Motivated Control of Multiple Reactive Architectures

James Gwatkin, Darryl Davis
Department of Computer Science
University of Hull
Cottingham Road
Hull HU6 7RX, U.K.
{J.Gwatkin, d.n.davis}@dcs.hull.ac.uk

Abstract

The use of motivations provides a useful method for developing autonomous robots. This paper describes an architecture, currently under development, that uses motivation to aid the control of a mobile robot. The proposed architecture contains a set of reactive components, a set of deliberative components and seeks to investigate some key issues. The first and most general issue is how can motivations be integrated within a mobile robot architecture to aid control. There are however some more specific problems that need to be addressed. One such problem is how can a robot identify objects within its environment, also known as the anchoring problem. Another issue arises from the use of multiple sensors. Can sensory fusion provide a better description of the robots environment and if so how is that achieved? The final issue to be investigated is how the can the deliberative component control the reactive component to achieve its goals.

1. Introduction

Mobile robots provide an essential tool when investigating the interaction of cognitive architectures and the physical environment. Robots have been used to investigate many different aspects of artificial intelligence such as mapping and localization techniques (Stachniss and Burgard, 2005, Wolf and Sukhatme, 2005), robot perception and navigation (Fiala and Basu, 2004, Marques et al., 2002) and robot learning (Hougen et al., 1996).

One area of interest is the use of motivation in the development of autonomous robots. The obvious question is how can motivation be incorporated into a mobile robot architecture. There have been various different methods proposed to achieve this (Marques et al., 2002, Stoytchev and Arkin, 2001). These architectures generally use a motivational subsystem to monitor the robots internal state and/or responses in its perceptual system, decide on the best course of action and then configure the robots behaviour accordingly.

The architecture described here differs in that motivations are always present. The internal parameters of the motivations can be changed by any part of the architecture at any time, but they do not monitor the system and wait for that change. Another main difference is that the motivations do not directly control or modify the robots' external behaviour, they merely trigger certain processes within the robot architecture. The use of motivation in this way will be explained in a little more detail in a later section. The rest of this section will introduce some of the key ideas and previous research that has influenced the design of the architecture presented here.

1.1 Mind as a Control System

An important aspect to the design of the architecture is the view that mind is a control system (Sloman, 1993). Mind as a control system takes the view that mind is a collection of many different control state subsystems passing data between them asynchronously as opposed to a single computation that can be reduced to a single state at any given time.

In essence control states are general behaviour within the agent. Control states can exhibit external behaviour, such as an obstacle avoidance mechanism, or reflect and control internal states such as hunger. The advantage of this approach to mind is that each control processes can be isolated and investigated. Each control process can be tested on quantifiable quantities such as speed and accuracy. For example, an obstacle avoidance behaviour on a robot could be implemented using different sensor modalities such as vision or sonar. Each control process can be tested in various different environments to determine the environments where vision performs better than sonar and vice versa.

1.2 Motivation

From a psychology viewpoint, motivation is the driving force behind all actions of an organism (Beck, 2000, Toates, 1998). If it is taken that actions are performed to achieve a positive internal state, then motivation is the search for positive internal states and the avoidance of negative ones. Positive internal states can be taught, learnt or are pre-programmed into the agent. The importance of a specific motivation is dependent on internal factors (hunger) and external factors (presence of food).

From the viewpoint of mind as a control system, motivation can be thought of as a control process (Sloman, 1993, Davis, 2001). A motivator often includes the following components (Sloman et al., 1994).

- Semantic content, a proposition P that represents a possible state that might be true or false.
- A motivational attitude to P e.g. make true, make false etc.
- A rationale if the motivator arose from explicit reasoning.
- An indicator of the current belief about P e.g. true, false etc.
- An importance value such as high, low etc.
- At the deliberative level, a plan or set of plans for achieving a motivator.
- A commitment status such as adopted, rejected, undecided.
- A dynamic state e.g. being considered, nearing completion etc.

Motivation as a control process can take one of several sub-states (Davis, 2001); Attitudes which are pre-programmed responses to specific sensory or internal cues; desires that underpin goals and provide a general preferred internal state; and goals that are specific tasks that the agent needs to achieve. Much of the previous work done on motivation as a control process (Davis, 1996, Davis, 2001, Nunes, 2001, Bourgne, 2003) has involved agents in simulated environments. It is therefore important to see whether this concept of motivation can translate from a simulated agent to a situated and embedded robot.

