2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

A Representation Method Based on the Probability of Collision for Safe Robot Navigation in Domestic Environments

S.A.M. Coenen, J.J.M. Lunenburg, M.J.G. van de Molengraft and M. Steinbuch

Abstract— **This paper introduces a three-dimensional volumetric representation for safe navigation. It is based on the OctoMap representation framework that probabilistically fuses sensor measurements to represent the occupancy probability of volumes. To achieve safe navigation in a domestic environment this representation is extended with a model of the occupancy probability if no sensor measurements are received, and a proactive approach to deal with unpredictably moving obstacles that can arise from behind occlusions by always expecting obstacles to appear on the robot's path. By combining the occupancy probability of volumes with the position uncertainty of the robot, a probability of collision is obtained. It is shown that by relating this probability to a safe velocity limit a robot in a real domestic environment can move close to a certain maximum velocity but decides to attain a slower safe velocity limit when it must, analogous to velocity limits and warning signs in traffic.**

I. Introduction

Robots that navigate in indoor, domestic environments face an environment that encompasses obstacles and uncertainties. Obstacles generally vary in their size and shape and can be static as well as dynamic. Uncertainty typically arises from three sources [1]: (i) sensing uncertainty due to noisy and false sensor measurements, (ii) uncertainty about the environment due to (partially) unknown parts of that environment and (iii) robot position uncertainty due to localization errors and external disturbances acting on the robot. In the presence of these characteristics a robot always needs to guarantee that it is safe, i.e., that it can ensure to come to a timely stop when a collision is imminent. Safe navigation can be achieved by moving a robot at very low velocities, typically below 0.1 m/s. However, by incorporating knowledge on the environment a robot can move with higher velocities without becoming unsafe.

Approaches to safe navigation in a domestic environment generally consist of a motion planner that represents the environment and searches and executes a path through this represented environment. A representation of the environment that is three-dimensional is a prerequisite to deal with the challenges that a typical domestic environment poses [2]. Approximate cell decompositions are a common and popular approach to manage this representation [3], [4]. Searching and executing a path in an environment with obstacles and uncertainties is typically achieved using a planner that finds a path to the goal that is executed by a reactive algorithm [3].

All authors are with the Dept. of Mechanical Engineering, Eindhoven University of Technology, The Netherlands. ✉:s.a.m.coenen@student.tue.nl, {j.j.m.lunenburg, m.j.g.v.d.molengraft, m.steinbuch}@tue.nl.

The research leading to these results is part of the R3-COP project, funded by ARTEMIS and NL Agency (Dutch Ministry of Economic Affairs).

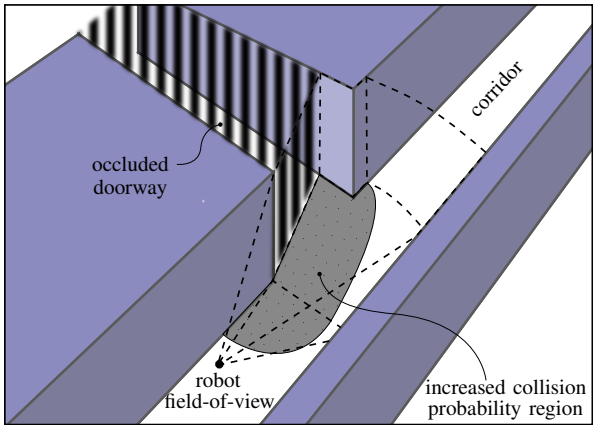


Fig. 1. A corridor with a doorway that is occluded to a three-dimensional sensor. In a domestic environment such occlusions increase the probability of collision for a robot as an obstacle can emerge from such occlusions.

Such a reactive algorithm controls the robot at real-time to avoid imminent collisions by stopping or swerving the robot when an obstacle is known to be on the robot's path.

Space that is unknown, due to occlusions and a limited sensory range, poses a threat to a robot that navigates in a domestic environment as moving obstacles might emerge from behind occlusions onto the robot's path. An exemplary situation occurs when a robot traverses a corridor as depicted in Fig. 1. Near unknown space, such as a passage or a doorway that is occluded to the sensors, an obstacle can suddenly emerge. This requires the robot to lower its velocity such that it can guarantee that it will not collide with a possible incoming obstacle. The robot can increase its velocity when a confined part of the environment, like the part of the corridor past the doorway, is known to be free as there is no uncertainty arising from any unknown space. A reactive algorithm that has no information about this threat can approach at a velocity that does not allow the robot to detect the imminent collision and react to it to avoid collision. Hence, analogous to velocity limits and warning signs in traffic, a representation is necessary that allows the robot to decide when it is able to move at a certain maximum velocity and when it must maintain a slower pace to ensure safety.

II. Related Work and Contribution

To achieve safe behavior some approaches modify the robot's planned path or motion using the obstacle representation. They enlarge obstacles by inflating their representation with a 'safe' distance that encompasses the robot position uncertainty as well as the robot sensing uncertainty [5], [6]. While this may ensure that a robot keeps a greater distance

from obstacles, it may also prohibit a robot to traverse tight spaces. This is undesirable as tight spaces such as a doorway are typically present in a domestic environment. Other approaches limit the robot's velocity based on distance to obstacles on the robot's path [7] or the amount of clearance on both sides of the robot [8]. The former approach will fail when a moving obstacle emerges from occluded regions, e.g., a doorway in a corridor as shown in Fig. 1. The latter approach is conservative in that it will scale down its velocity in a narrow corridor, while in such a confined part of the environment the robot can drive safely at higher velocities, for example past and prior to the doorway in the corridor depicted in Fig. 1.

