The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Proactive Avoidance of Moving Obstacles for a Service Robot utilizing a Behavior-Based Control

Michael Göller, Florian Steinhardt, Thilo Kerscher, J. Marius Zöllner and Rüdiger Dillmann
FZI - Research Center for Information Technology
76131 Karlsruhe, Germany
[goeller, steinhardt, kerscher, zoellner, dillmann]@fzi.de

***Abstract*— A main challenge in the application of service robotics is safe and reliable navigation of robots in human everyday environments. Supermarkets, which are chosen here as an example, pose a challenging scenario because they usually have a cluttered and nested character. The robot has to avoid collisions with static and even with moving obstacles while interacting with nearby humans or a dedicated user respectively. This paper presents a hierarchical approach for the proactive avoidance of moving objects as it is used on the robot shopping trolley InBOT. The behavior-based control (bbc) of InBOT is extended by a reflex and a reactive behavior to ensure adequate reaction times when confronted with a possible collision. On top of the bbc a spatio-temporal planner is situated which is able to predict environmental changes and therefore can generate a safe movement sequence accordingly.**

I. INTRODUCTION

Our goal is to enable robots to operate in highly frequented human environments. Exemplarily we have chosen a supermarket scenario. Here the Interactive Behavior Operated Trolley InBOT (Fig. 1) addresses several everyday problems such as helping the customer to find desired products without extensive search, or relieving customers from the burden of pushing the shopping cart using his own force all the time, especially if the cart is heavily loaded or the customer is elderly or handicapped. Here we are involved in ongoing HRI research (e.g. [10], [2]). The chosen environment poses several challenges: supermarkets are often cluttered and contain several dynamic objects or persons which makes safety highest priority. A behavior-based control is available on InBOT that copes with task planning, navigation and avoidance of static obstacles.

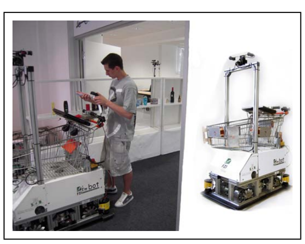

Fig. 1. The Interactive Behavior Operated shopping Trolley (InBOT). It is equipped with a mecanum drive for full holonomic movements and the environment is observed by two laser range finders. Additionally the user can find a wide range of equipment for interaction with the robot. The setup is described in [6]. The left hand side shows a scene from an user study.

This paper focuses on the avoidance of accidental collisions with moving obstacles such as customers with shopping carts who are hurrying down a corridor being distracted by the products in the shelves or talking with each other. The identification and tracking of the moving obstacles is not part of this paper. The corresponding movement model (direction, velocity, variances) is provided by an external component. This task is performed in a three-level approach: first a reflex moves the robot directly away from dynamic obstacles which came too close or started moving, enabling InBOT to regain a safety distance. Second a reactive behavior lets the robot free the predicted path of an approaching object, guaranteeing a suitable reaction time. To solve complex trapping situations these components are topped by a spatio-temporal planner which generates a safe and efficient long-term path.

The remaining paper is organized as follows: Following the introduction the paper continues with a brief analysis of related work and the description of the present control of the robot InBOT (section III). After this section IV deals with the reactive avoidance of the objects while section V presents the spatio-temporal planner. In the final sections the experimental evaluation is described and the results are discussed.

II. RELATED WORK

There are several different popular methods to avoid dynamic objects so far. They can be grouped in two main classes: The plan based approaches: Hu et al. first plans a path around the static obstacles, the dynamic obstacles are considered in an extra step that controls the robot-velocity on the path around the static obstacles [9]. Another very popular approach are Elastic Bands. First a path around the static obstacles is planned and later deformed with the elastic-bands-method to direct the robot around the moving obstacles [12]. Hoeller et al. use a modified probabilistic path planner to avoid predicted trajectories of a human. These predicted positions block the probabilistic planner from adding a new waypoint near this position [8]. Large et al. describe a realtime dynamic obstacle avoidance system learning typical trajectories of moving obstacles and feeding them into an iterative motion planner based on Velocity Obstacles [11]. Bennewitz et al. use learned motion patterns of persons. Hidden Markov Models are derived to estimate future movements. The probabilistic belief is incorporated into the path planning process [1].