2. Reactive Components

Reactive architectures are a simple yet effective way of generating behaviours that allow robots to interact with a physical environment in real time. By linking responses in the robots perceptual system directly to its actuators, the need to generate an internal representation of the environment is eliminated. The following section discusses the various different reactive architectures implemented including a brief description of the omni-directional vision system that has been added to the robot.

2.1 Vision System

The robot is an old amigobot from Active Media. It has eight sonar sensors and communicates with a desktop PC via a radio modem. In addition the robot has had a cheap web-cam bolted on to the top with a spherical mirror to provide an omni-directional vision system that gives the robot a 360° field of view. The web-cam is connected to the desktop PC via a USB cable (figure 1).

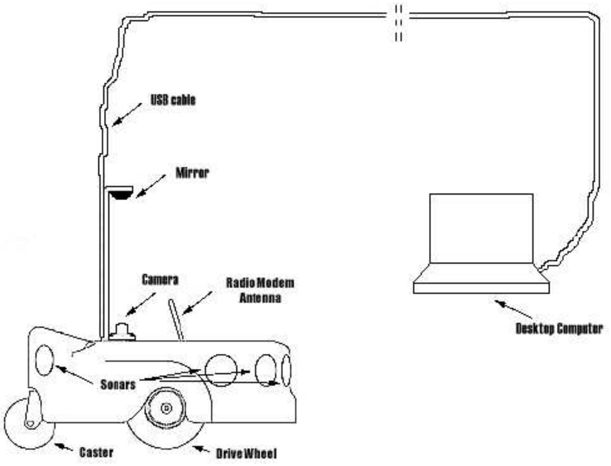

Figure 1: Robot with added omni-directional vision system.

The vision system works by passing a Sobel edge detection filter over the image. The image is then separated into segments of 5° surrounding the robot. Each segment is then searched for any values that exceed a specific threshold. The default value for this threshold can be modified by the robot architecture as required. By assuming that the area directly next to the robot is the floor, any detection can be considered the leading edge of an object (figure 2). Leading edges can be used as cues for obstacle avoidance or the basis for anchoring as explained in the following sections.

2.2 Reactive Architectures

Several simple behaviours, such as *avoid obstacle* and *emergency stop and turn*, have been implemented on the robot. As there are two sensor modalities, the behaviours can operate using the vision system or the sonar sensors. At this level there is no sensory fusion of the sonar and vision system. For example the *avoid object using sonar* and *avoid object using vision* behaviours operate independently of each other. This means that the avoid object behaviour can operate on three levels, sonar only, vision only or a combination of the two.

On top of this, four different methods for combining the various behaviours have been developed. The first is a priority method. In this architecture a priority is given to each behaviour before the robot is activated. The active behaviour with the highest priority is car-

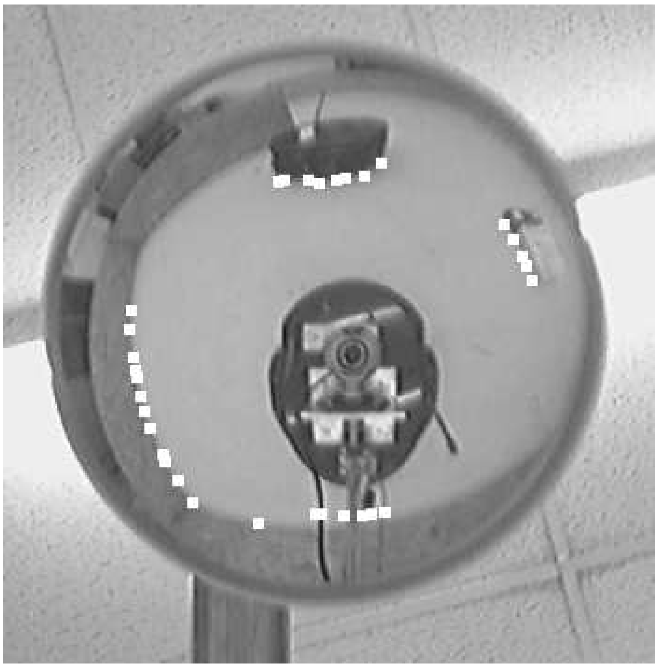

Figure 2: Leading edge detection using vision system.

