system of Vocus as it is used here has been extended and combined with a triangulation method for localizing objects using the stereo camera system of the Kurt3D robot by Stefan May and Maria Klodt.

The overall accomplishment of a roaming movement is simply achieved by providing a set speed to the *Brake*, *Steer*, and *Turn* behavior of the task execution component's behavior layer.

Summarizing the characteristics of the introduced *Explore* task, it provides the means for executing a simple sequence of movements and actions that have to be specified when instantiating the task. The scheduling it accomplishes is on a level low enough to not demand the affiliation to a higher-level architectural component. Note, that the actual implementation of such a task like the *Explore* task could as well be managed by the level of the architectural scheduler, e.g. by the execution component instead of the task execution component, but the design considerations that lead to this decisions have already been elaborated in section 4.2.2.

5.4.2 Demonstrating Tasks

A demonstration of the defined *Explore* task can be found with the supplementary videos. The video demonstrates the robot that is roaming its environment and applies its object localization algorithm every once in a while. If it is able to locate an interesting object, it approaches its location and stops in front of it being a placeholder for some object manipulation actions that might be executed when the crane is ready. Afterwards, the robot reverses a small distance and starts roaming again.

6 Summary and Outlook

The first section of this last chapter will give a summary of the thesis presenting the applied line of argument and the developed concepts. The remainder will finally point out some further directions that might extend the system beyond its functionality as it is provided here.

6.1 Summary

The overall task of this thesis was to design an affordance-based task execution component focussing especially on the requirements proposed by the MACS architecture and to develop a prototypical implementation of this component.

While the overall work has been motivated by giving a description of how affordance theory may augment nowadays robotics, an interpretation of the affordance term has been shaped that forms the basis of the newly developed task execution component.

According to this interpretation, affordances correspond to the relations between environmental features and an agent's capabilities for accomplishing an action. In other words, an affordance is a function centered view on the environment such that an agent perceives which action it can perform on which object: it perceives which actions are afforded.

This thesis is part of the MACS project, whose main target it is to demonstrate the applicability and usefulness of affordance theory in robotics. For that reason, the need was highlighted, in this project, to integrate the affordance concept explicitly as a first-class citizen into the overall architecture. This explicitness of the representation is one of the key aspect of the project since only an explicit exploitation of the concept can demonstrate the benefits of affordance theory while distinguishing itself in expressiveness from other approaches that, more or less, only claim to use affordances.

Following this line of argument, in this thesis, the design issue has been put on the architectural layer of a task execution component to provide the underlying means for grounding the affordance concept within the very layer that actually accesses a robot's actuators.

It has been argued that affordances are most interesting and can best be demonstrated not on a reactive level like, for instance, an obstacle that affords avoiding but rather on the level of purposeful actions like, for instance, a can that affords lifting or a door that affords opening.

This definition shows the potentiality for actually designing the task execution component based on the affordance concept and yields to the desired actuator-close grounding or anchoring of affordance theory in the overall architecture. This anchoring has been

achieved by distinguishing the three different layers of *behaviors*, *actions*, and *tasks* in which the explicit definition of a concept of actions forms the direct response point of this component to affordance theory.

While the task execution layer of behaviors organizes the basic navigational abilities of a robot and the layer of tasks mainly forms a conceptual interface to superordinate architectural components, it has been argued that actions are what affordances afford.

Since an affordance is defined as the relation between environmental features and the agents capabilities to act upon those features, the concept of actions as being purpose- and meaningful doings, like e.g. lift, push, and stack, has been argued to represent the just mentioned response point to affordance theory.

The explicitness of the formulation is furthermore enhanced by defining actions as self-contained units which distinguishes this concept from many other approaches that try to achieve similar complexity by combining multiple, oftentimes reactive, low-level behaviors. An action on the other hand encapsules all necessary control loops for actuator control within that one particular action that is accomplished by these control loops. It is believed that this encapsulation eases the architectural representation and thus the usage of this directly affordance-related concept within the other parts of the architecture, especially when it comes to deliberation, learning and execution control.

