In Proceedings for the joint conference of ISR 2010 (41st Internationel Symposium on Robotics) und ROBOTIK 2010 (6th German Conference on Robotics), 7-9 June 2010, Munich, Germany.

A set-based global dynamic window algorithm for robust and safe mobile robot path planning

Sylvia Horn, Klaus Janschek
Institute of Automation, Technische Universität Dresden, Germany

Abstract

In recent papers we proposed a novel approach for mobile robot localization based on set theoretic methods. In this paper we complete this set-based approach by introducing a novel set-based mobile robot path planning algorithm, which can handle the outputs of the set-based localization. In opposite to most known path planning algorithms the approach considers not only kinematic and dynamic constraints of the robot but also noisy measurements from its employed sensors. Realistic simulation experiments validate the effectiveness of the proposed approach.

1 Introduction

Robust and collision free mobile robot navigation in unknown dynamic environments requires forecasting path planning and real-time obstacle avoidance. There are two main approaches to handle this challenge. Global path planning algorithms (e.g. road map techniques or the extended potential field method) generally assume known obstacle positions. But for real world applications unknown or time variant environments must be taken as general design baseline. Thus the computed path can be assumed as collision-free only for small forecast time intervals and must be periodically recalculated. The second task in parallel, dynamic obstacle avoidance methods, like the bubble band technique or the vector field histogramm, employ continuous perception of the surrounding environment. It is well known that motion control derived from obstacle measurement data only is susceptible to local minima. A possible solution, which combines the advantages of global path planning and the dynamic obstacle avoidance, is given with the GLOBAL DYNAMIC WINDOW APPROACH according to [1]. But as most common planning algorithm the standard GLOBAL DYNAMIC WINDOW APPROACH supposes, in opposite to localization tasks where uncertainties are supposed by default, still ideal measurements.

This is indeed resolved in recent path planning approaches [2, 3], but they just consider uncertainties in sensing and neglect motion errors as well as any kinematic or dynamic constraint of the robot. Further the named approaches handle the assumed uncertainties in a baysian framework, which is not able to give a certain prediction. This undermines the wanted guaranteed collision-freeness. In that case a set theoretic approach for handling these uncertainties can be more promising. Up to now set theoretic state estimation techniques have been applied in the field of mobile robotics predominantly for solving localization topics as in [4, 5, 6, 7], which differ basically in various geometric structures describing the sets. A novel set-based estimator has been introduced in [8, 9] combining the benefits of different set theoretic descriptions providing an inclusion certainty region of the robot pose with minimized volume. Starting from these promising results, the set-based estimation approach has been extended to path planning problems in terms of a novel SET-BASED GLOBAL DYNAMIC WINDOW APPROACH. This solves the task of calculating a robust and collision-free path in a partially unknown dynamic environment under consideration of possible unknown but bounded model and measurement disturbances. The outline of the paper is as follows. In section 2 we provide the basics of the set-based state estimation whereas section 3 recalls the fundamental ideas of the dynamic window approach. Further the proposed set-based planning algorithm is introduced in section 4. Section 5 gives implemation details and reports the results of the accomplished simulation experiments. Final conclusions are drawn in section 6.

2 Set theoretic state estimation

Common estimation techniques as the EXTENDED KALMAN FILTER or the PARTICLE FILTER deal with measurement uncertainties based on stochastic descriptions. It is well known, that this always requires a precise knowledge of existing uncertainty probability distributions. In cases, where the underlying uncertainty distributions are not available, a more general approach for handling these quantities as set-based uncertainties is possible. The set theoretic methods use geometrical properties of the uncertainties assuming that the distributions are unknown from

their internal structure, but known to be bounded (UBB - unknown but bounded). In this case it is more appropriate to talk about "inclusion certainty" of the uncertain system parameters. In many applications this approach is more realistic than a stochastic description, e.g. in case of geometric tolerances, quantization problems, bounded noise sources or not well known sensor systems. For set theoretic estimation the true state is - in opposite to the probabilistic "uncertainty" - guaranteed to be inside a specified set, i.e. for the estimated state a certain inclusion set can be specified. To describe these regions, compact convex geometrical structures such as hyperquaders [10], parallelotopes [11, 12] or ellipsoids [13, 14, 15] are employed. As the path planning algorithm presented in this paper is only dealing with hyperquaders and parallelotopes, ellipsoids will not be considered in the further explanations.
A n-dimensional hyperquader, or interval vector, $\mathcal{Z}^{\square}$ is a vector with n interval components

$$\mathcal{Z}^{\square} = \left([z_1^-, z_1^+], \ldots, [z_n^-, z_n^+] \right)^{\mathrm{T}} = \left(\mathbf{z}^-, \mathbf{z}^+ \right),$$

with the edge lengths

$$\mathrm{len}\left(\mathcal{Z}^{\square} \right) = \left(\mathbf{z}^+ - \mathbf{z}^- \right).$$

