Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Reflective Navigation: Individual Behaviors and Group Behaviors

Boris Kluge
InMach Intelligente Maschinen GmbH
Helmholtzstr. 16, 89081 Ulm, Germany
kluge@inmach.de

Erwin Prassler
Bonn-Rhein-Sieg University of Applied Science
Grantham-Allee 20, 53757 Sankt Augustin, Germany
erwin.prassler@fh-brs.de

Abstract— **An approach to motion planning among moving obstacles is presented, whereby obstacles are modeled as intelligent decision-making agents. The decision-making processes of the obstacles are assumed to be similar to that of the mobile robot. A probabilistic extension to the velocity obstacle approach is used as a means for navigation and modeling uncertainty about the moving obstacles' decisions.**

Index Terms— **mobile robot navigation, opponent modeling, intention reasoning**

I. INTRODUCTION

Automatic guidance of a robotic system is commonly seen as the problem of planning a collision-free motion through the surrounding space taking into account geometric properties and various physical constraints. Often this view is perfectly sufficient to operate the robotic system, especially, if the system only needs to move in a structured workspace without interference with other agents. However, we question that this type of collision-free locomotion qualifies as *intelligent locomotion*. This holds true in particular, if we expect the robotic system to permanently operate in a human-inhabited natural environment and to co-exist and co-operate with humans in an intuitive, human-friendly way.

Consider a situation where two persons want to pass through a doorway from opposite sides. If both persons are equally polite they will give way to each other. This will lead to a deadlock situation where both persons wait for the other person to pass the doorway first. It often happens that both people - noticing that the opponent gives way - make another attempt to pass the door at the same time and again block each others way. Humans are able to resolve such conflict situations even if it occasionally takes some time. They are able to do so because they do not only take geometric and other physical properties into account while moving, but also reflect on the opponents intention.

To account for such non-physical aspects of locomotion and navigation we have introduced the concept of *reflective navigation*. Reflective navigation combines two mechanisms. It accounts for the uncertainty in the perception and prediction of the motion behavior of other agents by introducing probabilistic envelopes for their velocity and heading (*probabilistic velocity obstacles*). It furthermore employs a *recursive model* of the behavior of the opponents. 'Recursive means that the model does not only involve a prediction of the opponents decision, but also foresees that in its decision making the opponent itself reflects on our robots behavior. Recursive modeling in a nutshell involves the following steps:

1) perceiving the surrounding environment and agents
2) modeling the perception of these agents
3) modeling the decision making of these agents
4) prediction of action of these agents
5) making own decision based on agents models
6) performing action

The concept of reflective navigation is described in more detail in [3]. In this paper we give only a brief recap of the underlying theoretical concepts and then we will results of experiments in which reflective navigation was applied by a group of agents. The variation in the behaviors reaching from aggressive and careless locomotion to moderate and amicable locomotion in fact only resulted from the different levels of reflection employed by the various agents.

The paper is organized as follows. Sections II and III introduce the velocity obstacle approach and its probabilistic extension. Recursive agent modeling forms the basis for the reflective navigation proposed in this paper and is presented in Section IV. The correspondence between recursive depth and level of reflection is explained in Section V. A set of experiments illustrate the approach in Section VI, and the discussion in Section VII precedes the concluding Section VIII.

II. VELOCITY OBSTACLES

This section gives a brief introduction to the original velocity obstacle approach [2].

Let B_i and B_j be circular objects with centers c_i and c_j and radii r_i and r_j, moving with constant velocities $v_i = \dot{c}_i$ and $v_j = \dot{c}_j$. To decide if these two objects are on a collision course, it is sufficient to consider their current positions together with their relative velocity $v_{ij} = v_i - v_j$, see Fig. 1. Let

$$\hat{B}_{ij} = \{c_j + r \,:\, r \in \mathbb{R}^2, |r| \leq r_i + r_j\}, \qquad (1)$$

$$\lambda_{ij}(v_{ij}) = \{c_i + \mu v_{ij} \,:\, \mu \geq 0\}. \qquad (2)$$

Then B_i and B_j are on a collision course, if and only if $\hat{B}_{ij} \cap \lambda_{ij}(v_{ij}) \neq \emptyset$.