Uncertainties can explicitly be taken into account to prevent collisions. A range of approaches estimate the probability of collision of the robot along its path [1], [9]–[12]. These approaches plan a path by sampling the environment for feasible robot configurations that reduce the probability of collision during the execution of a path. These approaches result in robot motions with a lower probability of collision by explicitly considering the robot position and sensing uncertainty. However, these approaches do not take into account the uncertainty that arises due to unknown parts of the environment. In [2], a method is introduced that does track the unknown space in three dimensions. To guarantee safe behavior it is ensured that the robot never traverses this unknown space. However, it is assumed that the environment is mostly static, i.e., it can not ensure safe behavior if an obstacle emerges from an occluded part of the environment. Furthermore, once any unknown space is marked as free it will remain free. This makes the representation overconfident in its assumption that space is free as this space can become occupied again in a changing environment. Hence, a time-dependent occupancy probability model is necessary, instead of merely tracking unknown space.

Finally, some approaches explicitly model the uncertainty that arises due to unpredictable moving obstacles, e.g., humans [13]–[15]. Moving obstacles are extracted from subsequent sensor readings and their position and velocity is estimated to obtain a probabilistic model that resembles the risk of collision. A moving obstacle can, however, be occluded up to an imminent collision.

Hence, current approaches that deal with uncertainties to allow safe navigation are not suited for a domestic environment. Not all present uncertainties are considered in an integrated approach and they lack an explicit model to deal with the uncertainty that arises due to occlusions in an environment with unpredictably moving obstacles.

This paper contributes a three-dimensional representation that allows a motion planner to decide when it can move at a certain maximum velocity and when it must maintain a slower pace to ensure safety based on the probability of collision. This is achieved by explicitly representing the multiple uncertainties that are present in a domestic environment and using the probability of collision to determine a safe velocity limit, analogous to velocity limits and warning signs in traffic, to ensure safe navigation.

III. Environment Representation

In this section the proposed environment representation for safe navigation is elaborated. First, this representation, based on the *OctoMap* framework [4], will be introduced and it is explained how it uses probabilistic fusion of sensor measurements to be robust against uncertainty in sensing. Secondly, it is described how uncertainty due to unknown space is correctly represented if no measurements are received and how the probability of obstacles appearing on the robot's path from behind occlusions is represented using a proactive approach [16]. These first two parts results in the representation of the probability of volumes being occupied by an obstacle. Next, the probability of the robot occupying a volume is represented using a model of the robot position uncertainty. Finally, the probability occupancy of obstacles and the probability occupancy of volumes by the robot are combined to represent a probability of collision. This probability is then related to a safe velocity limit that the robot must attain in order to guarantee safety.

A. Robot Sensing Uncertainty Representation

The *OctoMap* framework provides a volumetric octree-based representation of the environment [4]. It models the environment as free, occupied and unknown cubic volumes or so-called voxels. Sensor measurements are integrated probabilistically using occupancy grid mapping [17]. This technique allows a probabilistic fusion of multiple sensor measurements making it robust to noisy and false sensor measurements. Also, it allows the integration of measurements from multiple, different sensors.

The occupancy of a voxel is updated as measurements are received. Each voxel n, with a resolution r, has a probability $P(n)$ of being occupied. The occupancy probability of all volumes is typically initialized to unknown, i.e., the uniform prior $P(n) = 0.5$. Then, $P(n)$ is updated based on a sensor specific model as measurements $z_{1:k}$ up to time step k are received. The update rule for the estimated voxel occupancy, using the log-odds (L) notation [17], is

$$L(n \mid z_{1:k}) = L(n \mid z_{1:k-1}) + L(n \mid z_k), \tag{1}$$

with

$$L(n) = \log \left[\frac{P(n)}{1 - P(n)} \right], \tag{2}$$

where $L(n \mid z_1 : k)$ is the estimated log-odds probability of a voxel given measurements $z_{1:k}$, $L(n \mid z_{k-1})$ is the previously estimated log-odds probability and $L(n \mid z_k)$ denotes the log-odds probability of voxel n being occupied given the measurement z_k. The update of occupancy probability is typically performed in log-odds as using additions is faster than the multiplications that are necessary when (1) is expressed in $P(n)$.

$L(n \mid z_k)$ relies on a sensor model that relates the sensor measurements to the occupancy probability of a voxel:

$$L(n \mid z_k) = \begin{cases} l_{\text{free}}, & \text{if } n \text{ is marked as free} \\ l_{\text{occ}}, & \text{if } n \text{ is marked as occupied} \end{cases} \tag{3}$$

Given equally likely measurements ($l_{\text{free}} = l_{\text{occ}}$), a voxel that is marked as free k times needs to be marked as occupied equally many k times before its occupancy probability is equal again. This makes the representation unable to change as quickly as the environment. As discussed in [4], the representation can be made adaptable to a changing environment by limiting the probability in the update rule in (1) to a lower and upper bound on the log-odds value, respectively l_{min} and l_{max} or p_{min} and p_{max} on the probability. For more details on the update formula and its background the reader is referred to [17] and [4].