The unit hyperquader is defined with

$$1^{\square} = \left([-1, 1], \ldots, [-1, 1] \right)^{\mathrm{T}}.$$

Further a n-dimensional parallelotope $\mathcal{Z}^{\diamond}$ is represented as

$$\mathcal{Z}^{\diamond}\left(\check{\mathbf{z}}, \mathbf{T} \right) = \left\{ \mathbf{z} \in \mathbb{R}^n : \mathbf{z} = \check{\mathbf{z}} + \mathbf{T}\boldsymbol{\alpha}, \left\| \boldsymbol{\alpha} \right\|_{\infty} \leq 1 \right\},$$

where $\check{\mathbf{z}}$ denotes the center of the parallelotope and $\mathbf{T}$ the definition matrix containing the direction and length of the parallelotope edges. $\left\| \boldsymbol{\alpha} \right\|_{\infty}$ stands for the ∞-norm.
For state estimation we consider a n-dimensional nonlinear system

$$\dot{\mathbf{z}} = f\left(\mathbf{z}, \mathbf{u} \right).$$

Here $\mathbf{z}$ refers to the state vector and $\mathbf{u}$ marks the unknown, but measurable system input. Further it is considered that the input measurements are corrupted by an additive UBB uncertainty $\mathcal{E}_u^{\square} = [\boldsymbol{\epsilon}_u^-, \boldsymbol{\epsilon}_u^+]$. The initial conditions of the system are also uncertain and given by $\mathcal{Z}_0^{\diamond}\left(\check{\mathbf{z}}_0, \mathbf{T}_0 \right)$.
The recursive set theoretic state estimation needs a linear state space prediction model, which can be derived by linearization and discretization as

$$\mathbf{z}_{k+1} = \left(\mathbf{A}_k + \mathbf{L}_k \right) \mathbf{z}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{l}_k \qquad (1)$$

with the time-variant state space and input matrices $\mathbf{A}_k$ and $\mathbf{B}_k$. The matrices $\mathbf{L}_k$ and $\mathbf{l}_k$ represent uncertainties, which belong to the accomplished linearization process via Taylor expansion and subsequent discretization.
Assuming for now that the linearization error can be neglected, the estimated parallelotope set $\mathcal{Z}_{k+1}^{\diamond}\left(\check{\mathbf{z}}_{k+1}, \mathbf{T}_{k+1} \right)$ according to (1) and $\mathcal{Z}_0^{\diamond}\left(\check{\mathbf{z}}_0, \mathbf{T}_0 \right)$ can be computed with

$$\begin{aligned} \mathbf{T}_{k+1} &= \tilde{\mathbf{T}}_{k+1}\, \mathrm{diag}\left(\frac{1}{2} \mathrm{len}\left(\mathcal{Z}_{k+1}^{\square} \right) \right) \\ \check{\mathbf{z}}_{k+1} &= \mathbf{A}_k \check{\mathbf{z}}_k + \mathbf{B}_k \mathbf{u}_{\mathbf{k}} \end{aligned} \qquad (2)$$

where $\tilde{\mathbf{T}}_{k+1} = \mathbf{A}_k \mathbf{T}_k$ and $\mathcal{Z}_{k+1}^{\square} = 1^{\square} + \tilde{\mathbf{T}}_{k+1}^{-1} \mathbf{B}_k \mathcal{E}_u^{\square}$.
By using the given formula it is assumed, that $\mathbf{T}_{k+1}$ stays regular and well-conditioned for the whole time. If this assumption does not hold, e.g. because of a disadvantageous system matrix $\mathbf{A}_k$, another approach calculating $\mathbf{T}_{k+1}$ can be employed (see [11]).
Further the paper refers to some basic set manipulation techniques, which are defined as follows. The Set-Difference of two sets $\mathcal{P}$ and $\mathcal{Q}$ is given with

$$\mathcal{P} \backslash \mathcal{Q} := \left\{ \mathbf{z} \in \mathbb{R}^n | \mathbf{z} \in \mathcal{P}, \mathbf{z} \notin \mathcal{Q} \right\},$$

the Minkowski-Addition of $\mathcal{P}$ and $\mathcal{Q}$ with

$$\mathcal{P} \oplus \mathcal{Q} := \left\{ \mathbf{z} + \mathbf{q} \in \mathbb{R}^n | \mathbf{z} \in \mathcal{P}, \mathbf{q} \in \mathcal{Q} \right\}$$

and last but not least the Pontryagin-Difference of $\mathcal{P}$ and $\mathcal{Q}$ with

$$\mathcal{P} \ominus \mathcal{Q} := \left\{ \mathbf{z} \in \mathbb{R}^n | \mathbf{z} + \mathbf{q} \in \mathcal{P},\ \forall\, \mathbf{q} \in \mathcal{Q} \right\}.$$