The defined and developed system has been implemented as far as it is was possible considering the missing crane manipulator of the demonstration robot and the only late and partially available perceptual modules.

The basic applicability of the implemented system has, again as far as possible, been shown in separate demonstrations for each task execution layer of behaviors, actions, and tasks.

6.2 Outlook and Further Work

While the concept of the developed task execution component has been defined and motivated in detail, its actual implementation is, unfortunately, far from being complete.

Despite the fact that of course the next step of the work is to integrate the manipulator and perceptual units and to implement the according actions, some interesting further directions can already be stated that may enhance the component in the context of the MACS project but might as well help to extend the definitions beyond the restricted scope of the demonstrator scenarios.

The probably most interesting extension to the system would be to allow for suggestions of the task execution component on which actions are applicable in the current context. Many of the aforementioned behavior-based approaches already provide this functionality on a lower-level. Referring again to Stoytchev (2005a), his defined behaviors, like for instance *position-wrist*, are designed as environmentally triggered processes. Of course, it is not reasonable to define an action, as it has been defined in this thesis, to be environmentally triggered since although many objects are reachable one does not simply reach for them. Nevertheless, such an applicability context of an action might well be used to enrich the representation of actions on a level that can be used in the

affordance representation since it could be interpreted as a, virtually speaking, tickling in the fingers of the robot to try out a specific action.

A similar and not less interesting point is for instance highlighted by Cooper and Glasspool (2001) who speak of environmental triggered behaviors like *novelty seeking* or *boredom avoidance*. Similar to Arkin et al. (1993) who defines an *avoid-past* behavior repelling the robot from areas it has just visited, actions could as well support such considerations. While it is of course an architectural design question of where to include such reasoning processes for action arbitration, actions could in a first step simply remember their history of application. For instance, if they simply maintain, in form of an internal state, a counter that decays over time and reflects the number of successful executions of that particular action, an action arbitration mechanism could easily use this information to, for instance, select another action to apply during an *Explore* task. In the context of the MACS architecture such an extension could easily be included as a virtual sensor.

Besides these possible extensions that might enhance the action layer of the defined task execution component, especially the task layer provides arbitrarily many possibilities for extensions.

Speaking in the direct terms of the MACS scenarios, a mapping functionality would be helpful that could, in an active perception way of thinking, explore the environment actively combining the searching for objects that afford actions with mapping capabilities. It would presumably be helpful to be able to generate maps that remember the locations of test objects and what they have afforded to the robot in the past as a kind of symbolic environmental interpretation. Similar to this idea, Doherty et al. (2005) have already introduced traversability and standability maps. Their approach, however, mainly concerns affordances that would be classified on a behavioral layer in the context of the definitions provided in this thesis. They only describe a vague concept of affordance maps that can be related to the purposeful layer of actions. Their ideas of, for instance, graspability maps is not considered as being of much help here since it is not reasonable to maintain different maps for each affordance. Furthermore, affordances of that type are normally connected to real objects in the environment and not to whole areas.

When integrating mapping capabilities in the approach the extensions could also include more sophisticated environment exploration techniques like an extension of the standard *next-best-view* planning approaches to more affordance-related *next-best-action* planning approaches. One might nicely combine this idea with the boredom avoidance proposal from above.

While these last suggestions probably exceed the reasonable effort that is justifiable in terms of the MACS project it could definitely enrich the system and is regarded as a necessary step when porting the architecture to more complex scenarios.

The certainly most important aspect of the future work regarding this task execution component is to eventually evaluate the defined concepts in the context of a complete affordance-based architecture. It has to be reviewed whether the actual concept of behaviors, actions, and tasks actually augments the overall architecture in the way as

it is intended here. Unfortunately, such an information can only be reached and such a statement can only be made if the architecture is being evaluated as a whole, postponing this much needed process to the very end of the MACS project.