B. Environment Uncertainty Representation

The update rule introduced in (1) models the occupancy probability of voxels under the assumption that they receive measurements. However, parts of the environment typically receive no measurements as they are occluded to the robot's sensors. It is important that the update rule in (1) also describes the occupancy probability of voxels if no measurements are received. For example, consider a robot that moves in an environment such that all voxels n in the representation are receiving measurements, either marking voxels as free or occupied. Then, given a static environment, the occupancy of the voxels will approach the threshold, i.e., either $P(n) = p_{\text{min}}$ or $P(n) = p_{\text{max}}$. However, it is incorrect to assume that the occupancy of those voxels that do not receive measurements does not change if the environment is dynamic. In other words, such a representation is over-confident in its assumption that space is either free or occupied. Hence, the occupancy of a voxel is more realistically modeled to become unknown again as no measurements are received for some time.

A time-dependent occupancy probability model is added to the representation to deal with this environment characteristic. Instead of only updating a voxel n if a measurement is received, it is updated at every time step k that the environment is updated. Thereto, the sensor model, introduced in (3), can be extended with the occupancy probability update rule

$$l_{\text{dec}} = (k - k^n_{z,\text{last}})\Delta_{\text{dec}} \quad \text{if } n \text{ is } \textbf{not} \text{ marked}, \tag{4}$$

where $k^n_{z,\text{last}}$ is the time step at which a voxel n received its last measurement update and Δ_{dec} indicates the rate of decay of the probability of a voxel n in l_{dec}/k. The sign of the log-odds value l_{dec} depends on the occupancy of a voxel according to:

$$l_{\text{dec}} = \begin{cases} +l_{\text{dec}}, & \text{if } l < 0, \text{ i.e., } P(n) < 0.5 \\ -l_{\text{dec}}, & \text{if } l > 0, \text{ i.e., } P(n) > 0.5 \end{cases} \tag{5}$$

Hence, if a voxel receives no measurements it gradually turns to unknown again, i.e., $P(n) = 0.5$. This is visualized for a voxel that is at the lower probability threshold in Fig. 2. The update rule in (5) is based on a model that is linear in log-odds similar to the sensor measurement update model in (1). Therefore, the rate of decay Δ_{dec} can intuitively be chosen as a rate at which measurements are discarded again,

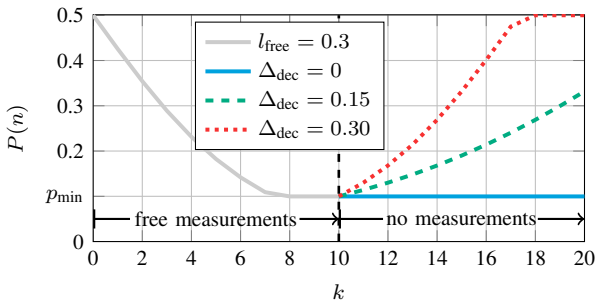


Fig. 2. The occupancy probability of a voxel $P(n)$ decreasing with $l_{\text{free}} = -0.3$ according to (3) as it is marked as free at the first ten time steps and increasing with l_{dec} according to (5) for various Δ_{dec} as it is not marked for ten following time steps.

such that the probability returns to unknown. For example, a measurement that decreases the probability of occupancy is discarded at the next step the environment is updated and no measurement is received by choosing $\Delta_{\text{dec}} = l_{\text{free}}$. Hence, a voxel becomes unknown again after an equal amount of time steps that no measurement is received. The update rule in (5) also allows to let the occupancy probability of an occupied voxel decrease to unknown as no measurements are received. This would however require more knowledge of obstacles to discriminate between them. For example, a voxel that belongs to a static obstacle, e.g., a wall, is more likely to remain occupied than a voxel that belongs to a moving obstacle, e.g., a human. As obstacles are not modeled separately for now all voxels that are occupied are considered to be static. This means that $l_{\text{dec}} = 0$ if $P(n) > 0.5$ in (5).

The representation of occupancy probability is three-dimensional. However, to keep the computational complexity of the representation tractable the three-dimensional grid of voxels is projected down to a two-dimensional representation. Therefore, each column of voxels in the occupancy map is projected down to a grid cell with occupancy probability p according to

$$p_{\mathbf{c}} = \max_i P(n_i), \tag{6}$$

where $\mathbf{c}$ is a grid cell with resolution r. Taking the maximum occupancy probability is a conservative strategy that is necessary for safe navigation as it ensures that the robot never underestimates the possibility that a voxel is occupied at any height in a column.