3 The dynamic window approach

The DYNAMIC WINDOW APPROACH (DWA) is an obstacle avoidance technique, which takes into account kinematic and dynamic constraints of a mobile robot. To determine the next motion command the local dynamic window approach [16] directly examines a well chosen velocity space. The so-called dynamic window comprises all possible tuples of the velocity v and the angular velocity ω, which are achievable by the robot - in this special case a synchro drive robot [16] - at the next timestamp under consideration of the current velocity and the acceleration capabilities of the robot. The dynamic window can be further reduced by selecting those tuples that allow the robot to come to standstill before hitting an obstacle, given the current range measurements. For computation of the admissible velocities the approach assumes that the robot moves only in circular arcs, which are represented by such tuples. **Figure 1** shows a typical dynamic window for a synchro drive robot assuming that v and ω are independent. The new motion command can now be determined from the whole set of admissible velocities by applying an objective function. The objective function favors velocities, which lead to fast forward motion, large distances to obstacles and a good alignment to the goal heading. But since the dynamic window approach only considers goal heading alignment and employs no further planning techniques the algorithm is still susceptible to local minima. Here the GLOBAL DYNAMIC WINDOW APPROACH (GDWA) given in [1] implemented on an omni-drive platform adds global thinking by employing the NF1 [17] algorithm to the objective function. The sensed information about the free space is mapped into an occupancy grid and the accomplished NF1 computes the shortest way to the goal.

Instead of considering only the alignment to the goal, the alignment to the shortest path will be used for next motion command determination. This allows some advantages of global path planning without prior information about the robot environment. But depending on the obstacle arrangement the robot can still end up in a local minimum. Both techniques - the DWA and the GDWA - also have in common, that they don't consider uncertainties in the sensory data. Hence, depending on the quality of the sensory data, collisions with obstacles cannot be excluded completely.

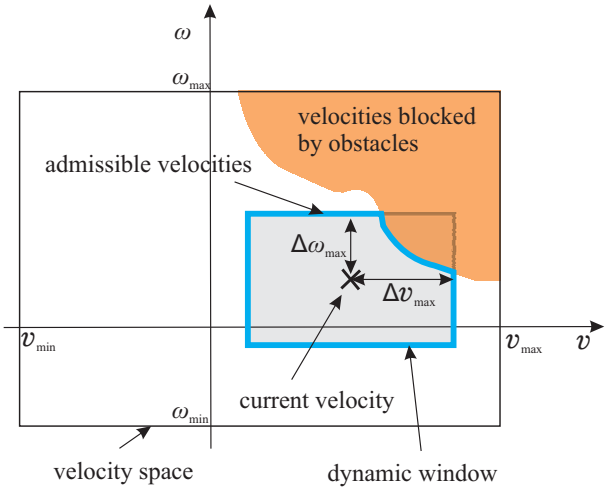

Figure 1: The dynamic window of a synchro-drive robot [16]: The big rectangle shows all achievable speeds v and ω. The small rectangle represents the dynamic window. Blocked are all velocities, which would lead to a collision with obstacles. Admissible are all velocities in the dynamic window, which are not blocked

4 The set-based global dynamic window approach

The aim of the SET-BASED GLOBAL DYNAMIC WINDOW APPROACH (SGDWA) is the guaranteed collision-free determination of motion commands considering noisy measurements, slip and a robot with an extended surface. In principle the SGDWA procedure shown in **Figure 2** follows the already introduced GDWA. The possible robot motions will be predicted and this knowledge will be used together with sensory data for determining the next motion command. But the SGDWA holds some extra beneficial features. Firstly the SGDWA predicts, in opposite to the DWA, the robot motion exactly by using the differential equations describing the system dynamics and kinematics. Secondly the applied parallelotope estimation technique introduced in section 2 calculates an inclusion set which covers all possible robot states assuming the modeled errors, like noisy odometry and drives or even wheel slip. This inclusion set of the true state provides the

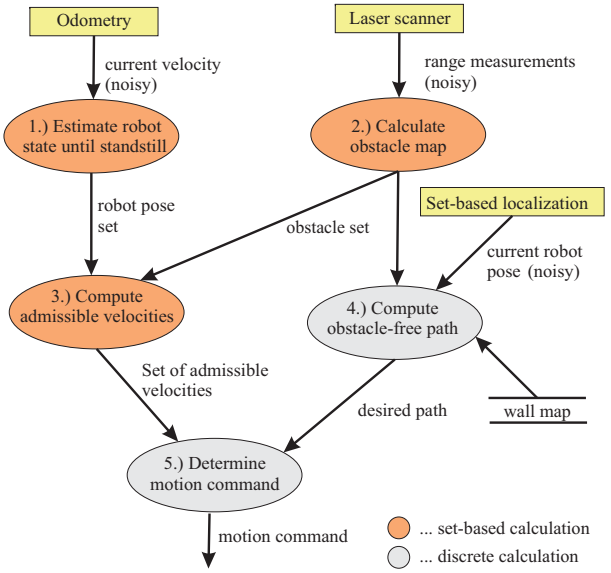

Figure 2: The principle of the proposed set-based global dynamic window approach

ability of this approach to ensure collision-freeness. Last but not least the noisy range data is fused with a coarse wall map to avoid the pinning of the robot in local minima. To fulfill the named conditions of the set theoretic framework the appearing uncertainties are assumed to be unknown but bounded. Further the discussed SGDWA refers to the implementation on a differential-drive robot which is in opposite to an omni- or synchro-drive robot non-holonomic. Perception is assumed to be provided by a laser ranger, an odometry unit and a gyroscope. Current position estimates of the robot are provided by a set-based localization system as described in [8, 9]. The accomplished steps are described more precisely within the next subsections.