A Controller of the Approach Action

This controller that is used as the basis for the *Approach* action is an implementation of Indiveri (1999) that has already been used on the Kurt3D robot (e.g. Lingemann et al., 2005).

Assume the following cartesian kinematic model, that is slightly simplified in terms of assuming a bicycle-like robot:

$$\begin{aligned} \dot{x} &= u \cos \phi \\ \dot{y} &= u \sin \phi \\ \dot{\phi} &= u \frac{\tan \psi}{l} = uc \end{aligned} \tag{A.1}$$

are the partial derivatives of the robot's change in pose w.r.t. time. Hereby, u is the linear velocity of the robot, l is its length, ϕ is the robot's global orientation angle whilst ψ is its local steering angle. The value c refers to the curvature of the robot, i.e. the actual bend of the trajectory the robot is driving that equals $\frac{\tan \psi}{l}$, i.e. the part of the motion model that actually accounts to the constraints of bicycle-like motion. The curvature is bound to some upper value that depends on the actual robotic system. In other words, this kinematic model neglects the fact that the Kurt3D robot can actually turn on the spot.

Transforming the model of Eq. A.1 from the cartesian model into the polar coordinate space yields the following derivatives w.r.t. time:

$$\begin{aligned} \dot{e} &= -u \cos \alpha \\ \dot{\alpha} &= -u \left(c - \frac{\sin \alpha}{e} \right) \\ \dot{\theta} &= u \frac{\sin \alpha}{e}, \end{aligned} \tag{A.2}$$

with (e, α) defining a location in polar space and θ the heading of the robot in that location.

The system is meant to converge to pose $(e, \alpha, \theta) \rightarrow (0, 0, 0)$ (polar).[1] Thus, due to the characteristics of polar space coordinates, e can be seen as the Euclidean distance of the current robot pose to the target with α describing the direction of the target.

To reach the target pose according to the motion model a closed loop controller is applied to acquire the set values for the current speed and curvature by using a quadratic

[1] Real world target coordinates can of course easily be locally transformed to meat these criteria.

Lyapunov function candidate for asymptotically stable error dynamics. The controller is specified by:

$$\begin{aligned} u &= \gamma e && : \gamma > 0 \\ c &= \frac{\sin\alpha}{e} + h\frac{\theta}{e}\frac{\sin\alpha}{\alpha} + \beta\frac{\alpha}{e} && : \beta, h > 0 \end{aligned} \tag{A.3}$$

with the Lyapunov parameters of γ, h and β.

As the target pose is $(0, 0, 0)$ all three parameters e, α and θ are supposed to converge to zero. This is reached if according to Eq. A.3 the Lyapunov parameters are chosen as:

$$h > 1; 2 < \beta < h + 1. \tag{A.4}$$

According to Eq. A.3 and A.4 the curvature of the system c is bounded on the trajectory and will asymptotically converge to zero as requested. By additionally specifying an upper bound (saturation) for u, the two requirements of saturated actuators and bounded curvature that have to be fulfilled for real robotic systems are incorporated yielding a controller that will eventually guide the robot to pose $(0, 0, 0)$. The robot will come to a halt at that position as (see Eq. A.3) u is scaled down by the decreasing distance e to the target. Of course, in real world applications, the target pose should have a certain tolerance. Defining u this way also constrains the system to drive only forward (i.e. u is always positive) as $e > 0$ holds and γ is by definition > 0.

Some exemplary trajectories that this behavior produces are depicted in Fig. 5.7 (p. 76).

The controller brings along the features of being a non-linear, time-invariant, globally, exponentially and asymptotically converging control law that is constrained to move in only one forward direction, resulting in smooth curves stopping automatically at the target pose (see Indiveri, 1999).