ried out with all active behaviours with a lower priority being discarded. The second method is an aggregate architecture and uses a weighting system to determine the appropriate action. The weight of a behaviour is calculated depending on the response in the perceptual system according to a weighting function. Figure 3 shows the weighting function of the *avoid object* and *constant velocity* behaviour. The weight can vary between 0 and 1 and is dependent on the distance measured by the sensors to an object. The resulting action is a weighted average of all the behaviours. The third method is a weighted priority method. In this method each behaviour is assigned a weight in the same way as the weighted aggregate method. The resulting action is the behaviour with the largest weight. The final method uses a suppression architecture. Within this architecture behaviours can inhibit other behaviours, for example the *constant velocity* behaviour is always active, if the *avoid object* behaviour is activated then it inhibits the *constant velocity* behaviour.

The four different combination methods and three different sensor modalities provide twelve possible reactive architectures for moving the robot around in its environment. The question is why bother having so many different architectures that do the same thing. The answer is that each architecture may be more suited to a particular environment than another. The sonar system can process information faster than the vision system can. This makes it ideal for empty, uncluttered environments. However the sonar sensors are poor at picking up small objects that the vision system finds easily.

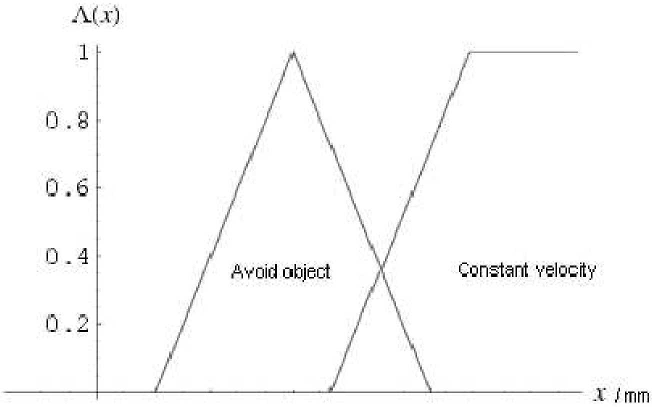

Figure 3: Behaviour weighting functions.

It is possible to design a single architecture that can utilise all sensor modalities and combination methods in the appropriate environment. However, if this approach is taken then the architecture would either need to recognise a change in environment, what that new environment was and then reconfigure itself, or a separate module that reasons about the environment would need to parameterise the behaviour to suit the new environment. It is hoped that switching between simple behaviours that are designed for one type of environment will prove more efficient and computationally faster than complex behaviours that require a great deal of parameterisation to cope with a new environment.

2.3 Experimentation

The large number of reactive architectures requires experiments to determine the most effective architecture within various different environments. To this end the various different architectures will be exposed to different environments and given a score depending on their performance. The environments will consist of an empty environment, an environment with a second amigobot present, one with a ball present, one with several small dynamic objects and also environments with various different combinations of these objects. The score will be based on a combination of the robots' average speed and the number of collisions. The score also provides default values for the affect model discussed in section 4.

2.4 Flexibility

One of the major advantages of using this type of architecture is its flexibility. New behaviours can be simply added to the architecture with little need for integration. If a particular behaviour does not conform to a particular combination method or sensor modality then it is simply not part of that architecture and is deactivated. A second advantage is the simplicity of configuring the architecture. Rather than configure each individual behaviour for the appropriate environment, only two parameters require alteration, the sensor modality and the combination method.

3. The Anchoring Problem

Anchoring is the way in which a correspondence is created and maintained between symbols and sensor data that refer to the same physical object. The anchoring problem is how to anchor objects using an artificial system (Coradeschi and Saffiotti, 2003). This section will discuss some of the possible methods used to anchor objects within the robots' environment.

3.1 Object Identification

Various different methods of identification are possible. The primary one used here will involve the vision system. By using the colour to identify the individual objects the use of complex shape models is avoided. The main assumption is that the colour of an object is invariant. Though this assumption may be inaccurate due to shadow and varying light conditions, by choosing objects with colours that contrast strongly with one and other, this effect is negligible.