4.1 Estimation of the robot state until standstill

The kinematic behavior of a differential drive robot is described by the ordinary differential equations

$$\dot{\mathbf{z}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{v}^{\mathrm{R}} \\ \dot{v}^{\mathrm{L}} \end{pmatrix} = \begin{pmatrix} 0.5\left(v^{\mathrm{R}} + v^{\mathrm{L}}\right)\cos\phi \\ 0.5\left(v^{\mathrm{R}} + v^{\mathrm{L}}\right)\sin\phi \\ 1/D\left(v^{\mathrm{R}} - v^{\mathrm{L}}\right) \\ a^{\mathrm{R}} \\ a^{\mathrm{L}} \end{pmatrix}, \qquad (3)$$

where $(x, y, \phi)^{\mathrm{T}}$ denotes the position and the orientation of the robot, $v^{\mathrm{R/L}}$ and $a^{\mathrm{R/L}}$ the velocity and accordingly the acceleration of the right or left wheel. D corresponds to the distance between the two actuated wheels. It should be noted, that by using a differential-drive robot in opposite to a synchro-drive or omni-drive the dynamic window of v and ω is described by a rhombus whereas the usage of v^{R} and v^{L} precedes in a paraxial dynamic window.

As mentioned in section 2 the parallelotope state estimation technique requires a linear system. Linearization and a subsequent discretization lead to the linear time-discrete state estimation model at timestamp k

$$\begin{aligned}\mathbf{z}_{k+1} &= \left(x_{k+1}, y_{k+1}, \phi_{k+1}, v^{\mathrm{R}}_{k+1}, v^{\mathrm{L}}_{k+1}\right)^{\mathrm{T}} \\ &= \left(\mathbf{A}_k + \mathbf{L_k}\right)\mathbf{z}_k + \mathbf{B}_k \underbrace{\begin{pmatrix} a^{\mathrm{R}}_k \\ a^{\mathrm{L}}_k \end{pmatrix}}_{\mathbf{u}_k} + \mathbf{l}_k, \quad (4)\end{aligned}$$

where $\mathbf{l}_k$ arises directly from the Taylor expansion as $\mathbf{L}_k$ models the linearization error and has to be computed numerically during every estimation step[1].

The accomplished motion prediction splits into two motion phases - an acceleration phase and a following deceleration phase. The acceleration phase involves only one time step and computes all possible states the robot can achieve considering its initial condition and its acceleration facilities as well as the modeled acceleration uncertainties $\mathcal{E}^{\square}_u$. Assuming a robot centric coordinate system the initial conditions at every k can be expressed by $\mathcal{Z}^{\diamond}_{k_0}\left(\check{\mathbf{z}}_{k_0}, \mathbf{T}_{k_0}\right)$ with $\check{\mathbf{z}}_{k_0} = \left(0, 0, 0, v^{\mathrm{R}}_k, v^{\mathrm{L}}_k\right)^{\mathrm{T}}$ and $\mathbf{T}_{k_0} = \mathrm{diag}\left(0, 0, 0, \epsilon_v, \epsilon_v\right)$. ϵ_v marks the assumed odometry error. The dynamic window $\mathcal{V}_k$ now results from the Pontryagin-Difference

$$\begin{aligned}\mathcal{V}_k = \mathrm{proj}\left(\mathcal{Z}^{\diamond}_{k_1}, \langle v^{\mathrm{R}}, v^{\mathrm{L}}\rangle\right) \\ \ominus \mathrm{proj}\left(\mathcal{Z}^{\diamond}_{k_0}, \langle v^{\mathrm{R}}, v^{\mathrm{L}}\rangle\right) \ominus \mathcal{E}^{\square}_u,\end{aligned}$$

where $\mathrm{proj}\left(*, \langle q, p\rangle\right)$ denotes the projection of $*$ in the q-p-space. The following deceleration phase estimation computes the state sets $\mathcal{Z}^{\diamond}_{k_2} \ldots \mathcal{Z}^{\diamond}_{k_m}$ assuming a deceleration until standstill with $a^{\mathrm{R/L}}_{\max}$ and $\mathcal{E}^{\square}_u$. For sufficiently small cycle times the union of $\mathcal{Z}^{\diamond}_{k_1}, \ldots, \mathcal{Z}^{\diamond}_{k_m}$ represents all possible states the robot can achieve until standstill given all speeds represented in the dynamic window.

4.2 Calculation of the obstacle set

Starting from a range measurement $\mathbf{r}_{k,n}$ at scan angle $\boldsymbol{\alpha}_{k,n}$ with the assumed range error ϵ^r and the scan angle step ϵ^{α} it can be easily verified, that the occurring set corresponds to a sector of a corona $\mathcal{C}_{k,n}$. Notice that $\mathcal{C}_{k,n}$ is a non-convex set, which cannot be handled easily within the set theoretic framework. So a minimum volume trapezoid $\mathcal{T}_{k,n}$ containing $\mathcal{C}_{k,n}$ can be calculated analytically. For considering also the robot dimensions given with $\mathcal{R}^{\square}$ the resulting trapezoid will be expanded via the Minkowski-Addition $\mathcal{T}_{k,n} \oplus \mathcal{R}^{\square}$ and circumscribed by an minimum volume parallelotope $\mathcal{O}^{\diamond}_{k,n}$.