Bibliography

P. Agre and I. Horswill. Lifeworld analysis. *Journal of Artificial Intelligence Research*, 6:111–145, 1997.

R. C. Arkin. Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8(4):92–112, 1989.

R. C. Arkin. The impact of cybernetics on the design of a mobile robot system: a case study. *IEEE Transactions on Systems, Man and Cybernetics*, 20(6):1245 – 1257, Nov-Dec 1990.

R. C. Arkin. *Behavior-Based Robotics*. Intelligent Robots and Autonomous Agents. MIT-Press, Cambridge, MA, USA, 1998.

R. C. Arkin and T. Balch. Aura: Principles and practice in review. *Journal of Experimental and Theoretical Artificial Intelligence JETAI*, 9:175–188, 1997.

R. C. Arkin, K. Ward, T. Balcher, T. R. Collins, A. M. Henshaw, D. C. MacKenzie, E. Nitz, and D. Rodriguez. Buzz: An instantiation of a schema-based reactive robotic system. In *Proc. of the Int. Conf. on Intelligent Autononmous Systems*, pages 418–427, 1993.

R. Bajcsy. Active perception. In *Proceedings of the IEEE*, pages 966 – 1005, 1988.

L. Bogoni and R. Bajcsy. Interactive recognition and representation of functionality. *Computer Vision and Image Understanding*, 62(2):194–214, September 1995. Special Issue on Functionality.

R. P. Bonasso, R. J. Firby, E. Gat, D. Kortenkamp, D. Miller, and M. Slack. Experiences with an architecture for intelligent, reactive agents. *Journal of the Robotics Society of Japan*, 9(1):237–256, 1997.

R. Breithaupt and E. Rome. Report on experiment design. Deliverable D6.4.1, MACS Internal Technical Report, Fraunhofer Institute for Autonomous Intelligent Systems (FhG-AIS), Sankt Augustin, Germany, October 2005. Version 2.

R. Breithaupt, S. Frintrop, J. Hertzberg, E. Rome, and B. S. Müller. Specification of final demonstrator. Deliverable D6.1.1, MACS Internal Technical Report, Fraunhofer Institute for Autonomous Intelligent Systems (FhG-AIS), Sankt Augustin, Germany, October 2005. Version 2.

R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, March 1986.

R. A. Brooks. Intelligence without representation. *Artificial Intelligence Journal*, 47: 139–159, 1991a.

R. A. Brooks. New approaches to robotics. *Science*, 253:1227–1232, September 1991b.

R. A. Brooks. Intelligence without reason. In J. Myopoulos and R. Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 569–595, Sydney, Australia, August 1991c. Morgan Kaufmann. ISBN 1-55860-160-0.

T. Brox, B. Rosenhahn, D. Cremers, and H.-P. Seidel. High accuracy optical flow serves 3-D pose tracking: exploiting contour and flow based constraints. In A. Leonardis, H. Bischof, and A. Pinz, editors, *European Conference on Computer Vision (ECCV)*, volume 3952 of *LNCS*, pages 98–111, Graz, Austria, May 2006. Springer.

A. Chemero. An outline of a theory of affordances. *Ecological Psychology*, 15(2):181–195, 2003.

A. Chemero. Information and direct perception: A new approach. In P. Farias and J. Queiroz, editors, *Advanced Issues in Cognitive Science and Semiotics*, 2006. (to appear).

A. Chemero. Toward a situated, embodied realism. In J. Burgos and E. Ribes, editors, *Knowledge, Cognition and Behavior*, pages 75–93, 2007. (to appear).

J. H. Connell. SSS: A Hybrid Architecture Applied to Robot Navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2719– 2724, Los Alamitos, California, 1992.

R. Cooper and D. Glasspool. Learning action affordances and action schemas. In R. M. French and J. Sougne, editors, *Connectionist Models of Learning, Development and Evolution*, volume 16 of *Perspectives in Neural Computing*, pages 133–142, London, 2001. Springer-Verlag.

J. Cornwell, K. O'Brien, B. G. Silverman, and J. Toth. Affordance theory for improving the rapid generation, composability, and reusability of synthetic agents and objects. In *12th Conf on Behavior Representation in Modeling and Simulation*, May 2003.