And the reactive approaches: An approach using a multisensor based environment prediction is described by Song and Chang [13] where the predicted obstacle positions are used to calculate virtual forces to decelerate the robot. Castro et al. present a system which uses a laser range finder based obstacle detection and tracking and feeds this information in an extended dynamic window system [3].

The advantage of the deliberativity of the planners is obvious but we lack too many information (limited field of view, no exact and reliable map, sometimes no target of robot is available e.g. in *visual servoing mode* or using the power assisted control mode) to be able to rely always on a planner-based approach only. Therefore we combine a planner with reactive approaches which are developed to fit into our already present behavior network.

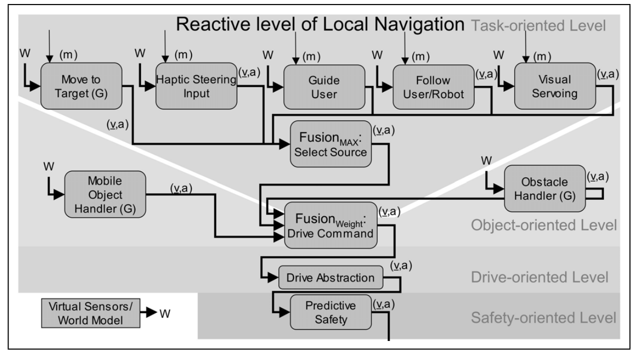

Fig. 2. The four levels of the behavior network which implements the *Reactive Layer*. Behavior groups, or subnetworks, are indicated with (G).

III. THE HIERARCHICAL CONTROL ARCHITECTURE

A three-layered navigation concept was developed to cover the three main fields of duty, see [7] and [6] for details. The *Strategic Layer* is responsible for the deliberative long-term planning (task-planner and topological navigation). The *Tactical Layer* deals with the prediction and deliberative avoidance of dynamic objects as well as with the interaction with nearby humans. And finally at the bottom of the hierarchy the *Reactive Layer*'s goal is to fulfill actual movement tasks implementing a behavior network as control. It will be described in the following how the reactive components for the avoidance of dynamic obstacles are woven into the remaining navigation system.

A. The Behavior Network implementing the Reactive Layer

The main task of the *Reactive Layer* is to provide the repertoire of basic behaviors, which offer the basic skills of the robot. To fulfill the required fast response to environmental changes a very tight coupling of sensors and controllers is needed. To raise robustness all activated behaviors work completely independent from each other.

In the *Object-Oriented Level* of the network (Fig. 2, see [5]) a task/goal-oriented attracting vector is merged with attracting and repelling vectors added by the various behaviors. Due to the dynamic environment no path planning is performed - the robot calculates the most beneficial movement live instead. The behaviors on this level are grouped in two sub-networks: one each for the avoidance of static and of dynamic obstacles.

IV. REACTIVE AVOIDANCE OF DYNAMIC OBSTACLES

This behavior group is located in the *Object-Oriented Level* of the *Reactive Layer* (Sec. III). It consists of two modules that both generate a repelling vector to be added to the vectors of the *Object-Oriented Level* which finally add up to the resulting velocity set-point vector. The first one is a safety reflex that generates a vector that points directly away from a nearby dynamic object to keep or regain a safety distance. The second behavior needs a movement model acquired in the *Tactical Layer* by mid-term observations of the object. Based on the movement model a probabilistic temporal repelling field is calculated. This field is then merged to a resulting repelling vector. Both behaviors are based on a database of detected dynamic obstacles in robot's close vicinity and their characteristics such as position and movement model.

A. The Escape Reflex

This reflex lets the robot retreat directly away when in the range of a moving object. To calculate the repelling velocity vector ($\vec{u}_{Escape}$) for the behavior a repelling vector ($\vec{R}_O$) for each visible dynamic object (O) is calculated. The direction of the repelling vector is directly away from the dynamic object ($\hat{OR}$) and its length depends on the distance between object and robot ($|\vec{d_{OR}}|$) as well as on the velocity of the object ($|\vec{v_O}|$). The area of influence is restricted by the maximum radius (r_{max}) of the repelling field.

$$\vec{u}_{Escape} = \sum^{O}(\vec{R}_O) \tag{1}$$

$$\vec{R}_O = \hat{OR} \cdot \frac{max(r_{max} - |\vec{d_{OR}}|; 0)}{r_{max}} \cdot |\vec{v_O}| \tag{2}$$

B. The Evade Behavior

This behavior clears the way for nearby moving objects. Therefore a probabilistic estimation of the object's path is calculated based on the estimated movement model of the object. Based on the resulting probability field, the repelling movement vector is calculated, which lets the robot move away from the potentially occupied space and therefore makes a collision improbable.