0-7803-8232-3/04/$17.00 ©2004 IEEE

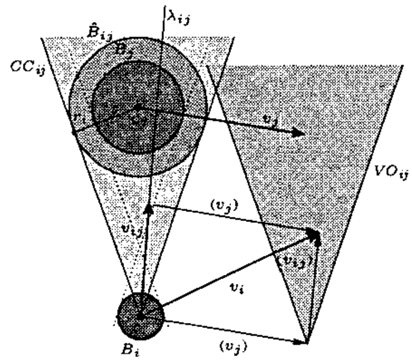

Fig. 1. Collision cone and velocity obstacle

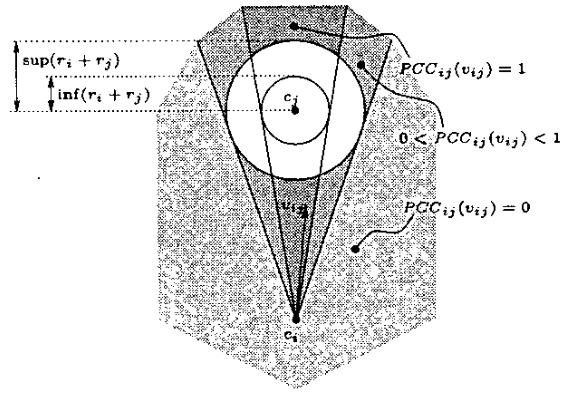

Fig. 2. Probabilistic collision cone

Therefore we can define a set of colliding relative velocities, which is called the *collision cone* CC_{ij}, as

$$CC_{ij} = \{v_{ij} \,:\, \hat{B}_{ij} \cap \lambda_{ij}(v_{ij}) \neq \emptyset\}. \tag{3}$$

In order to be able to decide if an absolute velocity v_i of B_i leads to a collision with B_j, we define the *velocity obstacle* induced by B_j for B_i to be the set

$$VO_{ij} = \{v_i \,:\, (v_i - v_j) \in CC_{ij}\}, \tag{4}$$

which is, in other words,

$$VO_{ij} = v_j + CC_{ij}. \tag{5}$$

Now for B_i, any velocity $v_i \in VO_{ij}$ will lead to a collision with B_j, and any velocity $v_i \notin VO_{ij}$ for B_i will avoid any collision with B_j.

In general, object B_i is confronted with more than one other moving object. Let $\mathcal{B} = \{B_1, \ldots, B_n\}$ the set of moving objects under consideration. The *composite velocity obstacle* induced by $\mathcal{B}$ for B_i is defined as the set

$$VO_i = \cup_{j \neq i} VO_{ij}. \tag{6}$$

For any velocity $v_i \notin VO_i$, object B_i will not collide with any other object.

Finally, a simple navigation scheme based on velocity obstacles can be constructed as following. The moving and non-moving obstacles in the environment are continuously tracked, and the corresponding velocity obstacles are repeatedly computed. In each cycle, a reachable velocity is chosen which avoids collisions and approaches a motion goal, for example a maximum velocity towards a goal position, or a velocity minimizing the difference to a target velocity.

III. PROBABILISTIC VELOCITY OBSTACLES

Let B_i and B_j be circular objects with centers c_i and c_j and radii r_i and r_j as in the previous section. However, now we will consider uncertainty in shape and velocity of the objects. This allows to reflect the limitations of real sensors and object tracking techniques, i.e. it offers a method to cope with known noise in the agent's perceptual system.

A. Shape Uncertainty

We define the *probabilistic collision cone* of object B_j relative to object B_i to be a function

$$PCC_{ij} : \mathbb{R}^2 \to [0, 1] \tag{7}$$

where $PCC_{ij}(v_{ij})$ is the probability of B_i to collide with B_j if B_i moves with velocity v_{ij} relative to B_j. See Fig. 2 for an example where a probabilistic collision cone is induced by uncertainty in the exact radii of two circular objects.

B. Velocity Uncertainty

Furthermore, there may be uncertainty about object velocities, for example introduced by a finite tracking rate. Therefore we represent the velocity of object B_j by a probability density function

$$V_j : \mathbb{R}^2 \to \mathbb{R}_0^+. \tag{8}$$

Now the *probabilistic velocity obstacle* of object B_j relative to object B_i is defined as the function

$$PVO_{ij} : \mathbb{R}^2 \to [0, 1] \tag{9}$$

which maps absolute velocities v_i of B_i to the according probability of colliding with B_j. It is

$$PVO_{ij}(v_i) = \int_{\mathbb{R}^2} V_j(v_j) PCC_{ij}(v_i - v_j) d^2 v_j \tag{10}$$

which is equivalent to

$$PVO_{ij} = V_j * PCC_{ij} \tag{11}$$

where $*$ denotes the convolution of two function. Notice the structural similarity to (5).