J. Diaz, A. Stoytchev, and R. C. Arkin. Exploring unknown structured environments. In *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS-2001)*, pages 145–149, Key West, Florida, 2001. 21-23 May.

P. Doherty, T. Merz, P. Rudol, and M. Wzorek. Tentative Proposal for a Formal Theory of Affordances, Tentative Proposal for an Affordance Support Architecture, Prototype: Affordance-Based Motion Planner. Deliverable D4.2.1 + D4.3.1, MACS Internal Technical Report, LiU-IDA Linköpings Universitet, Linköping, Sweden, October 2005.

G. Dorffner, J. Irran, F. Kintzler, and P. Pölz. Robotic learning architecture that can be taught by manually putting the robot through action sequences. Deliverable D5.3.1, MACS Internal Technical Report, Österreichische Studiengesellschaft für Kybernetik (OFAI), Vienna, Austria, October 2005a.

G. Dorffner, P. Pölz, J. Irran, and F. Kintzler. Overview of existing affordance learning approaches. Deliverable D5.1.1, MACS Internal Technical Report, Österreichische Studiengesellschaft für Kybernetik (OFAI), Vienna, Austria, February 2005b. draft version.

A. P. Duchon, W. H. Warren, and L. P. Kaelbling. Ecological robotics. *Adaptive Behavior*, 6(3):473–507, 1998.

P. Fitzpatrick and G. Metta. Grounding vision through experimental manipulation. *Philosophical Transactions of the Royal Society: Mathematical, Physical, and Engineering Sciences*, 361(1811):2165–2185, 2003.

P. Fitzpatrick, M. Giorgio, L. Natale, S. Rao, and G. Sandini. Learning about objects through action - initial steps towards artificial cognition. In *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, pages 3140–3145, Tapiei, Taiwan, September 14-19 2003.

S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search*, volume 3899 / 2006 of *Lecture Notes in Computer Science, Subseries: Lecture Notes in Artificial Intelligence (LNAI)*. Springer, Berlin/Heidelberg, 2005.

E. Gat. On three-layer architectures. In D. Kortenkamp, R. Bonasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, MIT Press, pages 195–210, Cambridge, MA, USA, 1998.

E. J. Gibson, K. E. Adolph, and M. A. Eppler. Affordances. In R. A. Wilson and F. C. Keil, editors, *The MIT Encyclopedia of the Cognitive Sciences*, pages 4–6. MIT Press, 1999.

J. J. Gibson. *Motion picture testing and research.* U.S. Government Printing Office, Washington, DC, 1947. (Army Air Force Aviation Psychology Program Research Rep. No. 7).

J. J. Gibson. *The Perception of the Visual World.* Houghton Mifflin, Boston, 1950.

J. J. Gibson. *The Ecological Approach to Visual Perception.* Lawrence Erlbaum Associates, Hillsdale, 1979.

D. W. Glasspool. The Integration and Control of Behaviour: Insights from Neuroscience and AI. *Visions of Mind: Architectures for Cognition and Affect*, pages 125–148, 2005.

R. Hartanto, F. Schönherr, M. Mock, and J. Hertzberg. Target-oriented mobile robot behaviors for office navigation tasks. In M. Mock, T. Nakajima, and S. Moody, editors,

Proceedings of the 2nd IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (WSTFEUS 2004), pages 104–108, Vienna, Austria, May 2004. IEEE Computer Society.

R. Hartley and F. Pipitone. Experiments with the subsumption architecture. In *Proceedings 1991 IEEE International Conference on Robotics and Automation*, volume 2, pages 1652–1658. IEEE Computer Society Press, April 9-11 1991.

H. R. Hartson. Cognitive, physical and perceptual affordances in interaction design. *Behaviour and Information Technology*, 22(5):315–338, 2003.