To calculate the repelling velocity vector ($\vec{u}_{Evade}$) for the behavior a repelling vector ($\vec{R}_O$) for each visible dynamic object is calculated. The direction ($\hat{\vec{v}}_{O\perp}$) of this object-dependent repelling vector is orthogonal to the movement direction of the object ($\hat{\vec{v}}_O$). The length of the vector depends on the actual velocity of the object ($|\vec{v_O}|$) and a force factor ($f(\vec{d_O^M}, \vec{V_O})$). The force factor is a function of the distance vector between robot and the model-based movement estimation of the object ($\vec{d_O^M}$, see Fig. 3) as well as the variance ($\vec{V_O}$) of the movement model. It consists of two components, one based on the distance along the path of the object (the x-component) the other one based on the distance between the robot and the path of the object (the y-component). Both components are restricted by corresponding distance restriction functions $(D_X^{max}(), D_Y^{max}(V_Y, d_X))$. These mainly depend on the parameter for the maximum distance of influence (d_X^{max}, d_Y^{max}). $D_Y^{max}(V_Y, d_X)$ also takes the variance and the distance along the path (d_X) into account. Therefore the area of effect enlarges with the distance to the object and with the variance (Fig.4).

$$\vec{u}_{Evade} = \sum^{O}(\vec{R}_O) \tag{3}$$

$$\vec{R}_O = \hat{\vec{v}}_{O\perp} \cdot f(\vec{d_O^M}, \vec{V_O}) \cdot |\vec{v_O}| \tag{4}$$

$$f(\vec{d_O^M}, \vec{V_O}) = \left(1 - \frac{min(d_X; D_X^{max}())}{D_X^{max}()}\right) \cdot \left(1 - \frac{min(d_Y; D_Y^{max}(V_Y, d_X))}{D_Y^{max}(V_Y, d_X)}\right) \tag{5}$$

$$\vec{d_O^M} = \begin{pmatrix} d_X \\ d_Y \end{pmatrix}, \vec{V_O} = \begin{pmatrix} V_X \\ V_Y \end{pmatrix}, D_X^{max}() = d_X^{max} \tag{6}$$

$$D_Y^{max}(V_Y, d_X) = d_Y^{max} \cdot (V_Y + 1) \cdot ((V_Y \cdot d_X) + 1) \tag{7}$$

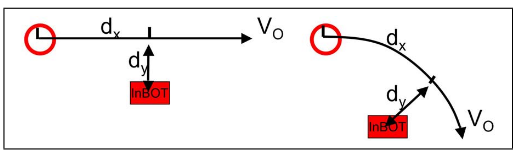

Fig. 3. A dynamic object (red circle) approaches InBOT (red rectangle). Two examples for the movement model dependent distance vector ($\vec{d_O^M} = (d_X, d_Y)$) are illustrated. Left: the estimated movement is linear. Right: the estimated movement of the object is a curve. In both cases two components are calculated: the first one along the movement direction, the second one orthogonal.

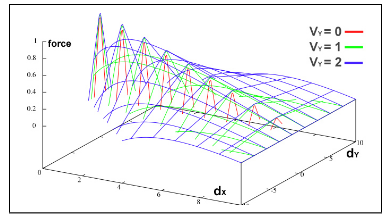

Fig. 4. The complete repelling force field of one dynamic object for three different y-variances of the movement model. The width of the area of influence in y-direction depends on the y-variance (V_Y) of the movement model. The length depends on the velocity of the object (the object moves along the X-axis).

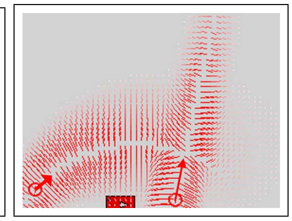

Fig. 5. Two dynamic objects (circle) approach the robot (box). The various red lines represent the repelling vector fields. The object moving in a curve is harder to predict resulting in a raised variance and following up in a grown vector field.