C. Composite Probabilistic Velocity Obstacle

The probability of B_i colliding with any other obstacle when traveling with velocity v_i is the probability of not avoiding collisions with any other moving obstacle. Therefore we may define the *composite probabilistic velocity obstacle* for B_i as the function

$$PVO_i = 1 - \prod_{j \neq i} (1 - PVO_{ij}). \tag{12}$$

D. Navigating with Probabilistic Velocity Obstacles

In the deterministic case, navigating is rather easy since we consider only collision free velocities and choose a velocity which is optimal for reaching the goal. But now, we are confronted with two competing objectives: reaching a goal and minimizing the probability of a collision.

Let $U_i : \mathbb{R}^2 \rightarrow [0, 1]$ a function representing the utility of velocities v_i for the motion goal of B_i. Since the full utility of a velocity v_i is only attained if (a) v_i is dynamically reachable, and (b) v_i is collision free, we define the *relative utility function*

$$RU_i = U_i \cdot D_i \cdot (1 - PVO_i), \tag{13}$$

where $D_i : \mathbb{R}^2 \rightarrow [0, 1]$ describes the reachability of a new velocity.

Now a simple navigation scheme for object B_i based on probabilistic velocity obstacles is obtained by repeatedly choosing a velocity v_i which maximizes the relative utility RU_i.

IV. RECURSIVE PROBABILISTIC VELOCITY OBSTACLES

Traditionally, when navigating a mobile robot among moving obstacles (like humans), these obstacles' abilities to avoid collisions and their resulting motion behaviors are not taken into account. In contrast to this plain obstacle modeling, recursive modeling techniques presume the opponents (or more generally, the interaction partners) to deploy decision making processes for navigation similar or equal to the approach for the robot.

A. Obstacle Modeling

Obstacles are assumed to perceive their environment and deduce according reactions, the reasoning process being similar to that of the robot. That is, any obstacle B_j is assumed to take actions maximizing its relative utility function RU_j. Therefore, in order to predict the action of obstacle B_j, we need to know its current utility function U_j, dynamic capabilities D_j, and velocity obstacle PVO_j.

The utility of velocities can be inferred by recognition of the current motion goal of the moving obstacle. For example, Bennewitz et al. [1] learn and recognize typical motion patterns of humans. If no global motion goal is available through recognition, one can still assume that there exists such a goal which the obstacle strives to approach, expecting it to be willing to keep its current speed and heading.

By continuous observation of a moving obstacle it is also possible to deduce a model of its dynamics, which describes feasible accelerations depending on its current speed and heading.

Finally, the velocity obstacle PVO_j for obstacle B_j is simply computed using the world and self models of robot B_i, assuming similar perception among the objects.

Using this information, the future motion of an obstacle can be predicted by deriving a probabilistic description of its future velocity from its relative utility function.

B. Recursive Modeling

Having such obstacle models as described above, the predicted obstacle velocities can be used to replace the observed velocities in the robot's own obstacle avoidance scheme. Such higher levels of reasoning in the navigation process will be denoted by an according superscript, i.e. we will write RU_i^1 for a relative utility function of from the plain (probabilistic) velocity obstacle approach, RU_i^2 for a relative utility where the velocity of each obstacle B_j is deduced from its assumed relative utility functions RU_j^1, and so on. Generally speaking, in order to obtain RU_i^d, obstacle velocities are deduced from RU_j^d for obstacles B_j.

The conversion from relative utilities to (probabilistic) velocities is accomplished by a simple normalization

$$V_i^d = norm\left(RU_i^d\right) \tag{14}$$

which expresses the assumption that objects will move according to their relative utility function. Special care has to be taken for the case when the involved integral vanishes, i.e. the scaling factor would become infinite (see [3]).

This reflective modeling step is not restricted to recursion of depth one by a matter of principle. However, computational demands will increase with this depth of the recursion, and intuitively one does not expect recursion depths of more than two or three to be of broad practical value, since such deeper modeling is not observed when we are walking as human beings among other humans.