H. Heft. Affordances and the body: An intentional analysis of gibson's. ecological approach to visual perception. *Journal of the Theory of Social Behavior*, 19:1–30, 1989.

H. Heft. *Ecological Psychology in Context: James Gibson, Roger Barker and the Legacy of William James's Radical Empiricism.* Lawrence Erlbaum Associates, Inc., Mahwah, NJ, 2001.

F. Heintz, P. Doherty, B. Wingmann, P. Rudol, and M. Wzorek. A software prototype for affordance support – The Entity Structure Generation Module (ESGM). Deliverable D4.3.2, MACS Internal Technical Report, Linköpings Universitet (LiU/IDA), Linköping, Sweden, September 2006. Draft, version 1.

D. Holz. Kontinuierliche Umgebungskartographie mittels 3D-Laserscanner auf autonomen mobilen Robotern. Diploma thesis, University of Applied Sciences Cologne, July 2006.

G. L. Humphreys. Objects, affordances ...action! *Nature Neuroscience*, 4:84–88, January 2001. At the 2000 London Conference Glyn Humphreys gave his Presidents' Award Lecture on the cognitive neuroscience of action selection.

IAIS-3DLS. 3D laser scanner, 2006. URL `http://www.3d-scanner.net`.

G. Indiveri. Kinematic Time-invariant Control of a 2D Nonholonomic Vehicle. In *Proceedings of the 38th Conference on Decision and Control, (CDC '99)*, pages 2112–2117, Phoenix, USA, December 1999.

H. Jaeger and T. Christaller. Dual dynamics: Designing behavior systems for autonomous robots, 1998.

K. S. Jones. What is an affordance? *Ecological Psychology*, 15(2):107–114, 2003.

D. Kirsh. Today the earwig, tomorrow man? *Journal of Artificial Intelligence*, 47(1-3): 161–184, 1991.

K. Koffka. *Principles of Gestalt Psychology.* Harcourt Brace, New York, 1935.

K. Konolige. COLBERT: A Language for Reactive Control in Saphira. In *Proceedings of 1997 German Conference on Artificial Intellgence*, pages 31–52, Freiburg, Germany, 1997.

K. Konolige and K. Myers. The saphira architecture for autonomous mobile robots. In D. Kortenkamp, R. Bonasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, MIT Press, pages 211–242, Cambridge, MA, USA, 1998.

K. Konolige, K. L. Myers, E. Ruspini, and A. Saffiotti. The Saphira Architecture: A Design for Autonomy. *Journal of Experimental & Theoretical Artificial Intelligence (JETAI)*, 9(1):215–235, 1997.

D. N. Lee. The optic flow field: The foundation of vision. *Royal Society of London Philosophical Transactions Series B*, 290:169–178, July 1980.

K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann. About the control of high speed mobile indoor robots. In *Proc. of the Second European Conference in Mobile Robotics*, pages 218 – 223, Ancona, Italy, 2005.

K. F. MacDorman. Grounding symbols through sensorimotor integration. *Journal of the Robotics Society of Japan*, 17(1):20–24, 1999.

K. F. MacDorman. Responding to affordances: Learning and projecting a sensorimotor mapping. In *Proc. of 2000 IEEE Int. Conf. on Robotics and Automation*, pages 3253–3259, San Fransisco, California, USA, 2000.

MACS. Webpage, 2004. URL `http://www.macs-eu.org`.

P. Makowski. Survey on architectures and frameworks for autonomous robots. In *Proc. Agrobotics Workshop*, Aalborg, November 2004.

M. J. Matarić. Behavior-based robotics. In Robert A. Wilson and Frank C. Keil, editors, *The MIT Encyclopedia of Cognitive Sciences*, pages 74–77. MIT Press, April 1999.

Merriam-Webster. Online dictionary, 2006. URL `http://www.m-w.com`.

G. Metta and P. Fitzpatrick. Early integration of vision and manipulation. *Adaptive Behavior*, 11(2):109–128, June 2003.