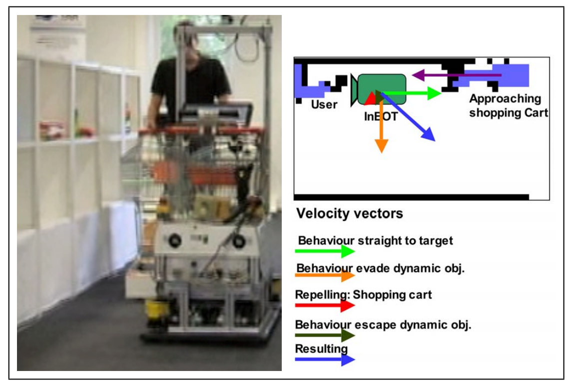

Fig. 6. In a corridor InBOT is confronted with a shopping cart approaching from the front. On the right hand side the corresponding occupancy map with the velocity set point vectors from the individual behaviors is sketched. Fig. 7 shows InBOT's reaction.

C. Experimental Results

Fig. 5 shows the repelling vector field generated by the two behaviors *Escape Dynamic Objects* (circular part) and *Evade Dynamic Objects* (oval part). Here one object does not move straight ahead but in a curve. Additionally the prediction of one object is bad, so the variance of the movement model rises. Fig. 6 shows a scene with an shopping cart approaching from the front while Fig. 7 presents the resulting movement of the robot. It is repelled from the predicted path of the objects and moves around the critical zones smoothly. This concept works well as long as only few dynamic objects are in the close vicinity of the robot and the robot has sufficient space to avoid them. But scenes can occur where the robot cannot escape the objects. For example if two objects move in parallel and trap the robot between them or too many objects are heading for the robot from different sides. To avoid these situations the predictive handler for dynamic obstacles will be introduced in the next section. The combined concept is based on the division of duty between a predictive part for the long- and midterm control and a part for the reactive micro-management.

V. PREDICTIVE PLANNER FOR PROACTIVE AVOIDANCE OF DYNAMIC OBSTACLES

The two presented reactive behaviors manage the short term collision avoidance of dynamic objects. But they do not perform a deliberative analysis of the scene. The planner to be introduced here generates a spatio-temporal sub-plan in the local environment to reach a given target while taking the predicted movements of all visible dynamic objects into account. The goal is to basically avoid critical situations which could not be solved by reactive behaviors only (e.g. trapping situations between objects). The plan consists of sub-targets with time constrains, it is not a trajectory for the robot. The execution of the plan is performed by the basic behaviors (*StraightToTraget*, *AvoidObstacles*, *EvadeDynamicObject*, *EscapeFromDynamicObject* and the *Safety Reflex*

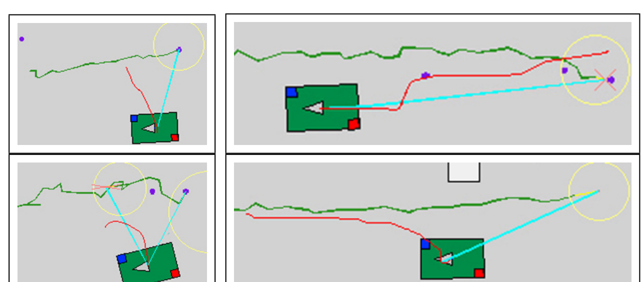

Fig. 7. Resulting movements: Left: The standing robot frees the path of an approaching object. In the bottom scene the object was very hard to track resulting in a jittery tracking line and a not optimal movement of the robot. Right: here the robot is moving itself while confronted with an approaching object (top: head to head, bottom: object is overtaking the robot).

- see [7], [5]) which guarantee safety even if the planning process should fail or the objects suddenly change direction or velocity drastically.