C. Implementation and Complexity

In order to implement the presented approach, the involved functions $\mathbb{R}^2 \rightarrow \mathbb{R}$ need to be discretized. Currently this is accomplished by a partitioning of $\mathbb{R}^2$ into cells, and an approximation by stair functions which are constant on each cell. This approach is simple and straightforward, but as a consequence the computational complexity depends on the amount of uncertainty. Let $\sigma(f)$ the set of cells needed to represent $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ by a step function (i.e. the set of cells where the value of the step function is non-zero), and $N_i := |\sigma(D_i) \cup \sigma(V_i)|$ the number of cells needed to represent the uncertain dynamic capabilities and velocity of an agent. Then one can show that the time complexity of the presented approach is

$$O\left(r \sum_{i=1}^{n} \left(N_i \sum_{j \neq i} N_j\right)\right) \tag{15}$$

where r denotes the maximum reasoning level and n the number of agents [3].

V. LEVELS OF REFLECTION

In the previous section we have introduced the concept of *relative utility function* and described its recursive formulation

$$RU_i^d = D_i \cdot U_i \cdot \prod_{j \neq i} (1 - norm\left(RU_j^{d-1}\right) * PCC_{ij}) \tag{16}$$

which is obtained as a combination of Equations 11, 12, 13, and 14. Reasoning about the relative utility of velocities and finding a velocity which maximizes the relative utility function on a recursion level d refers to the velocities on level $d-1$ maximizing the relative utility functions of all other agents in the surrounding. The recursion terminates at level $d = 0$ with $RU_i^0 = D_i \cdot U_i$, where a suitable velocity is selected only based on utility and reachability. We denoted these levels of recursion as *levels of reflection*. This is motivated through the fact, that each level of recursion leads to a distinct behavior which may be associated with a certain level of contemplation or reflection.

There are basically no limits regarding the levels of recursion; at least no theoretical limits. Our results, however, showed that for pragmatic reasons we only need to consider the following four levels of recursion (and reflection).

A. Level $d = 0$: No perception, no reflection

This level does not really have a "cognitive equivalent". It is rather technically motivated: it terminates the recursion.

$$RU_i^0 = D_i \cdot U_i \tag{17}$$

An agent employing this level does neither perceive the environment nor does it reflect on the behavior and decision of other agents. It will ignore other agents or objects and the probabilistic velocity obstacles imposed by them and select a velocity, which maximizes only utility and reachability. Again, this level has only a technical motivation.

B. Level $d = 1$: Perception without reflection

This level in fact does have a "cognitive equivalent". When employing this level of recursion and reflection, the robot perceives the environment and the objects and other agents therein and it also perceives and accounts for their locomotion (velocity and heading) and the corresponding probabilistic obstacle velocities. On this level the relative utility function is defined as follows:

$$\begin{aligned} RU_i^1 &= D_i \cdot U_i \cdot \prod_{j \neq i} (1 - PVO_{ij}^0) \\ &= D_i \cdot U_i \cdot \prod_{j \neq i} (1 - V_j^0 * PCC_{ij}). \end{aligned} \tag{18}$$

The behavior shown by an agent reasoning on this level is identical with the behavior shown by an agent, which uses the plain (probabilistic) velocity obstacle without recursion.

C. Level $d = 2$: Reflecting about other agents' intended behavior

By employing this level of recursion the agent really reflects about the intended behavior of other agents. The formulation of its own relative utility function involves the relative utility function of its opponents at level $d = 1$:

$$\begin{aligned} RU_i^2 &= D_i \cdot U_i \cdot \prod_{j \neq i} (1 - PVO_{ij}^1) \\ &= D_i \cdot U_i \cdot \prod_{j \neq i} (1 - norm\left(RU_j^1\right) * PCC_{ij}). \end{aligned} \tag{19}$$

Informally this means the robot expects the fellow agents in the surrounding environment to perceive the environment and to avoid collisions by selecting suitable velocities. So the robot in fact accounts for what he thinks that the other agents have decided to do.

An agent reasoning on this level actually shows a rather surprising behavior. Since it takes it for granted that its fellow agents can perceive the environment and avoid collisions it behaves itself in a rather ruthless and aggressive way. It pretends to accept the risk of a collision and enforces its right of way like a rude youngster with his new car (see Section Experiments).