C. F. Michaels. Information, Perception, and Action: What should ecological psychologists learn from Milner and Goodale (1995)? *Ecological Psychology*, 12:241–258, 2000.

C. F. Michaels. Affordances: Four points of debate. *Ecological Psychology*, 15(2):135 148, 2003.

R. R. Murphy. Case studies of applying gibson's ecological approach to mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(1):105–111, January 1999.

R. R. Murphy. *Introduction to AI Robotics*. Intelligent Robots and Autonomous Agents. MIT Press, Cambridge, MA, USA, 2000. ISBN 0262133830.

B. S. Müller and I. Stratmann. Exploiting action and function knowledge in computer vision and robotics: The case of gibson's affordances. MACS Internal Technical Report, Fraunhofer Institute for Autonomous Intelligent Systems (FhG-AIS), Sankt Augustin, Germany, 2004.

U. Neisser. *Cognition and Reality: Principles and Implications of Cognitive Psychology.* Freeman, 1976.

D. A. Norman. *The Design of Everyday Things.* MIT Press, 2002. Reprint, 2002 edition. Originally published: The Psychology of Everyday Things, 1988.

D. A. Norman. Affordance, conventions, and design. *Interactions*, 6(3):38–42, 1999.

Donald A. Norman and Tim Shallice. Attention to action: Willed and automatic control of behavior. In R. J. Davidson, G. E. Schwartz, and D. Shapiro, editors, *Consciousness and Self-Regulation: Advances in Research and Theory*, pages 1–17. Plenum Press, New York, 1986.

A. Nüchter, H. Surmann, K. Lingemann, and J. Hertzberg. Semantic scene analysis of scanned 3D indoor environments. In *Proceedings of the 8th International Fall Workshop Vision, Modeling, and Visualization (VMV'03)*, pages 215 – 222, Munich, Germany, November 2003. IOS Press.

A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. Heuristic-based laser scan matching for outdoor 6D slam. In *KI*, pages 304–319, 2005a.

A. Nüchter, K. Lingemann, J. Hertzberg, H. Surmann, K. Pervölz, M. Hennig, K. R. Tiruchinapalli, R. Worst, and T. Christaller. Mapping of rescue environments with Kurt3D. In *Proceedings of the IEEE International Workshop on Rescue Robotics (SSRR '05)*, Kobe, Japan, June 2005b.

A. Nüchter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, and H. Surmann. 3D mapping with semantic knowledge. In *Proceedings of the RoboCup International Symposium 2005*, Osaka, Japan, July 2005c.

E. S. Reed. *Encountering the World - Toward an Ecological Psychology.* Oxford University Press, 1996.

D. Remondini and A. Saffiotti. A modular, hierarchical, reconfigurable controller for autonomous robots. In *Proc. of the IEEE Intl. Conf. on Methods and Models in Automation and Robotics*, pages 585–590, Miedzyzdroje, Poland, 2006.

E. Rome. Multi-Sensory Autonomous Cognitive Systems Interacting with Dynamic Environments for Perceiving and Using Affordances - Part B Scientific and Technical Description. MACS Internal Technical Report, Fraunhofer Institute for Autonomous Intelligent Systems (FhG-AIS), Sankt Augustin, Germany, October 2003.

E. Rome. MACS Periodic Activity Report 1. MACS Internal Technical Report, Fraunhofer Institute for Autonomous Intelligent Systems (FhG-AIS), Sankt Augustin, Germany, October 2005.

E. Rome, E. Şahin, R. Breithaupt, J. Irran, F. Kintzler, L. Paletta, M. Çakmak, E. Uğur, G. Üçoluk, M. R. Doğar, P. Rudol, G. Fritz, G. Dorffner, P. Doherty, M. Wzorek, H. Surmann, and C. Lörken. Development of an affordance-based control architecture. Deliverable D2.2.2, MACS Internal Technical Report, Institute for Intelligent Analysis and Information Systems (FhG-IAIS), Sankt Augustin, Germany, July 2006a. Draft version 1.

