This work has been accepted to the IEEE 2020 International Conference on Intelligent Transportation Systems.
The published version may be found at https://doi.org/10.1109/ITSC45102.2020.9294388.

Grid-Based Stochastic Model Predictive Control for Trajectory Planning in Uncertain Environments

Tim Brüdigam[1], Fulvio Di Luzio[2], Lucia Pallottino[2], Dirk Wollherr[1], and Marion Leibold[1]

***Abstract*— Stochastic Model Predictive Control has proved to be an efficient method to plan trajectories in uncertain environments, e.g., for autonomous vehicles. Chance constraints ensure that the probability of collision is bounded by a predefined risk parameter. However, considering chance constraints in an optimization problem can be challenging and computationally demanding. In this paper, we present a grid-based Stochastic Model Predictive Control approach. This approach allows to determine a simple deterministic reformulation of the chance constraints and reduces the computational effort, while considering the stochastic nature of the environment. Within the proposed method, we first divide the environment into a grid and, for each predicted step, assign each cell a probability value, which represents the probability that this cell will be occupied by surrounding vehicles. Then, the probabilistic grid is transformed into a binary grid of admissible and inadmissible cells by applying a threshold, representing a risk parameter. Only cells with an occupancy probability lower than the threshold are admissible for the controlled vehicle. Given the admissible cells, a convex hull is generated, which can then be used for trajectory planning. Simulations of an autonomous driving highway scenario show the benefits of the proposed grid-based Stochastic Model Predictive Control method.**

I. Introduction

Model Predictive Control (MPC) has shown to be an efficient method to control autonomous systems, especially autonomous vehicles [1]–[3]. MPC is an optimization-based control method which iteratively solves an optimization problem, possibly with constraints, on a finite horizon. The benefit of iteratively solving an optimization problem with hard constraints enables planning trajectories and maneuvers in dynamic environments without collisions. A major challenge for autonomous driving is to handle uncertainties, especially future behavior of other traffic participants. Exact maneuver predictions are impossible and modeling the exact execution of maneuvers is not perfectly accurate. Therefore, uncertainties must be included in the MPC problem formulation.

While Robust Model Predictive Control (RMPC) yields robust and safe solutions, these solutions are often impractical in dense traffic or for longer prediction horizons, as accounting for the worst-case uncertainty realizations results in overly conservative trajectory planning. Conservatism is reduced by applying Stochastic Model Predictive Control (SMPC) [4] or Scenario Model Predictive Control (SCMPC) [5] with chance constraints. In contrast to robust hard constraints, a chance constraint is only required to be satisfied according to a predefined risk parameter, allowing less conservatively planned trajectories, as worst-case uncertainty realizations are not considered. Combining RMPC and SMPC for mobile robots is considered in [6].

SMPC [7], [8] and SCMPC [9], [10] have previously been applied in autonomous driving. A combination of SMPC and SCMPC to handle uncertainty of surrounding vehicles is suggested in [11] where SCMPC accounts for maneuver uncertainty and SMPC copes with maneuver execution uncertainty of other vehicles. However, solving the optimization problem with chance constraints often requires Gaussian probability distributions or considers only the most likely future motion of other vehicles to simplify the predictions. Additionally, considering an increased number of dynamic objects with chance constraints can be computationally challenging.

An Occupancy Grid (OG) [12] is a mapping grid of the environment where each grid cell is assigned a probability that a certain area is occupied. In [13] and [14] first approaches of OGs for autonomous vehicles have been proposed. In OGs concepts like objects, pedestrians, and vehicles do not exist and data fusion from multiple sensors is efficient. As summarized in [15], OGs have been developed in many ways with focus on autonomous driving for both highway and urban traffic scenarios, e.g., in [16], [17]. However, research has been mainly carried out on how to treat data in order to determine a correspondence between sensors and grid cells, and how occupancy probability is assigned and updated with the accumulation of new data.

In this paper we present a grid-based SMPC method for trajectory planning in uncertain environments. While it is possible to apply the proposed method to various autonomous systems, here we will focus on autonomous vehicles. The proposed method provides a simple strategy to handle arbitrary uncertainty of future vehicle motion. The computational effort of the SMPC optimization problem is manageable and the approach scales well with an increased number of surrounding vehicles or other obstacles. We consider a grid for the environment, i.e., the road. For every predicted step, each cell then gets assigned a probability value, representing its occupancy probability by an obstacle. All cells with an occupancy probability value larger than a predefined SMPC risk parameter are inadmissible, where the risk parameter works as a threshold. Given the admissible

*The authors gratefully acknowledge the financial support by the BMW Group within the CAR@TUM project.

[1] T. Brüdigam, D. Wollherr, and M. Leibold are with the Chair of Automatic Control Engineering at the Technical University of Munich, Germany. {tim.bruedigam; dw; marion.leibold}@tum.de

[2] F. Di Luzio and L. Pallottino are with the Department of Information Engineering at the University of Pisa, Italy. L. Pallottino is also with the Research Center E. Piaggio of the University of Pisa. f.diluzio@studenti.unipi.it; lucia.pallottino@unipi.it

cells, convex admissible state constraints are defined for the optimization problem.