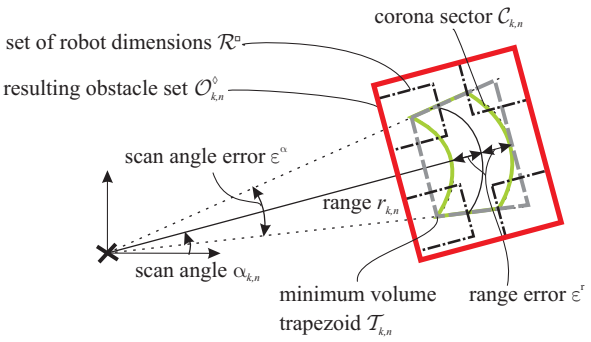

Figure 3: Exemplary calculation of the obstacle set $\mathcal{O}^{\diamond}_{k,n}$ resulting from range $r_{k,n}$ at scan angle $\alpha_{k,n}$ considering the robot dimension set $\mathcal{R}^{\square}$

The described computation of the obstacle set is illustrated for one range value in **Figure 3**. To receive the complete obstacle set $\mathcal{O}^{\diamond}_k$ this procedure has to be repeated for every range value.

4.3 Computation of the admissible velocities

For computation of the admissible velocities the intersection of the obstacle set $\mathcal{O}^{\diamond}_k$ and the state sets $\mathcal{Z}^{\diamond}_{k_1} \ldots \mathcal{Z}^{\diamond}_{k_m}$ has to be calculated, which leads to the problem of a five-dimensional state space and a two-dimensional obstacle set. Therefore the obstacle set can be easily expanded to the missing dimensions using the minimum volume hyperquader circumscribing $\mathcal{Z}^{\diamond}_{k_1} \ldots \mathcal{Z}^{\diamond}_{k_m}$ without having any effect on the inclusion certainty of the true state. So the admissible robot states which do not lead to collision are located within the sets (see **Figure 4a**)

$$\tilde{\mathcal{Z}}_{k_i} = \mathcal{Z}^{\diamond}_{k_i} \backslash \mathcal{O}^{\diamond}_k, \; i = 1, \ldots, m.$$

Further for determining the admissible velocities all $\tilde{\mathcal{Z}}_{k_i}$, $i = 2, \ldots, m$ are transformed in $\mathcal{Z}_{k_1}$-space via

$$\tilde{\mathcal{Z}}^1_{k_i} = \left\{\tilde{\mathbf{A}}^{-1}_{k_i}\mathbf{z} - \tilde{\mathbf{b}}_{k_i} \in \mathbb{R}^n | \mathbf{z} \in \tilde{\mathcal{Z}}_{k_i}\right\}$$

with

$$\tilde{\mathbf{A}}_{k_i} = \mathbf{A}_{k_{i-1}} \cdot \mathbf{A}_{k_{i-2}} \cdot \ldots \cdot \mathbf{A}_{k_1}$$

and

$$\begin{aligned}\tilde{\mathbf{b}}_{k_i} = \mathbf{B}_{k_{i-1}}\mathbf{u}_{k_{i-1}} \\ + \sum_{j=1}^{i-2}\left(\mathbf{A}_{k_{i-1}} \cdot \mathbf{A}_{k_{i-2}} \cdot \ldots \cdot \mathbf{A}_{k_{j+1}}\right)\mathbf{B}_{k_j}\mathbf{u}_{k_j}.\end{aligned}$$

Now the set of admissible velocities (see **Figure 4b**) results from

$$\begin{aligned}\mathcal{V}^a_k = \bigcap_{i=2}^{m} \mathrm{proj}\left(\tilde{\mathcal{Z}}^1_{k_i}, \langle v^{\mathrm{R}}, v^{\mathrm{L}}\rangle\right) \\ \cap \mathrm{proj}\left(\tilde{\mathcal{Z}}_{k_1}\langle v^{\mathrm{R}}, v^{\mathrm{L}}\rangle\right) \cap \mathcal{V}_k.\end{aligned}$$

[1]Note: The robot cycle means the closed loop of sensing, planning and executing commands at one timestamp k. The estimation step, denoted with k_*, just names the forecasting propagation of the robot states and will be performed several times during one robot cycle.