As mentioned, in a domestic environment an obstacle can move on the robot's path from behind an occlusion, as illustrated in Fig. 1. The occupancy probability model that has been introduced so far must be extended with a velocity model to deal with this. This model is based on a proactive approach that is introduced in [16]. By proactive it is meant that the robot is always expecting that a moving obstacle can appear on the robot's path from an occluded region. This is achieved by inflating the occupancy probability of cells in the two-dimensional projected map that are not marked as an obstacle, as depicted in Fig. 3. The inflation distance d_{obs} depends on the distance that an obstacle can travel within the time the robot needs to come to a stop,

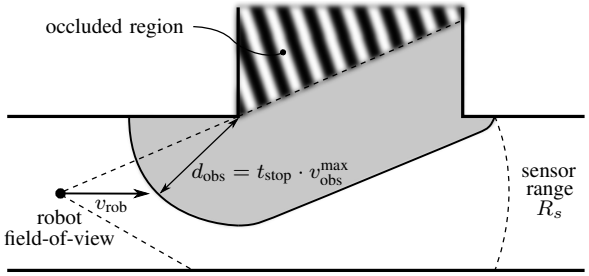


Fig. 3. An obstacle can appear on the robot's path out of a region that is occluded to the robot. To represent this possibility the uncertain region is inflated with the maximum distance the obstacle can travel within the time the robot needs to come to a stop.

$t_{\text{stop}} = v_{\text{rob}}/a_m + t_d$, where v_{rob} is the robot's velocity, $-a_m$ is the maximum deceleration and t_d is the maximum update delay that is present before an incoming obstacle is actually detected. The distance an obstacle can travel with a maximum velocity $v_{\text{obs}}^{\text{max}}$ is now $d_{\text{obs}} = t_{\text{stop}} v_{\text{obs}}^{\text{max}}$, which is conservative as it assumes that an obstacle will maintain its maximum velocity. However, this assumption is necessary as any obstacle trajectory is assumed to be unknown.

The cells of the grid map with an occupancy probability are inflated with the obstacle inflation distance d_{obs}. A grid map cell can be affected by the inflation of multiple cells. Hence, these probabilities must be merged and this is done by taking the maximum probability of all inflated probabilities. Hereby, it is assumed that the inflated probabilities are dependent. In practice this means that all inflated cells are regarded as one potential source of collision risk. By guaranteeing that the robot is safe in this situation it can also be guaranteed that it is so in the case of multiple sources of collision risk.

C. Robot Position Uncertainty Representation

The uncertainty that is present in the robot's position is due to errors in the estimation of the robot's position relative to a map of its environment. This uncertainty can be modeled as a bivariate normal (Gaussian) distribution

$$\mathbf{x} \sim \mathcal{N}_2(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{7}$$

where the mean $\boldsymbol{\mu}$ is the robot position at $\mathbf{x} = (x; y)$ and $\boldsymbol{\Sigma}$ is the two-dimensional covariance matrix. For simplicity, it is hereby assumed that robot has a circular shaped platform. The occupancy probability of a cell $\mathbf{c}$ by the robot can be determined according to the normalized distribution $\mathcal{N}(\mathbf{c}; \boldsymbol{\mu}, \boldsymbol{\Sigma})r^2$, where r^2 is the surface of $\mathbf{c}$. The distribution of the robot's position is considered within a prediction interval to limit computation. Given a probability of $1 - \alpha$ the robot's position is guaranteed to be in a certain region $\mathcal{A}$ centered around the mean $\boldsymbol{\mu}$. The region $\mathcal{A}$ is an ellipsoid with a 'radius' k, given by the relation $(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) = k^2$ that can be deduced from (7). Given α, the ellipsoid shaped region $\mathcal{A}$ follows from $k = \sqrt{\chi_2^2(\alpha)}$, where $\chi_2^2 \alpha$ is the upper (100α) percentile from the two-dimensional chi-squared distribution. For example, for $\boldsymbol{\Sigma} = [0.1\,0; 0\,0.1]$ and $\alpha = 0.05$, the robot's position is located with a probability of $p = 0.95$ in a circle with a radius $k = \sqrt{0.1 \cdot 5.99}$.

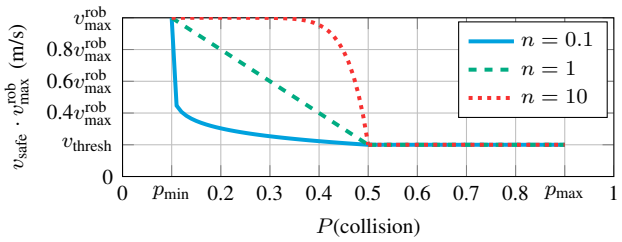


Fig. 4. The safe velocity limit as a n-degree polynomial function of the robot's maximum velocity. A threshold velocity, $v_{\text{thresh}} = 0.2 v_{\text{max}}^{\text{rob}}$, allows the robot to move at a limited velocity in the presence of any risk of collision. The robot can be allowed to move faster by taking more risk.