A. Baseline: Data and Preprocessing

The environmental robot-centered occupancy map (10x10m) is extended with a time-dimension generating a 3D occupancy map (Z-axis for time). Static objects are assumed as time invariant and therefore span the complete Z-axis. The position of dynamic objects is altered according to the movement estimation of the object (see Fig. 8). A new grid has to be built for each plan: The time-distance t between two X-Y layers is set so that it matches the time interval needed by the robot to move one cell along the x axis at default speed ($t = \frac{cellsize}{v_{def}}$ - with v_{def} usually 0.5 to 1 m/s). The height (time) of the 3D occupancy grid limits the length of the path that can be found. The resolution of the basic occupancy grid is reduced and moving obstacles as well as special objects (the user, other robots) are deleted from this occupancy grid. The static obstacles are enlarged (the size of the robot can now be assumed as point like) and added to the 3D grid. Additionally for each time-step an enlarged obstacle is entered at the predicted positions of the moving obstacles. The resulting predictive 3D obstacle grid is shown on the right side of Fig. 8. We decided to generate a new plan each time instead of altering the old one because we observe drastic changes in the environmental grid due to the robot-centered grid, the limited and changing field of view, the dependency of the robot's actual velocity and finally the fuzzy prediction of the moving objects' behavior which suddenly can be subject to major changes.

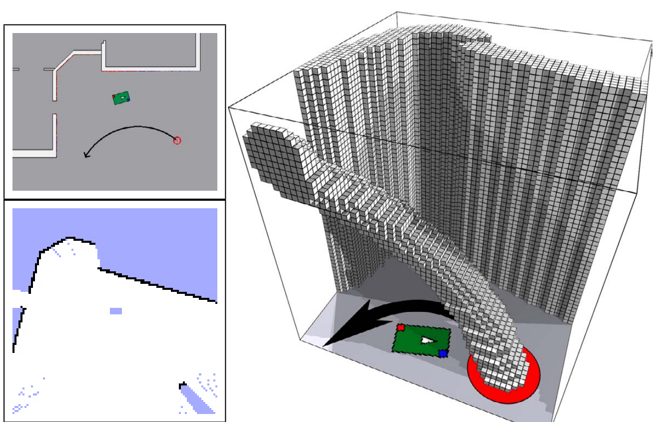

Fig. 8. Left: The scene: a dynamic object (red circle with arrow) moves in the close vicinity of the robot. Below the corresponding occupancy grid is shown. Right: predictive 3D occupancy grid with mobile obstacle in the front ($50 \times 50 \times 50$ cells, cellsize: $20cm \times 20cm \times 0.4sec$).

B. Spatio-Temporal Calculation of Safe Path using an A* Algorithm

A starting cell and the goal coordinates have to be set to calculate a path around the obstacles in the predictive 3D occupancy grid. The robot is positioned at the middle of the lowest time-layer of the grid - the start cell. All cells that share the X-Y-coordinates of the goalpoint are possible goalcells for the A* search. If the goalpoint is outside of the occupancy grid a substitute goalpoint on the grid's border is choosen.

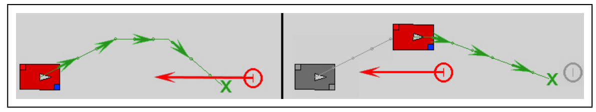

Fig. 9. The robot's target is the green 'X' at the right side of the scene. But a dynamic object approaches from this direction (red circle with arrow). Therefore a plan of sub-targets is generated that leads the robot around it.

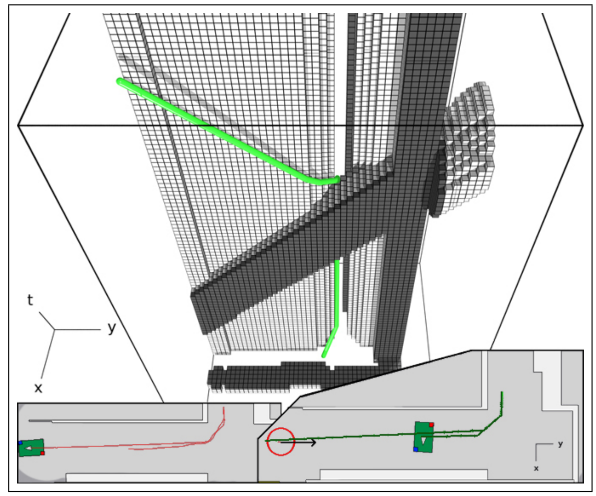

Fig. 10. The robot's target is at the left end of the corridor. But a dynamic object approaches from this direction (red circle with arrow). Therefore a plan of sub-targets is generated that leads the robot back around the corner, lets it wait there until the object passed and finally leads to the target again. Bottom/right: plan in 2D view, Middle: plan in a cut open 3D view, Bottom/left: actual path taken by the robot)

The A* algorithm is forced to take a time step for each X/Y movement, therefore a cell has 9 neighbors which are located in the succeeding time layer $t' = t + 1$. Occupied cells in the 3D obstacle grid are obviously off-limits.