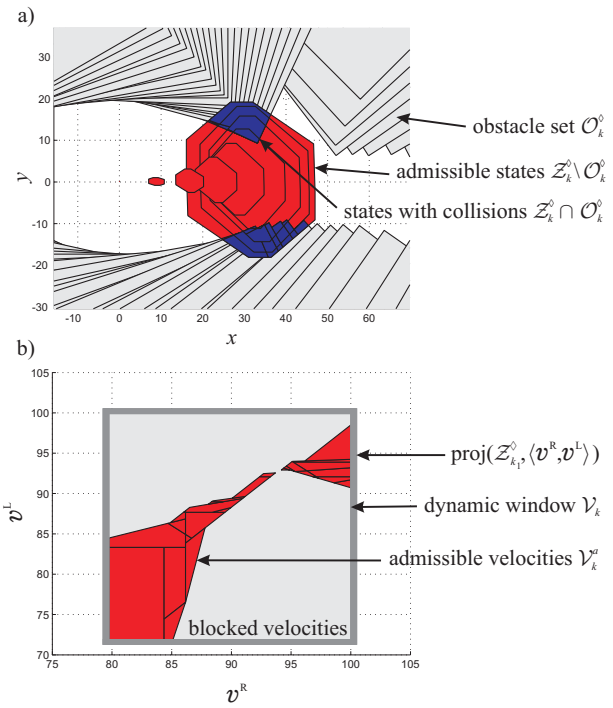

Figure 4: Calculation of the admissible velocities: a) projection in the x-y-plane and b) projection in the v^{R}-v^{L}-plane.

4.4 Obstacle-free path planning

Similar to the GDWA the NF1 path planning technique is used for adding global thinking to the approach. But for adopting the SGDWA also to complex structured office environments some prior knowledge of the environment is given in form of a coarse occupancy map containing information about walls. For joining the wall information with the range data, the occupancy map is transformed into the robot centric coordinate system by using the position and orientation of the robot provided from the assumed set-based localization system and the gyroscope. The range data is converted to pixel coordinates in the resulting occupancy map.

At this point the raw noisy values are used without considering the error bounds since the calculated path is only applied for determining a control command from the already collision-free velocities $\mathcal{V}_k^a$. As commonly known, path calculation can fail, if the uncertainties in position and orientation are too large. In this case collision-freeness is still guaranteed but the successful achievement of the given goal cannot be guaranteed.

4.5 Determination of the velocity command

As mentioned above collision-freeness is already assured by applying $\mathcal{V}_k^a$ as selection for the optimal motion command $\mathbf{c}_k = (c_k^{\mathrm{R}}, c_k^{\mathrm{L}})^{\mathrm{T}}$. So the further calculation is not executed via set theoretic techniques but by decomposition of $\mathcal{V}_k^a$ in discrete values $(\mathbf{v}_k^R, \mathbf{v}_k^L)$. As in [1, 16] the next motion command is then chosen by applying the objective function

$$G\left(v^R, v^L\right) = \alpha \cdot \mathrm{vel}\left(v^R, v^L\right) + \beta \cdot \mathrm{nf1}\left(v^R, v^L\right)$$

to all admissible velocity tuples $(\mathbf{v}_k^{\mathrm{R}}, \mathbf{v}_k^{\mathrm{L}})$. The objective function comprises of the linear combination of the two functions $\mathrm{vel}\left(v^{\mathrm{R}}, v^{\mathrm{L}}\right)$, which favors fast motion, and $\mathrm{nf1}\left(v^{\mathrm{R}}, v^{\mathrm{L}}\right)$ for goal heading. The values of those functions are normalized to the interval $[0, 1]$.

The function $\mathrm{vel}\left(v^{\mathrm{R}}, v^{\mathrm{L}}\right)$ is defined as follows:

$$\mathrm{vel}\left(v^{\mathrm{R}}, v^{\mathrm{L}}\right) = \gamma \cdot \frac{0.5\left(v^{\mathrm{R}} + v^{\mathrm{L}}\right)}{v_{\max}} + \delta \cdot \frac{1/D\left|v^{\mathrm{R}} - v^{\mathrm{L}}\right|}{\omega_{\max}},$$

where $v_{\max} = \max\left(0.5\left(\mathbf{v}_k^{\mathrm{R}} + \mathbf{v}_k^{\mathrm{L}}\right)\right)$ denotes the maximum translational velocity and respectively $\omega_{\max} = \max\left(1/D\left|\mathbf{v}_k^{\mathrm{R}} - \mathbf{v}_k^{\mathrm{L}}\right|\right)$ the maximum rotational velocity. The parameters γ and δ can be adjusted to modify the robots motion behaviour, where $\gamma \gg \delta$ should be chosen. This definition of $\mathrm{vel}\left(v^R, v^L\right)$ differs from the cited approaches in remunerating also rotational motions. This can avoid the premature standstill of the robot without reaching the goal.

The term $\mathrm{nf1}\left(v^R, v^L\right) = 1 - |\theta| / \pi$ is taken from [1] and ensures, that the robot moves towards the goal, where θ marks the angle between the planned path and the possible robot orientation after the acceleration phase. It is advisable to calculate the path alignment in a certain distance to the robot to reach a smooth motion.

5 Simulation results

Realistic simulation experiments have been accomplished via MATLAB in a 2D robot environment developed at our institute. The simulation is based on a differential drive robot with the dynamic constraints $a_{\max} = 1.6\mathrm{m/s}^2$ and $0 \leq v^{\mathrm{R/L}} \leq 1\mathrm{m/s}$. The motion calculation is based on the differential equation set given in (3) and is executed with a one-step ordinary differential equation solver. The simulation experiments are considering several possible error sources like:

- errors of the range measurements (noise only)
- errors of the initial sensors like odometry (noise) or gyroscope (drift and noise)
- errors in the actuated acceleration (noise) and slip
- errors in the estimated pose of a localization module based on set theoretic techniques.