D. Combined Representation

The environment uncertainty model in Section III-B and the robot position model in Section III-C are combined to obtain a probability of collision. The probability of a cell being occupied by an obstacle and the probability of that same cell being occupied by the robot are independent. Hence, the probability of collision, i.e., a cell is occupied by both the robot and an obstacle, is equal to the product of both probabilities. Now, the probability of collision at a specific cell can be determined according to

$$P(\text{collision}) = \sum_{\mathbf{c} \in \mathcal{A}} P(\mathbf{c} = \text{occ}) \cdot \mathcal{N}(\mathbf{c}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) r^2, \tag{8}$$

This probability of collision must be related to a velocity limit v_{safe} that ensures safe navigation. Thereto, the maximum velocity must be lowered as the probability of collision increases, analogous to velocity limits that arise in traffic. The maximum velocity can be derived given that $d_{\text{rob}} + d_{\text{obs}} < R_s$ must always hold to avoid collision. Here, R_s is the limited sensory range and $d_{\text{rob}} = \frac{1}{2} a_m (v_{\text{rob}}/a_m)^2 + v_{\text{rob}} t_d$ is the maximum distance that the robot travels to come to a stop. Given $d_{\text{obs}} = t_{\text{stop}} v_{\text{obs}}^{\text{max}}$, the maximum robot velocity is

$$v_{\text{rob}}^{\text{max}} = -v_{\text{obs}}^{\text{max}} - a_m t_d + \sqrt{a_m^2 t_d^2 + (v_{\text{obs}}^{\text{max}})^2 + 2 a_m R_s}. \tag{9}$$

One could argue that to be strictly safe the robot must have a zero velocity in the presence of any probability of collision. However, the robot must accept some risk of collision during navigation as it is generally not absolutely certain that space is free. This is an inherent consequence of the update rule in (1) and the clamping threshold p_{min} on the occupancy probability. Hence, a threshold velocity v_{thresh} is used, that allows to robot to drive at a velocity close zero in the presence of any uncertainty. The choice for v_{safe} is a trade-off between moving safely at v_{thresh} or moving faster with more risk of collision. This trade-off is visualized in Fig. 4 for a polynomial function with different degrees. Of course, the exact function to relate the probability of collision to a safe velocity limit is a design choice that will most likely depend on the specific application at hand.

IV. Experimental Results

The representation that is proposed in Section III is verified using the AMIGO robot, a domestic service robot developed by Eindhoven University of Technology. AMIGO

competes in the RoboCup@Home League. This annual competition, where domestic service robots compete in performing household tasks, is part of the international RoboCup project [18]. Experiments have been performed in a simulation scenario based on the RoboCup@Home League 2013 set-up and in a real domestic environment at the Robotics Lab of Eindhoven University of Technology.

A. The AMIGO Robot

AMIGO has a four-wheeled omni-directional base that is capable of navigating through wheelchair-accessible areas. Its torso is equipped with two anthropomorphic arms to perform manipulation tasks. To extend the reach of the arms the torso is connected to the circular base with a lifting mechanism. In this study the arms are not used. A Hokuyo UTM-30LX Laser Scanner, positioned at the front-side of its base, provides a 220° view at 30 cm above the ground. AMIGO uses adaptive Monte Carlo localization (AMCL) [19] to localize itself on a static, a-priori map. A Microsoft Kinect mounted with a pan-tilt unit on top is used to provide three-dimensional pointcloud data. For this experiment both the laser scanner and the Kinect are used for navigation, but any range sensor that produces pointcloud data can be used.

The environment representation, introduced in Section III, is used in the existing motion planning approach of AMIGO. This approach consists of a global planner that searches for a plan to the goal and a local planner that computes a velocity such that this global path is followed. The global planner uses an A* algorithm that searches for a cost-optimal path in terms of distance to obstacles. Re-planning is performed at a fixed frequency of 2 Hz. A re-plan is executed if it is significantly better in terms of the cumulative cost of the path. The local planner computes an omni-directional velocity to reach the next pose (x, y, θ) of the global path. It simulates a trajectory forward along the global path that tries to maximize the velocity while attaining the velocity limit that the representation imposes, i.e., v_{safe} as introduced in Section III-D. High-level decisions are made in an executive layer on top of the planners. These decisions include recovery behavior, e.g., clearing a part of the map if AMIGO thinks it is stuck, and the gaze direction of the Kinect at a fixed distance on the global path in front of AMIGO.

Experiments are performed in simulation and in a real domestic environment to verify that the proposed representation allows a robot to navigate safely in a domestic environment. For both experiments the parameters are set according to Table I. The maximum robot velocity robot is determined according to (9), with $t_d = \frac{2}{3}$ s. This maximum delay is based on the minimum update rate of the representation which is 3 Hz. As it can take up to two measurements before an obstacle is detected the maximum delay is equal to $\frac{2}{3}$ s. The voxel resolution of the representation is $r = 0.05$ m, such that AMIGO can still fit through the narrowest doorway with a width of 80 cm. The covariance terms of the position uncertainty model are based on the covariance matrix provided by the AMCL module. For simplicity, the maximum variance obtained from previous tests with AMIGO is used.

TABLE I
DIFFERENT MODEL PARAMETERS USED IN EXPERIMENTS

sensors		environment		robot	
l_{free}	-1.10 $(p = 0.25)$	$v_{\text{obs}}^{\max}$	1.0 m/s	$v_{\text{rob}}^{\max}$	0.7 m/s
l_{occ}	$+0.85$ $(p = 0.70)$	R_s	3.2 m	v_{thresh}	$0.2 v_{\text{rob}}^{\max}$
$l_{\min}$	-1.40 $(p = 0.20)$	t_d	$\frac{2}{3}$ s	a_m	0.5 m/s^2
$l_{\max}$	$+2.20$ $(p = 0.90)$	r	0.05 m	σ_x^2, σ_y^2	0.1 m
				α	0.05