The heuristic funtion used for the A* search is the straight-line distance to the goal in an X-Y layer ($\sqrt{\Delta x^2 + \Delta y^2}$) multiplied with the factor $\sqrt{2}$ to estimate the needed time to get to the goal (this is equivalent to the straight-line distance in the 3D occupancy grid). This factor is a correct estimation if the A*-search can find a free way to the goal without having to wait or having to take a detour around an obstacle.

The search will be aborted if the current cell in the A* search is located in the top X-Y layer $t = t_{max}$.

If the A* search does not find a path to the goal there is no free path to the goal in the given time constraint and the search has to be restarted on the next new occupancy grid data. In the meantime the robot relies on the reactive behaviors to avoid collisions.

C. Optimization of Path

The found path consists of a list of sub-goal points. If the subgoals are too close together the flexibility of the behavior-based control is hindered. If there are no obstacles the path still consists of many subgoals in a straight line even if only the goalpoint itself would be sufficient to provide a collision-free path.

Hence the path is simplified to contain only the needed points to drive around the obstacles by repeatedly removing sub-goals and then testing for collisions with occupied cells. If the resulting path is collision-free the removed subgoals are removed permanently.

D. Activation of Behavior-Based Control

Once a path is found and simplified its subgoals are supplied to the behavior-based control as goalpoints. The behavior-based control moves the robot to these points successively.

E. Experimental Results

The described algorithm was tested thoroughly. An easy example is displayed in Fig.9. A dynamic object approaches from the direction the robot intends to move in. After detecting the object a plan is generated to move around the object. A more challenging scene is depicted in Fig. 10. This time there is not sufficient space to move around the object. Therefore the plan leads the robot back around a corner to let the object pass before continuing with moving towards the target.

VI. EXPERIMENTAL EVALUATION

Here we will presents some of the experiments performed to evaluate the concept. When not stated otherwise all experiments where performed with the real robot in the real environment.

A. Comparison of components

Fig.11 shows the effect of the different levels for obstacle avoidance. Using only the behaviors for the avoidance of static obstacles a collision occurred if the robot was not significantly faster than the object. The more methods were activated the earlier the robot started the dodging movement and the more efficient and safe became the path.

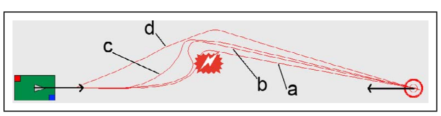

Fig. 11. The individual obstacle avoidance levels in comparison (higher levels always include all lower levels): a:AvoidStaticObstacles, b:EscapeReflex, c:EvadeBehavior, d:ProactivePlanner. On path (a) most of the time a collision occurs (depending on the proportion of the robot's velocity compared to the object's velocity).

B. Stress test

A stress test was performed in a partially simulated setup: Three objects are moving back and forth on straight paths always crossing the robot's target point. The robot's tracker for dynamic objects was manipulated to simulate the objects to be able to test the avoidance system independently from the quality of the tracker. The (real) robot has to continuously dodge the objects. Fig. 12 illustrates the setup and the path the robot has actually taken during a two-minute test run.

Table I shows the results. Using only the behaviors for the avoidance of static obstacles 34 collisions occurred. Using the reactive methods the number shrank to one. When

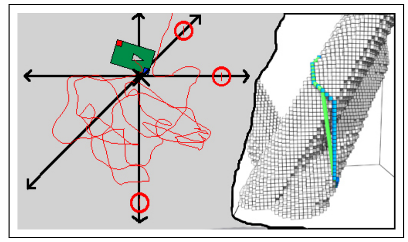

Fig. 12. Stress test: The robot is ordered to move to the X in the middle while three objects move back and forth on paths crossing just this point. In a two-minute test run there were 50 possible collisions. In the bottom right corner is an exemplary extract from the planner with the calculated and optimized path.