E. Rome, J. Hertzberg, G. Dorffner, and P. Doherty, editors. *Towards Affordance-based Robot Control*, Proceedings of Dagstuhl Seminar 06231, 2006b. To appear.

E. Rome, L. Paletta, G. Fritz, H. Surmann, S. May, and C. Lörken. Multi-sensor affordance recognition. Deliverable D3.2.1, MACS Internal Technical Report, Institute for Intelligent Analysis and Information Systems (FhG-IAIS), Sankt Augustin, Germany, August 2006c.

R. J. Ross. Marc – applying multi-agent systems to service robot control. Master's thesis, University College Dublin, Dublin, Ireland, 2003.

A. Saffiotti and K. LeBlanc. Active perception anchoring of robot behavior in a dynamic environment. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA-2000)*, pages 3796–3802, San Francisco, CA, April 2000.

C. M. Seifert. Situated cognition and learning. In R. A. Wilson and F. C. Keil, editors, *The MIT Encyclopedia of the Cognitive Sciences*, pages 767–769. MIT Press, 1999.

M. G. Slack. Sequencing formally defined reactions for robotic activity: integrating raps and gapps. In *Proceedings of SPIE's workshop on Sensor Fusion*, 1992.

A. C. Slocum, D. C. Downey, and R. D. Beer. Further experiments in the evolution of minimally cognitive behavior: From perceiving affordances to selective attention. In J. Meyer, A. Berthoz, D. Floreano, H. Roitblat, and S. Wilson, editors, *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, pages 430–439, Paris, France, 2000. MIT Press. September.

B. C. Smith. Situatedness/embeddedness. In R. A. Wilson and F. C. Keil, editors, *The MIT Encyclopedia of the Cognitive Sciences*, pages 769–771. MIT Press, 1999.

T. A. Stoffregen. Affordances and events. *Ecological Psychology*, 12(1):1–28, 2000.

A. Stoytchev. Toward a behavior-grounded representation of tool affordances. In L. Berthouze, H. Kozima, C. G. Prince, G. Sandini, G. Stojanov, G. Metta, and C. Balkenius, editors, *Proceedings of the Fourth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, pages 153–154, Genoa, Italy, 2004.

A. Stoytchev. Behavior-grounded representation of tool affordances. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3071–3076, April 2005a.

A. Stoytchev. Autonomous learning of tool affordances by a robot. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, Pennsylvania, July 9-13 2005b.

H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Journal Robotics and Autonomous Systems*, 45(3-4):181–198, 2003.

H. Surmann, K. Pervoelz, A. Nüchter, K. Lingemann, J. Hertzberg, and M. Hennig. Simultaneous mapping and localization of rescue environments. In *it- Information Technology*, number 47, pages 282–291. Oldenbourg press, October 2005.

M. T. Turvey. Affordances and prospective control: An outline of the ontology. *Ecological Psychology*, 4(3):173–187, 1992.

P. Ulam and R. C. Arkin. Biasing behavioral activation with intent. Technical Report GIT-GVU-06-11, Georgia Institute of Technology, College of Computing, GVU Center, 2006.

A. H. Vera and H. A. Simon. Sitauted action: A symbolic interpretation. *Cognitive Science*, 17:7–48, 1993.

W. H. Warren. Perceiving affordances: visual guidance of stair climbing. *Journal of Experimental Psychology. Human Perception and Performance*, 10(5):683–703, 1984.

R. A. Wilson and F. C. Keil, editors. *The MIT Encyclopedia of the Cognitive Sciences.* MIT Press, 1999.

R. Worst. KURT2 – a mobile platform for research in robotics. In U. Rückert and J. Sitte, editors, *Proceedings of the 2nd International Symposium on Autonomous Minirobots for Research and Edutainment*, pages 3–12, Brisbane, Australia, 2003. Queensland University of Technology.