Identification becomes more of a problem if the vision system is inactive. To identify a moving object using sonar, rather than use differential geometry, the robot is brought to a standstill. If there is a large change in one of the sonar sensors, it can be assumed that a dynamic object is present. Identification of stationary objects using sonar requires the robot to ram the object of interest. If the object remains stationary it can be assumed that the object is a wall. If however a dynamic object is subsequently detected then it can be assumed that the object is one of interest.

The design of the anchoring system was kept simple to dramatically reduce implementation time. However this simplicity can cause errors and erroneous readings could be labelled as objects of interest. Assuming the vision system is functioning correctly errors should be reduced to a minimum. It will be down to other components to recover from any errors caused by erroneous data. It should be noted however that even if the sophistication of the anchoring process was increased errors can still occur, for example an unfamiliar object introduced to the environment could be incorrectly classified.

3.2 Sensory Fusion

One technique that can aid the accuracy of an objects' description is by fusing the robots sensors. In this example the vision system has an accurate angular resolution but has trouble accurately determining the objects distance. Alternatively the sonar system can provide an accurate measure of the distance to an object but has a poor angular resolution. The senses are fused by simply associating the distance provided by the sonar reading at angle theta to an object found in the vision system at theta. Again due to the simplicity of the system errors can occur from overlapping sonar areas, reflection of the sound waves from corners and occasionally picking up the sonar clicks from the second amigobot. As with the identification method any false beliefs are dealt with by other components within the architecture.

4. Deliberation

All the systems described so far have been reactive, that is they have no memory of their last state, they only carry out their pre-programmed instructions according to their input to generate an output. Given that specific reactive architectures will have been identified to be effective in certain environments and/or appropriate for certain motivational states, a specific reactive control process can be activated or deactivated depending on these various internal and external factors. For the purpose of this research the selection of the current reactive architecture will be handled by deliberative processes.

Unlike many hybrid architectures, the deliberative components are not used every processing cycle, but only as required by the rest of the architecture. Hence the deliberative component, while always present, can be either static or dynamic. At start-up the deliberative processing is dynamic. It chooses a motivational goal, selects an appropriate reactive architecture for that goal, calls that reactive configuration, and then becomes dormant. When static, deliberative variables (such as the current motivator and its internal parameters) are held in the memory of the desktop computer and correspond to some part of state space. Hence if a deliberative process becomes dormant its variables remain constant but still remain accessible to other processes; acting as a memory of its last state. The reactive architecture is responsible for retriggering the deliberative component on success or failure of task. Once awakened the deliberative processing can use this feedback to simply pick up where it left off, choosing further goals or changing the reactive architecture to achieve some goal (or subgoal). The major parts of the deliberative processes used in this research are motivational constructs and belief structures. These are discussed in the remainder of this section.

4.1 Motivational Constructs

Rather than use motivations to directly control the robots actions or oversee the actions selected by other parts of the architecture, motivational constructs trigger processes that activate or deactivate various parts of the robots architecture. Each construct consist of variables that can be used to reflect the current internal state of the robot, the type and state of its environment, the sub-goals associated with any goal, the reactive architectures to use in achieving a goal, and other aspects of motivational behaviour as identified in earlier research (Davis, 1996, Davis, 2001, Nunes, 2001, Bourgne, 2003, Lewis, 2004). One example would be an explore environment motivation. The explore environment motivation would contain an affective affordance associated with each possible object of interest within the environment.

Affordances are properties offered to an agent by elements within its environment (Gibson, 1986). Fire affords warmth (positive affordance) and also burns (negative affordance). This example demonstrates that affordances are both objective and subjective as a fire does increase the temperature of the surrounding environment but this fact is immaterial to an agent that lives in a hot climate. Affective affordances are an extension to the theory of affordances. Earlier research (Bourgne, 2003, Lewis, 2004) developed a motivational control model for affective affordances that appeared to work well for agents in simulated environments. Affective affordance values change depending on the agents' internal and external states, including currently adopted goals. The higher the affordance value of an object the more important that object becomes to the agent. Negative experiences decrease affordances associated with objects. Specific reactive architectures will have different (inbuilt) success or failure conditions. In all cases, this trigger (or alarm) will cause the posting of data (perhaps a belief as described in the next section) to the deliberative level. This in turn triggers the deliberative architecture to update its state, potentially adapt the affect model in place and possibly reconfigure the active reactive architecture.