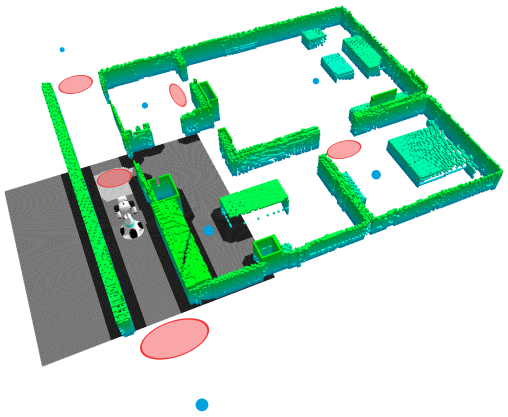


Fig. 5. A model of the 2013 RoboCup@Home League set-up used for the simulation experiments. The blue dots indicate the predefined goal locations that are randomly visited by AMIGO. The areas that are marked red indicate regions where an obstacle can cross with AMIGO.

B. Results of Simulation Experiment

Simulations are performed with AMIGO in a model of the RoboCup@Home League 2013 set-up, as shown in Fig. 5. The representation is tested for the velocity risk profiles $n = \{0.1, 1, 10\}$, probability decay rates $\Delta_{\text{dec}} = \{0.0, 0.15, 0.30\}$ and with and without taking into account moving obstacles, $v_{\text{obs}}^{\max} = \{0.0, 1.0\}$ m/s. The position uncertainty is modeled according to the parameters in Table I for all tests.

In a first test the robot drives through the corridor, as shown in Fig. 5, while an obstacle appears from the occluded doorway on its right, thereby blocking the robot's path. The rate of probability decay is not considered in this test. The velocity profile along the path is presented in Fig. 6. At the beginning of the test the part in front of the robot is unknown and hence it moves at the velocity threshold. When the obstacle velocity is taken into account, i.e., $v_{\text{obs}}^{\max} = 1.0$ m/s, the unknown area is inflated and thus the threshold velocity must be attained longer than for $v_{\text{obs}}^{\max} = 0.0$ m/s. Near the doorway the test with $v_{\text{obs}}^{\max} = 0.0$ m/s disregards the increased probability of collision due to moving obstacles resulting in a collision for a medium ($n = 1$) and high ($n = 10$) risk velocity profile. For $n = 10$ the robot even failed to recognize the obstacle and hit it with maximum velocity. Due to the position uncertainty the robot typically moves at a velocity near its threshold for a low risk velocity profile, thereby giving it enough time to detect the obstacle and to stop in time. The test with $v_{\text{obs}}^{\max} = 1.0$ m/s shows that a velocity near the maximum velocity, i.e., $v_{\text{rob}}^{\max} = 0.7$ m/s, is possible without collision.

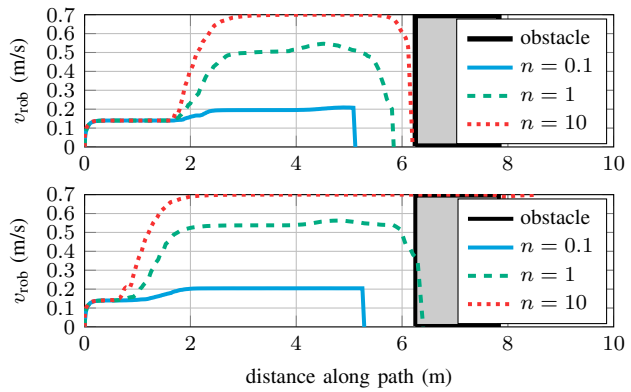


Fig. 6. The velocity of AMIGO with (upper figure) and without (lower figure) taking the obstacle velocity into account, respectively $v_{\text{obs}}^{\text{max}} = 1.0$ m/s and $v_{\text{obs}}^{\text{max}} = 0.0$ m/s, along the same path through the corridor of the simulation environment. Halfway the corridor an obstacle, indicated by the grey block, entered on AMIGO's path from an occluded doorway.

In a second simulation AMIGO drives to random locations, within a set of predefined locations as depicted in Fig. 5, for 30 minutes. Obstacles are modeled to move at random time intervals over predefined paths that will intersect with AMIGO, resulting in areas with an increased risk of collision similar to the test in the corridor. By navigating over a longer timespan the influence of the rate of probability decay (Δ_{dec}) can be determined on the performance. As a measurement of performance the number of collisions and the average velocity during the test is reported in Table II.

The results show that the rate of probability decay has a noticeable influence on the number of collisions. For $\Delta_{\text{dec}} = 0.15$ the number of collisions drops significantly. The difference with tests with $\Delta_{\text{dec}} = 0.3$ is not remarkable. Furthermore, the tests with a low risk velocity profile, i.e., $n = 0.1$, resulted in a robot that appeared too conservative as it was not able to achieve its goal position in most situations. Hence, accepting some risk results in a robot that navigates with a higher mean velocity without necessarily increasing the number of collisions. Furthermore, the robot was sometimes not able to achieve its goal because moving obstacles were not completely cleared and cluttered the environment with occupied voxels. A model for the decrease in occupancy probability for occupied volumes, as discussed in Section III-B, would alleviate this problem.