The values are chosen adequate to existing hardware. The random seed depends on the system clock and changes every run. Good-natured dynamic obstacles[2] are also simulated. The trajectory of the dynamic obstacles is unknown to the robot. The start configuration is kept equal over all simulation runs but the further motion of the robot and

[2] In this case good-natured means stopping within spitting distance to the robot.

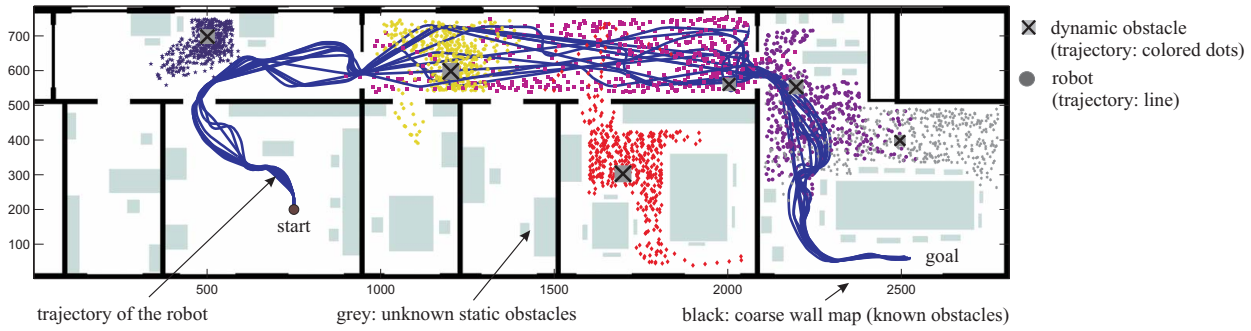

Figure 5: Trajectories of the robot (blue line) and the six dynamic obstacles (colored dots) over 12 runs. The coarse map used for the NF1 is shown in black. The grey obstacles are unknown to the robot and sensed only via laser ranging.

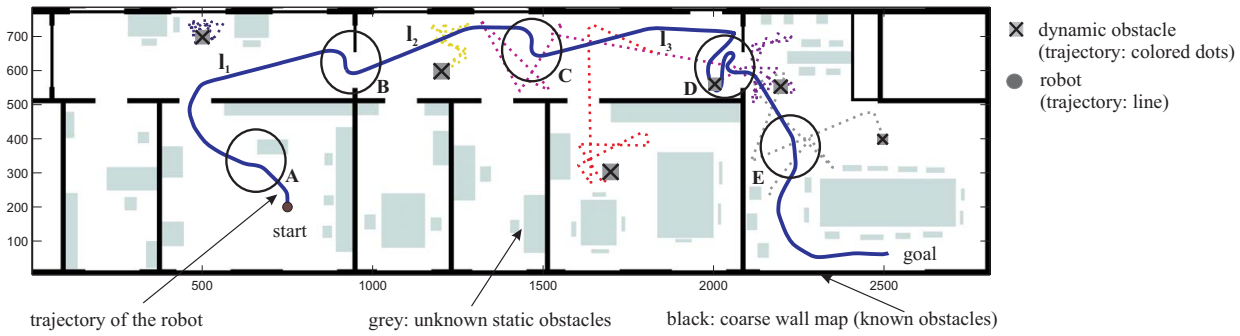

Figure 6: Trajectory of the robot (blue line) and of the six dynamic obstacles (dotted lines) in run 10. The coarse map used for the NF1 is shown in black. The grey obstacles are unknown to the robot and sensed only via laser ranging.

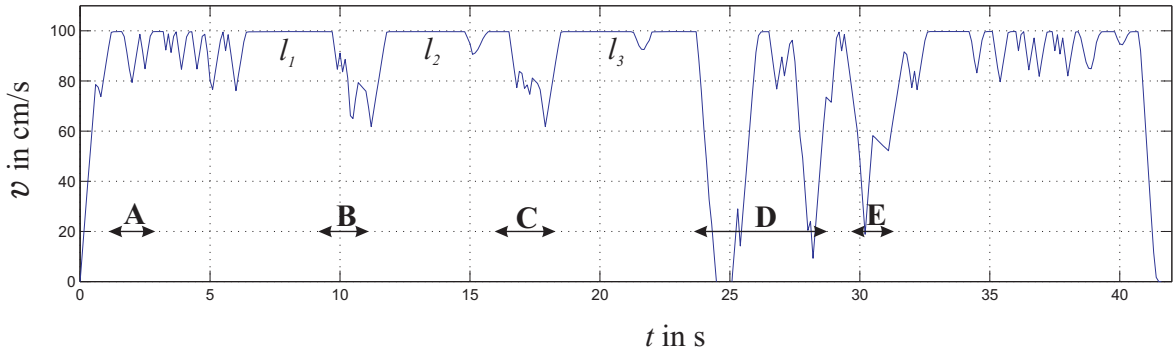

Figure 7: The matching velocity progress in run 10. The named tags match the tags in Figure 6.

the dynamic obstacles depend on the random seed. Anymore, for implementing the set theoretic calculations the free MATLAB Toolboxes MPT [18] and INTLAB [19] are employed.