A specific instantiated example of an explore environment motivation might be associated with finding balls and hence balls, their perceptual models and ball recognition behaviour models would have increased affordances. This would lead to the construct activating a reactive architecture suited to finding and responding to this type of object. Alternatively there may be an opportunistic motivation present such as hit ball. The presence of a ball in the environment increases the importance of this motivation so that it supersedes the explore motivation. This in turn triggers a process to invoke the *hit ball* architecture. The affective model ensures that architectures and behaviours (and plans if present) that lead to successful interactions between motivational state, active architecture and environmental object are rewarded with an increased affordance. Unsuccessful associations have their affordances decreased. In such a manner, the entire architecture adapts from an initial state to one that is effective in the types of environment currently encountered. The current experiments will determine how well this model maps onto the multiple reactive with occasional deliberative processing model.

4.2 Beliefs

The use of a belief system can help the robot recover from some of the errors described in section 3. Each object relevant to the robot has an associated data structure. This structure contains various variables that reflect the presence of an object, the location of the object if present and the confidence in that belief. The variables within the structure are modified by the robots sensor processing modules. If the object identification modules detect an object the relevant belief structure is updated. It is at this point that the object is said to be anchored.

The confidence of the belief is determined by the affective correspondence value of the current sensor modality. An affective correspondence refers to the confidence of the robot that the sensor data is an accurate reflection of the environment (Lewis, 2004). If the visual identification of an object remains consistent over time the affective correspondence of the vision system is increased and so the robots confidence in its beliefs based on visual identification is increased. Alternatively if the identification of an object based on the sonar system is inconsistent over time, its associated affective correspondence is reduced and therefore so are the robots beliefs based on sonar identification.

The belief variables can also be modified by the motivational constructs. For example if the importance value of a hit ball motivation is high, then the confidence of the relevant belief is kept high to maintain that motivation, regardless of the lack of detection from the object identification modules. Only if the affect model is so changed that the affordances associated with that motivation are reduced will this change. Like wise the belief structure can also modify the motivational construct. If the object identification module continues to fail to find the ball then the confidence in that belief is reduced, affecting the importance value of the motivation or possibly changing its state from processing to failed. In this way motivations and beliefs can modify each other.

5. Real-Time Control

With all the elements in place the final task is to develop a method to control the flow of information within the architecture. At the deliberative level, a variation on the well-known belief desire intention (BDI) model is used (Bratman, 1987, Georgeff and Rao, 1995, Georgeff et al., 1999). The variation we are using builds on the CRIBB (Children's Reasoning with Intentions, Belief and Behaviour) cognitive model (Wahl and Spada, 2000) used in earlier work (Lewis, 2004). In this the agent has a set of data structures that contain variables relating to a specific belief, the confidence in that belief and the confidence in the source of that belief. The semantic content of the structure is modified by the robots sensor processing systems and the agent's motivational constructs. The confidence (an aspect of the affect model described above in the context of the motivational constructs) can be modified by the belief update process or through actions associated with reasoning about desires and goals. The agent's desires or goals are represented by the motivational constructs as described above. The content of the constructs reflect the current state of that motivation, such as its affect qualities and the plans, behaviours or actions that allow the desire or goals to be realised. Changes in beliefs and affect result in changes to the set of viable motivations and shifts in the nature of each motivational construct, and so reflect changes in the currently selected desire. The final element of the BDI model is the agents intention. At the deliberative level this is the plans or behaviour or action or architecture held in the current active (set of) motivational construct(s). In the current experiment this is then represented as the current configuration of the reactive architecture. Depending on the current state of the set of possible motivational constructs, and the affordances associated with those motivations and the possible reactive configurations, a specific reactive architecture is triggered to achieve some goal.