TABLE I

RESULTS OF STRESS TEST: 50 PASSES DURING TWO MINUTES

Method used	Collisions	Average distance to closest object
Static avoidance	34	146 mm
Escape and Evade	1	665 mm
Above + Planner	1	1041 mm

additionally using the planner the distance to the closest object is significantly higher resulting in higher safety. The single collision in both cases was unavoidable because the (simulated) objects turn around without warning - beyond the laws of inertia - without having to (de-) accelerate.

TABLE II

OVERALL SUCCESS RATE: 100 TRIAL RUNS

Tracking information	Collisions	Risky	Safe
Using tracker	13	26	61
Using exact position and velocity	3	10	87

C. Overall success rate

Table II shows the overall success rate over various experiments. InBOT (the real robot) was approached 100 times by (real) shopping carts while performing various tasks. The ordinary shopping carts were steered by people who moved to a given goal chosen to provoke a collision while ignoring the robot, but some even rushed directly onto the robot. First our object tracker was used, then the exact positions were provided. The amount of safe runs shows that the avoidance concept performs well but tracker has to be further improved. The majority of collisions and risky runs (slight touching not hard enough to activate the robot's bumper) was caused when the cart was identified too late and the robot did not have sufficient time to accelerate away. There were 3 collisions and 10 risky situations in the run with the exact cart positions in which the drivers of the cart did not let InBOT a chance to avoid them properly. The reason is the acceleration and velocity limitation (1m/s) set for the robot which was significantly lower than the speed of some of the cart drivers. Additionally some forced the robot against a wall were InBOT fell to a dead stop shortly before hitting it and then was hit by the cart. Keeping in mind that the intention is to avoid accidental collisions the robot fulfilled our expectations. Finally we want to note here that in all cases of collisions InBOT was hit by the approaching object due to insufficient dodging - InBOT itself did not run into any objects or into the path of an approaching object. Therefore safety from the robot's point of view was always granted.

D. System in application

The last test we want to present includes some scenes from a shopping experiment where the robot is hindered in executing given tasks by dynamic obstacles (e.g. other shoppers with shopping carts). Fig. 13 shows a single shopping run in a small laboratory supermarket setting. The user is asked to pick up 4 products while other shoppers are moving around him with their carts. First the robot guides the user to the first product and simultaneously avoids the moving obstacle A, later in the *Following Mode* the robot keeps a safe distance from obstacles B and C. The encounter with cart C is shown in Fig. 14 as well. In all cases the robot did not move into other objects and did not collide with it's user (as described in section III and Fig. 2 the presented behaviors are merged with the behaviors for handling static obstacles and the safety module counterchecks all movement commands in the end).

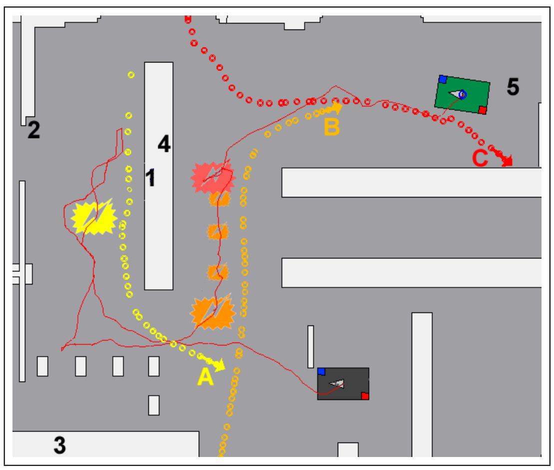

Fig. 13. Shopping run: the user starts the robot in the lower right (grey rectangle) and is guided to the first product (1) while avoiding the moving shopping cart A: strong bend to the left at the yellow flash symbol. After driving to the next product (3) the robot continues to product (4) and avoids the crossing cart B by slowing down and letting it pass. The robot continues by aligning itself behind the cart (orange flashes). While following to the checkout counter (5) the robot is cut by C. Again the robot lets the cart pass (red flash). (Robot path red, critical areas: flash symbols, the user - who is continuously impairing the robot being another close-by obstacle - is omitted for better perceptibility).