Figure 5 shows the trajectories of the robot and the dynamic obstacles of twelve simulation runs. It is noticeable that despite the populated area the robot proceeds in all runs successfully at high speeds without any collision. For detailed discussion **Figure 6** presents the results of one simulation run. The tags **A**, **B**, **C**, **D** and **E** indicate different obstacle avoidance maneuvers of the robot. Whereas **A** and **B** result from the presence of a static obstacle (**A** with an unknown and **B** with a known static obstacle), **C** and **E** mark the maneuvers belonging to encountered dynamic obstacles. **D** is a special case, where the robot comes across a dynamic obstacle in a door area. The strong maneuvers indicate the difficulty of avoidance under perpetuation of the goal heading. The long straight sections of the robot trajectory tagged with $\mathbf{l_1}$, $\mathbf{l_2}$ and $\mathbf{l_3}$ result from the employed NF1 method and are desired. The associated velocity progress in **Figure 7** confirms the good navigation performance of the robot at high speeds by using the introduced SGDWA.

6 Conclusions

This paper presented an enhanced DWA approach for robust robot navigation based on set theoretic methods assuming disturbed measurement data. It is shown that the introduced SET-BASED GLOBAL DYNAMIC WINDOW APPROACH is particularly well suited for partially unknown or changing environments where guaranteed collision-freeness is essential. The simulation experiments prove satisfying performance with respect to robustness, reliability and fast forward motion and thus demonstrate a consistent extension of the recently introduced set-based estimation filtering approach to path planning.

References

[1] Brock, O.; Khatib, O.: *High-speed navigation using the global dynamic window approach*, Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Vol. 1, 1999, pp. 341 - 346

[2] Ge, S.S.; Lai, X.; Mamun, A.A: *Boundary following and globally convergent path planning using instant goals*, IEEE Transactions on systems, man and cybernetics, Vol. 35-2, 2005, pp. 240-254

[3] Censi, A.; Calisi, D.; De Luca, A.; Oriolo, G.: *On robot motion planning with uncertainty*, Proceedings of the 2008 IEEE International Conference on Robotics and Automation, 2008, pp. 1798 - 1805

[4] Marco, M.D.; Garulli, A.; Giannitrapani, A.; Vicino, A.: *A set theoretic approach to dynamic robot localization and mapping*, Autonomous robots, Vol.16, 2004, pp.23-47

[5] Gning, A.; Bonnifait, P.: *Dynamic vehicle localization using constraints propagation techniques on intervals: a comparison with kalman filtering*, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, pp. 4144 - 4149

[6] Garulli, A.; Vincino, A.: *Set membership localization of mobile robots via angle measurement*, IEEE Transactions on robotics and automation, Vol. 17-4, 2001, pp. 450-463

[7] Hanebeck, U.D.; Schmidt, G.: *Set theoretic localization of fast mobile robots using an angle measurement technique*, Proceedings of the 1996 IEEE International Conference on Robotics and Automation, 1996, pp. 1387 - 1393

[8] Horn S.; Janschek K.: *A multi-model set membership estimator for mobile robot localization*, Proceedings of the Joint Conference on Robotics ISR/Robotik, VDI Berichte 1956-0090, 2006

[9] Horn S.; Janschek K.: *A multi-model set membership estimator for mobil robot localization - experimental results*, Proceedings of the 7th Conference on Mobile Robots and Competitions, 2007

[10] Alefeld, G.; Herzberger J.: *Introduction to Interval Computations*, Academic Press, New York, 1983

[11] Lohner, R.: *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*, PhD Thesis, Universität Karlsruhe, 1988

[12] Chisci, L.; Garulli, A.; Zappa, G.: *Recursive state bounding by parallelotopes*, Automatica, Vol.32-7, 1996, pp. 1049-1055

[13] Schweppe, F.C.: *Recursive state estimation: Unknown but bounded errors and system inputs*, IEEE Transactions on Automatic Control, Vol. AC-13/1, 1968, pp. 22-28

[14] Ros, L.; Sabater T.; Thomas, F.: *An ellipsoidal calculus based on propagation and fusion*, IEEE Transactions on systems, man and cybernetics, Vol. 32-4, 2002, pp. 430-442

[15] Calafiore, G.: *Reliable localization using set-valued nonlinear filters*, IEEE Transactions on systems, man and cybernetics, Vol.35-2, 2005, pp. 189-197

[16] Fox, D.; Burgard W.; Thrun S.: *The dynamic window approach to collision avoidance*, IEEE Robotics Automation Magazine, Vol.4, 1997, pp. 23-33

[17] Barraquand, J.; Latombe, J.C.: *Robot motion planning: a distributed representation approach*, The International Journal of Robotics Research, Vol.10, 1991, pp. 628-649

[18] Kvasnica, M.; Grieder P.; Baotić M.: *Multi-Parametric Toolbox (MPT)*, http://control.ee.ethz.ch/ mpt/, 2004

[19] Rump, S.M.: *INTLAB - INTerval LABoratory* in *Developments in Reliable Computing*, Kluwer Academic Publishers, http://www.ti3.tu-harburg.de/rump, 1999, pp. 77-104