5.1 Architecture Overview

The overall architecture operates as follows. The agent starts in its default architecture, *avoid* and *explore* behaviors for example. Sensor information and success/fail messages from that architecture modify and update the agents belief set. These changes, if appropriately significant, cascade through the motivational construct set and affordance models. The success of the reactive architecture (in finding the ball for example) improves the affordance offered by that architecture for finding the ball in future. Failure to accomplish some task reduces the affordance of that architecture for that task. Significant changes in the belief or perceptual models activate the deliberative architecture. Once the deliberative aspect of the overall architecture is triggered, the agent's beliefs and motivations can interact with each other, and the affordance model, causing the activation of an alternative reactive architecture depending on their current state. The affect model may change over time as the robot responds to its own attempts to perform similar tasks in different environments, and may provide the means by which the robot itself can learn how to prioritise between behaviours rather than rely on the values given in human guided experimentation. In this method there is no single module that monitors and controls the robots behaviour, control is distributed over the entire architecture. This means that rather than reducing the system to a single state to make a decision, the process is more fluid and the agent's beliefs and motivations ebb and flow to reflect its environment and internal state.

6. Conclusion

The architecture described seeks to demonstrate a method in which motivations can be integrated effectively within a cognitive architecture designed to control a robot. By using the approach of mind as a control system it is possible to determine how effective different approaches are to this problem. It is also hoped that the use of several motivational constructs to trigger one of a possible number of reactive architectures for a specific situation is more efficient than using a motivational module to parameterise each aspect of the reactive architecture to achieve the desired goal. The adopted model may prove to be beneficial for demonstrating how a robot can adapt and learn to use different behaviours in different ways in different environments to achieve its goals.

References

Beck, R. (2000). *Motivation: Theories and Principles.* Pearson Education, 4 edition.

Bourgne, G. (2003). Affect-based multi agent architecture (for a 5-aside football simulation). Master's thesis, School of Computer Science, University of Hull.

Bratman, M. (1987). *Intentions, Plans and Practical Reason.* Harvord University Press.

Coradeschi, S. and Saffiotti, A. (2003). An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43(2-3):85–96.

Davis, D. (1996). Reactive and motivational agents. In Muller, J., Wooldridge, M., and Jennings, N., (Eds.), *Intelligent Agents III: Agent Theories, Architectures and Languages.* Springer-Verlag.

Davis, D. (2001). Control states and complete agent architectures. *Computational Intelligence*, 17(3):621–650.

Fiala, M. and Basu, A. (2004). Robot navigation using panoramic tracking. *Pattern Recognition*, 37(11):2195–2215.

Georgeff, M., Pell, B., Pollack, M., Tambe, M., and Wooldridge, M. (1999). The belief-desire-intention model of agency. In Müller, J., Singh, M., and Rao, A., (Eds.), *Intelligent Agents V: Agent Theories, Architectures and Languages*, pages 1–10. Springer-Verlag.

Georgeff, M. and Rao, A. (1995). The semantics of intention maintenance for rational agents. In *Proceedings of the 14th International Joint Conference* *on Artificial Intelligence*, volume 1, pages 704–710. Morgan Kaufmann.

Gibson, J. (1986). *The Ecological Approach to Visual Perception.* Lawrence Erlbaum.

Hougen, D., Fischer, J., Gini, M., and Slagle, J. (1996). Fast connectionist learning for trailer backing using a real robot. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1917–1922.

Lewis, S. (2004). *Computational Models of Emotion and Effect.* PhD thesis, School of Computer Science, University of Hull.

Marques, L., Nunes, U., and Almeida, A. (2002). Olfaction-based mobile robot navigation. *Thin Solid Films*, 418(1):51–58.

Nunes, H. (2001). Investigation of motivation in agents using the simulation of 5-aside football. Master's thesis, School of Computer Science, University of Hull.

Sloman, A. (1993). The mind as a control system. In Hookway, C. and Peterson, D., (Eds.), *Philosophy and Cognitive Science.* Cambridge University Press.

Sloman, A., Beaudoin, L., and Wright, I. (1994). Computational modelling of motive-management processes. In Frijda, N., (Ed.), *Proceedings of the Conference of the International Society for Research in Emotions.* ISRE Publications.

Stachniss, C. and Burgard, W. (2005). Mobile robot mapping and localization in non-static environments. In *In Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 1324–1329.

Stoytchev, A. and Arkin, R. (2001). Combining deliberation, reactivity, and motivation in the context of a behavior-based robot architecture. In *In Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 290–295.

Toates, F. (1998). *Control of Behaviour.* Springer-Verlag.

Wahl, S. and Spada, H. (2000). Children's reasoning about intentions, beliefs and behavior. *Cognitive Science Quarterly*, 1:5–34.

Wolf, D. and Sukhatme, S. (2005). Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19(1):53–65.