It can be concluded that the increased probability of collision due to moving obstacles must be taken into account and the probability of occupancy must be time-dependent in order to navigate without collision. Furthermore, the trade-off between velocity and the probability of collision is clear. Accepting some risk of collision during navigation results in a robot that moves at its maximum velocity if it can and at its velocity threshold when it must to ensure safety.

C. Results of Real Environment Experiment

The proposed environment representation is validated by an experiment on the robot in a real domestic environment. This environment, shown in Fig. 7a, is a partial replica of the 2103 RoboCup@Home League set-up on a slightly smaller scale (1:0.9). In the experiment the robot visits a set of

TABLE II
SIMULATION TEST RESULTS

		v_{safe} profile		
	rate of change	low risk ($n = 0.1$)	medium risk ($n = 1$)	high risk ($n = 10$)
$v_{\text{obs}}^{\text{max}} = 1.0$ m/s	$\Delta_{\text{dec}} = 0.3$	$n_{\text{coll}} = 0$ $\bar{v} = 0.11$	$n_{\text{coll}} = 0$ $\bar{v} = 0.24$	$n_{\text{coll}} = 1$ $\bar{v} = 0.30$
	$\Delta_{\text{dec}} = 0.15$	$n_{\text{coll}} = 0$ $\bar{v} = 0.14$	$\mathbf{n_{\text{coll}} = 0}$ $\mathbf{\bar{v} = 0.27}$	$n_{\text{coll}} = 3$ $\bar{v} = 0.32$
	$\Delta_{\text{dec}} = 0.0$	$n_{\text{coll}} = 1$ $\bar{v} = 0.19$	$n_{\text{coll}} = 5$ $\bar{v} = 0.30$	$n_{\text{coll}} = 9$ $\bar{v} = 0.35$
$v_{\text{obs}}^{\text{max}} = 0.0$ m/s	$\Delta_{\text{dec}} = 0.3$	$n_{\text{coll}} = 0$ $\bar{v} = 0.13$	$n_{\text{coll}} = 3$ $\bar{v} = 0.26$	$n_{\text{coll}} = 7$ $\bar{v} = 0.31$
	$\Delta_{\text{dec}} = 0.15$	$n_{\text{coll}} = 0$ $\bar{v} = 0.14$	$n_{\text{coll}} = 3$ $\bar{v} = 0.29$	$n_{\text{coll}} = 7$ $\bar{v} = 0.35$
	$\Delta_{\text{dec}} = 0.0$	$n_{\text{coll}} = 4$ $\bar{v} = 0.20$	$n_{\text{coll}} = 12$ $\bar{v} = 0.33$	$n_{\text{coll}} = 14$ $\bar{v} = 0.43$

waypoints numbered $w_1 - w_7$. The robot is initialized with a three-dimensional map, as depicted in Fig. 7b, and a static map for localization, as depicted in Fig. 7c, that both are generated off-line. A variety of challenging obstacles are present, that are not in the a-priori map, such as a pair of mini wooden shoes between w_3 and w_4, the IV pole and overhanging sidetable near the hospital bed at w_4 and the desk chairs near w_5 and w_6. The parameter set as used in the simulation experiments is used for the experiment and, based on the simulation test results, the variable parameters are set to $v_{\text{obs}}^{\text{max}} = 1.0$ m/s, $\Delta_{\text{dec}} = 0.15$ and $n = 1$.

AMIGO navigated without collision in the domestic environment. In Fig. 7c the robot's footprint on the localization map and its executed path are shown at every second. The velocity along the executed path is displayed in Fig. 7d. The three-dimensional representation correctly represented the obstacles encountered during the run. At parts with an increased probability of collision due to occlusions, e.g., near the doorway in the corridor, the robot lowered its velocity, allowing it to stop in time if an obstacle suddenly appears on its path. In narrow passages the robot lowered its velocity to its threshold to be robust against position uncertainty. The effect of the probability decay rate Δ_{dec} is noticeable near w_4 and w_5. The space near these waypoints is marked as free as the robot approaches. However, as the robot continues to its next waypoint the space becomes unknown again as it is out of sensor range. Hence, a lower velocity limit is present. In open and known space, such as at the beginning of the corridor and near w_5, the robot achieved velocities near its maximum velocity. The sudden decreases in velocity between waypoints are the result of the local planner that requires the robot to rotate when the error in the orientation of the robot with respect to the global path is too large.

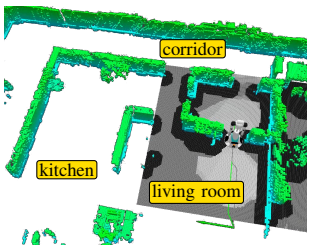
V. CONCLUSIONS AND FUTURE WORK

The proposed volumetric representation allows a robot to safely navigate in a domestic environment. It probabilistically models the occupancy of volumes as sensor measurements are received and, as opposed to typical representations, also if no measurements are received. Furthermore, the probability of moving obstacles appearing on the robot's path from

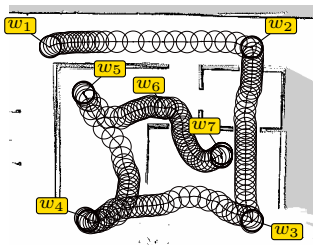
behind occlusions is taken into account. These probabilities are combined with a model of the robot position uncertainty to form a probability of collision. Based on this probability a safe velocity limit is defined. The approach is validated in a real domestic environment where a robot moves with velocities close to a maximum of 0.7 m/s but slows down when it must to ensure safety, e.g., near an occluded doorway.