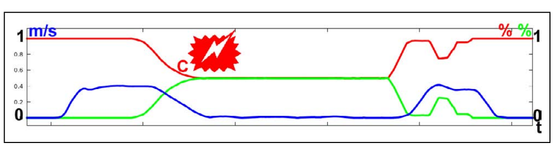

Fig. 14. Control Sharing: while in *Following Mode* the user exercises full control of the robot (control share of user: red), but when approached by the cart C the robot takes over a share of control to avoid the collision (control share of robot: green) resulting in a direction change and stopping of the robot (velocity of robot: blue).

VII. CONCLUSIONS

Finally the system for avoidance of dynamic obstacles is ready and operational. In several tests it has proven a suitable performance. First it consists of two reactive components to guarantee suitable reaction times: in a suddenly apparent pre-collision situation the robot can react with its maximum possible acceleration within 20ms (fastest cycle time in the control architecture). And second of a spatio-temporal planer to provide proactive and deliberative commands in complex situations. When the planning should fail or take too long as well as in situations when no planning is possible (such as in the *Power Assisted Control Mode* or the *Visual Servoing Mode* - here no target location is available) the gap is filled by the reactive behaviors. The system was designed to avoid accidental collisions and performs well under this assumption. If someone should really try to hit the robot he will succeed due to the velocity and acceleration limitations applied on the robot.

VIII. ACKNOWLEDGMENTS

This research has been carried out in the CommRob project (www.commrob.eu) and is partially funded by the EU (contract number IST-045441) under the 6th framework programme.

REFERENCES

[1] M.Bennewitz, W.Burgard, G.Cielniak, S.Thrun: Learning Motion Patterns of People for Compliant Robot Motion, International Journal of Robotics Research (IJRR), 24(1), 2005

[2] C.Bogdan et al. - Towards HRI on the Move with Mixed Initiative. In the proceedings of the Second International Symposium on New Frontiers in Human-Robot Interaction at the Convention for the Study of Artificial Intelligence and Simulation of Behaviour (AISB), Leicester (UK), March 2010.

[3] D.Castro, U.Nunes, A.Ruano: Reactive Local Navigation, Proceedings, IEEE 28th annual Conf. of the Industrial Electronics Society, 2002

[4] A.Elfes: Using occupancy grids for mobile robot perception and navigation, Computer, 22(6), 46-57 (1989).

[5] M.Göller, T.Kerscher, J.M.Zöllner, R.Dillmann: Obstacle-Handling of the holonomic driven Interactive Behavior Operated shopping Trolley InBOT, 2009, Proceedings of the IEEE 7th International Workshop on Robot Motion and Control (RoMoCo09)

[6] M.Göller, M.Devy, T.Germa, F.Lerasle, T.Kerscher, J.M.Zöllner, R.Dillmann: Setup and Control Architecture for an Interactive Shopping Cart, Proceedings of the 14th International Conference on Advanced Robotics (ICAR), 2009

[7] M.Göller, T.Kerscher, J.M.Zöllner, R.Dillmann: Behavior Network Control for a Holonomic Mobile Robot in Realistic Environments, Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR), 2008

[8] F.Hoeller, D.Schulz, M.Moors, F.E.Schneider: Accompanying Persons with a Mobile Robot using Motion Prediction and Probabilistic Roadmaps, Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems

[9] H.Hu, M.Brady, P.Probert: Navigation and Control of a Mobile Robot among Moving Obstacles, Proceedings of the 30th Conference on Decision and Control, 1991

[10] H.Hüttenrauch et al. - Evaluation of Robot Body Movements to Support Communication. In the proceedings of the Second International Symposium on New Frontiers in Human-Robot Interaction at the Convention for the Study of Artificial Intelligence and Simulation of Behaviour (AISB), Leicester (UK), March 2010.

[11] F.Large, D.Vasquez, T.Fraichard, C.Laugier: Avoiding Cars and Pedestrians Using Velocity Obstacles and Motion Prediction, 2004 IEEE Intelligent Vehicles Symposium

[12] S.Quinlan, O.Khatib: Elastic Bands: Connecting Path Planning and Control, Proceedings of the IEEE International Conference on Robotics and Automation, 1993

[13] K.T.Song, C.C.Chang: Reactive Navigation in Dynamic Environment Using a Multisensor Predictor, IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, Vol. 29, No. 6, 1999