D. Level 3: Reflecting about the other agents reflection

This level of recursion enables the agent not only to reflect about the behavior of other agents but also about the reasoning behind their behavior. In other words, the robot reflects about the other agents reflection. The corresponding relative utility function is:

$$\begin{aligned} RU_i^3 &= D_i \cdot U_i \cdot \prod_{j \neq i} (1 - PVO_{ij}^2) \\ &= D_i \cdot U_i \cdot \prod_{j \neq i} (1 - norm\left(RU_j^2\right) * PCC_{ij}). \end{aligned} \tag{20}$$

While the robot was aggressive employing level $d = 2$ it behaves rather moderate and amicable when contemplating on level $d = 3$. It assumes the other agent to show an aggressive and ruthless behavior and accordingly behaves wiser and more carefully.

As indicated above there is no technical bound for the levels of recursion and reflection, which our robot might employ in its reasoning. However, in our experiments we noticed, that for example an agent reflecting at a level $d = 4$ expects the other agents to behave wise and carefully and allows itself a more ruthless behavior showing a behaviors as if it was reasoning on level $d-2$. Similarly, a robot employing recursion level $d = 3$ shows a behavior as if it was employing recursion level $d - 2$. Accordingly, our reflective navigation approach usually employs only the first four levels of recursion.

VI. Experiments

In this section we are going to present some experimental results which illustrate the motion behaviors emerging from the reflective navigation approach described in the previous sections. We are first going to describe two experiments involving two individuals, which are on a collision course, and how they negotiate their way through individual reflection. Then we are going to compare this with the behaviors and the resulting locomotion of two groups of agents using the same levels of reflection. We would like to emphasize again that the resulting behavior only comes about through observing the other agents in the environment and through reflecting about their intentions and potential decisions and also about their levels of reflection. The agents involved in these experiments do not explicitly communicate with each other.

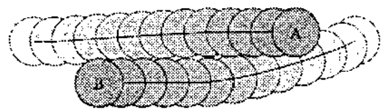

Fig. 3. Collision Course, 'A' at level 2 vs. 'B' at level 1

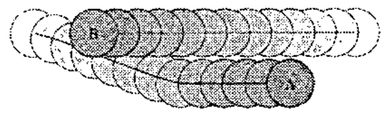

Fig. 4. Collision Course, 'A' at level 2 vs. 'B' at level 0

Figures 3 and 4 show a situation where two individual agents, one moving from the left to the right, the other from the right to the left, approach each other on a collision-course. In the experiment in shown Fig. 4 we assume that the left agent employs recursion level 2 in its reasoning, which means that it expects the opponent to be able to perceive the surrounding environment and avoid collisions. The assumption of the left agent is in fact justified. The agent on the right employs recursion level 1 for determining its locomotion, which means it will always try to move on a collision-free course. The resulting behavior is shown in Fig 3. Assuming that the agent coming from the right avoids a potential collision, the left agent continues to move along its original course in rather ruthless fashion and leaves it to the right agent to step out of the way. In the experiment in Fig. 4 we consider the same situation. This time, however, the right agent uses recursion level 0 for determining its course. It neither perceive the environment nor does it reflect in any way about the intention of other agents. It moves blindfold along a course, which it has determined as the most useful one. That means, the left agents assumption about the right agent's future behavior is wrong. Fig. 4 shows that the ruthless left agent eventually decides to step out of the way of the even more ruthless right agent, which keeps its original course, in spite of the collision danger.

In the two experiments in Figures 5 and 6 we consider two similar situations. This time, however, do not only involve two

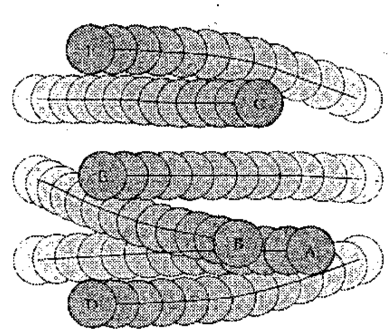

Fig. 5. Two groups on collision course, 'A', 'B', 'C' at level 2 vs. 'D', 'E', 'F' at level 1