Several directions can be indicated for improvement of the current approach in future work. Conservative assumptions were necessary to ensure safe navigation. By including more information these assumptions can be relaxed to result in more efficient robot navigation. The representation can be improved by adding explicit obstacle models of, e.g., humans as proposed in [13]–[15], but also of static parts of the environment, e.g., walls. By discriminating in obstacle representations, the probability of their presence can be modeled separately and thus more accurately. Furthermore, an actuated sensor can be controlled to actively reduce uncertainty in the vicinity of the robot. This will decrease the collision probability and thereby increase the safe velocity limit. For example, at start-up the robot is then able to directly face the uncertain space in front of its base, while during navigation it can look further ahead. The position uncertainty can also be modeled more accurately. It is now represented with a normal distribution based on an a-priori determined maximum variance, while directly using the covariance matrix from the AMCL module is more realistic. Finally, deeper insight in the choice of model parameters is desirable. Although those parameters will most likely depend on the specific application at hand, a theoretical and extended simulative analysis in different environment set-ups can reduce heuristic tuning of parameters and make the presented method better generalizable.

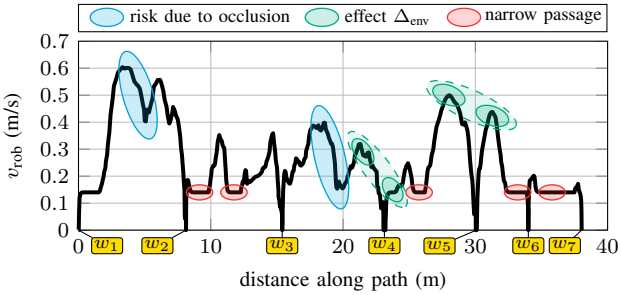
(a) Test set-up at the Robotics Lab of Eindhoven University of Technology.



(b) Robot visualization.



(c) Footprints on the localization map.



(d) The robot velocity along the executed path during the experiment.

Fig. 7. An experiment with AMIGO in a real domestic environment.

References

[1] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 895–913, Jun. 2011.

[2] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: robust navigation in an indoor office environment," in *IEEE Int. Conf. Robot. Autom.*, 2010, pp. 300–307.

[3] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *J. Intell. Robot. Syst.*, vol. 57, no. 1-4, pp. 65–100, Jan. 2010.

[4] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: an efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, pp. 1–18, 2013.

[5] D. Hsu, R. Kindel, J. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. Robot. Res.*, vol. 21, no. 3, pp. 233–255, 2002.

[6] W. Chung, G. Kim, M. Kim, and C. Lee, "Integrated navigation system for indoor service robots in large-scale environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 5, 2004, pp. 5099–5104.

[7] K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann, "About the control of high speed mobile indoor robots," in *Proc. Second European Conf. Mobile Robots*, 2005, pp. 218–223.

[8] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, 1997.

[9] P. Missiuro and N. Roy, "Adapting probabilistic roadmaps to handle uncertain maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 1261–1267.

[10] B. Burns and O. Brock, "Sampling-based motion planning with sensing uncertainty," in *IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3313–3318.

[11] L. J. Guibas, D. Hsu, H. Kurniawati, and E. Rehman, "Bounded uncertainty roadmaps for path planning," in *Algorithmic Foundation of Robotics VIII*. Springer, 2009, pp. 199–215.

[12] S. Patil, J. van den Berg, and R. Alterovitz, "Estimating probability of collision for safe motion planning under gaussian motion and sensing uncertainty," in *IEEE Int. Conf. Robot. Autom.*, 2012, pp. 3238–3244.

[13] R. Philippsen, B. Jensen, and R. Siegwart, "Toward online probabilistic path replanning in dynamic environments," in *IEEE/RSJ Int. Conf. Intell. Robots and Systems*, 2006, pp. 2876–2881.

[14] R. Philippsen, S. Kolski, K. Macek, and B. Jensen, "Mobile robot planning in dynamic environments and on growable costmaps," in *Workshop on Planning with Cost Maps at the IEEE Int. Conf. Robot. Autom.*, 2008.

[15] F. Rohrmüller, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic mapping of dynamic obstacles using markov chains for replanning in dynamic environments," in *IEEE/RSJ Int. Conf. Intell. Robots and Systems*, 2008, pp. 2504–2510.

[16] R. Alami, T. Siméon, and K. Madhava Krishna, "On the influence of sensor capacities and environment dynamics onto collision-free motion plans," in *IEEE/RSJ Int. Conf. Intell. Robots and Systems*, vol. 3, 2002, pp. 2395–2400.

[17] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, vol. 9, no. 2, p. 61, 1988.

[18] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "RoboCup: The robot world cup initiative," in *Proc. 1st Int. Conf. Autonomous Agents*, 1997, pp. 340–347.

[19] D. Fox, "KLD-sampling: Adaptive particle filters and mobile robot localization," *Advances in Neural Information Processing Systems*, vol. 14, no. 1, pp. 26–32, 2001.