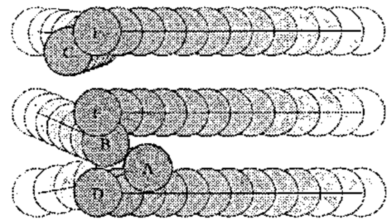

Fig. 6. Two groups on collision course, 'A', 'B', 'C' at level 2 vs. 'D', 'E', 'F' at level 0

individuals but two groups of individuals, A, B, C and D, E, F, approaching each other. Like in the first experiment, the left group reasons at recursion level 2 while the right group reasons at level 1 and hence at least perceives the environment and tries to avoid collisions. In Figure 5 we can observe that agent A and C keep stubbornly moving nearly on their original course. Agent F and D have to change their course to avoid a collision with A and C respectively. Since Agent E is caught between its two wingmen D and F and is furthermore approached by Agent A and C, it cannot help moving straight. Agent B, which under other circumstances would move as aggressively and stubbornly as Agent A and C, however, is able to reflect (recursion level 2) about the other agents' intention and future behavior. It decides to slow down and then follow Agent A.

In the last experiment again two groups of agents, A, B, C and D, E, F, approach each other. Agents A, B, and C reason on recursion level 2, while D, E, and F move blindfold like the right agent in the second experiment.

This time both groups of agents behave rather stubbornly and move rather aggressively. The result of their stubbornness is shown in Figure 6. Agent A collides with D, Agent B collides with E, and Agent C collides with F. Apparently too much stubbornness can hurt.

We would like to emphasis that although the above experiments are all simulations, the situations which are considered are quite natural and so is the behavior of the agents which emerge as a result of the reflective navigation approach described in the previous sections. The situation considered in the last experiment certainly has some similarity with the situation in an American football game and so has the emerging behavior of the agents. The situation in the third experiment has more of an Irish folk dance, where the dancers try hard not to run into each other.

VII. DISCUSSION

Finding good models of other object's dynamic capabilities D_j and utility functions U_j is a problem beyond the scope of this paper. When the action to be taken is considered the first step of a longer sequence, computing the utility function may involve motion planning, or even game-tree search, if reactions of other objects are taken into account. Due to the recursive nature of the approach, such a procedure would have to be applied for any object at any recursive level. This renders such enhancements of utility functions rather infeasible, since already single applications of such procedures are computationally expensive.

Oscillations appear in models for successive depths. Reconsider the collision course examples from Section VI with both objects facing each other. Assume at depth d, both objects avoid a collision by deviating to the left or to the right. Then in depth $d + 1$, none of the objects will perform an avoidance maneuver, since each object's depth-d model of the other object predicts that other object to avoid the collision. Subsequently, in depth $d + 2$, both objects will perform collision avoidance maneuvers again, an so on.

Another aspect of the presented approach is that it can serve as a basis for reasoning about the objects in the environment. One could compare the observed motion of the objects to the motions that are predicted by recursive modeling using different utility models (for example ranging from very defensive behavior up to a homicidal chauffeur), and classify them accordingly.

Finally, it is not yet clear to what extent human behavior can be modeled in the presented manner. However, the approach is considered a reasonable tradeoff between computational feasibility and more detailed modeling.

VIII. CONCLUSION

In this paper we presented a short overview of the theoretic concepts and some first experimental results of an approach which we denoted as *reflective navigation*. This approach is motivated by the insight that *intelligent locomotion* is more than path planning and obstacle avoidance based on geometric and other physical constraints. Human beings, when they move consciously, reflect about the intentions and potential future behaviors of other agents in the surrounding environment and occasionally even develop models of the reflection capabilities of these other agents.

Reflective navigation is a probabilistic approach which tries to account for the uncertainty inherent in natural (human) and artificial (machine) perception and decision making. The approach furthermore makes assumptions about the potential future behavior of other agents in the surrounding environment. As our first experimental results indicate the locomotion behavior emerging naturally from this approach shows some distinct similarity with the locomotion behavior of humans in similar situations.

The experimental results presented in the paper were obtained from a simulation, which clearly shows the performance of the approach. Much of our future work will go into an implementation of the approach on physical robots moving through natural crowded environments.

IX. ACKNOWLEDGMENTS

This work was supported by the German Department for Education and Research (BMB+F) under grant no. 01 IL 902 F6 as part of the project MORPHA.

REFERENCES

[1] M. Bennewitz, W. Burgard, and S. Thrun. Learning motion patterns of persons for mobile service robots. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2002.

[2] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17(7):760–772, July 1998.

[3] B. Kluge. Recursive probabilistic velocity obstacles for reflective navigation. In *Proc. of 1st Int. Workshop on Advances in Service Robots (ASER)*, Bardolino, Italy, Mar. 2003.