The two main benefits of using a grid-based SMPC approach for autonomous driving, compared to other SMPC approaches, are the following. First, it is not required to generate and consider an individual chance constraint for each vehicle or obstacle considered. The probability grid is generated given all obstacles and then the risk parameter threshold is applied to all cells, yielding a deterministic reformulation of the chance constrained optimization problem. This results in an optimization problem with convex state constraints, which can be solved efficiently. However, the stochastic nature of the problem is still accounted for as a probabilistic grid is initially generated. Second, it is not necessary to decide on a most likely behavior of other vehicles, as multiple predicted behavior options with arbitrary probability distribution can be considered. These properties facilitate the application to autonomous driving. The effectiveness of the presented approach is demonstrated in a highway simulation.

The remainder of the paper is structured as follows. Section II introduces the vehicle models and in Section III the method is derived in detail. Section IV shows a highway simulation with the proposed method, followed by conclusive remarks in Section V.

II. Vehicle Models

MPC requires system models for the prediction of futures states within the prediction horizon. We specifically consider trajectory planning for vehicles, where the controlled vehicle is known as the ego vehicle (EV) and surrounding vehicles as target vehicles (TVs). Focusing on vehicles allows to present the proposed gird-based SMPC method more comprehensively. However, TVs can be interpreted as dynamic obstacles in non-vehicle related trajectory planning tasks.

We consider a nonlinear EV model

$$\dot{\boldsymbol{\xi}}^{\text{EV}} = f\left(\boldsymbol{\xi}^{\text{EV}}, \boldsymbol{u}^{\text{EV}}\right) \tag{1}$$

with EV state $\boldsymbol{\xi}^{\text{EV}}$ and EV input $\boldsymbol{u}^{\text{EV}}$. For the discrete optimization problem model (1) needs to be discretized. Control constraints are imposed on both the steering angle and the acceleration, i.e., $\boldsymbol{u}_{\text{min}} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\text{max}}$, summarized as $\boldsymbol{u} \in \mathcal{U}$. The EV is subject to state constraints $\boldsymbol{\xi}^{\text{EV}} \in \Xi$, such as road restrictions, and specifically safety constraints $\boldsymbol{\xi}^{\text{EV}} \in \Xi^{\text{safe}}$, which ensure collision avoidance with other vehicles.

It is necessary for the EV to predict the future TV motion. It is assumed that the future TV motion is described by a linear discrete-time point-mass model subject to prediction uncertainty, similar to [11],

$$\boldsymbol{\xi}_{k+1}^{\text{TV}} = \boldsymbol{A}\boldsymbol{\xi}_{k}^{\text{TV}} + \boldsymbol{B}\boldsymbol{u}_{k}^{\text{TV}} + \boldsymbol{G}\boldsymbol{w}_{k}^{\text{TV}} \tag{2}$$

where $\boldsymbol{\xi}_{k}^{\text{TV}} = \left[x_k^{\text{TV}}, v_{x,k}^{\text{TV}}, y_k^{\text{TV}}, v_{y,k}^{\text{TV}}\right]^\top$ is the TV state at time step k, represented by longitudinal and lateral positions and velocities, and $\boldsymbol{u}_{k}^{\text{TV}} = \left[u_{x,k}^{\text{TV}}, u_{y,k}^{\text{TV}}\right]^\top$ is the control input consisting of longitudinal and lateral acceleration with the assumed to be known TV reference trajectory $\boldsymbol{\xi}_{k,\text{ref}}^{\text{TV}}$ and feedback law

$$\boldsymbol{u}_{k}^{\text{TV}} = \boldsymbol{K}\left(\boldsymbol{\xi}_{k}^{\text{TV}} - \boldsymbol{\xi}_{k,\text{ref}}^{\text{TV}}\right), \quad \boldsymbol{K} = \begin{bmatrix} 0 & k_{12} & 0 & 0 \\ 0 & 0 & k_{21} & k_{22} \end{bmatrix}. \tag{3}$$

The uncertainty in the prediction is taken into account by the random variable $\boldsymbol{w}_{k}^{\text{TV}}$ and $\boldsymbol{G} = \text{diag}\left(g_1, g_2, g_3, g_4\right)$.

The presented vehicle models are then used in the MPC optimization problem to predict the future EV and TV states.

III. Method

This section presents the main contribution of this work. Surrounding TVs have uncertain behavior. EV safety constraints must therefore consider the stochastic nature of the future TV motion to avoid collisions. Robustly accounting for the worst-case uncertainty results in overly conservative vehicle trajectories. Introducing chance constraints in SMPC allows to relax this conservatism, depending on a tunable risk parameter. The probabilistic chance constraints need to be reformulated into deterministic expressions, so that they can be solved within the optimization problem.

Assuming a grid representation of the environment, we derive a grid-based SMPC approach, which enables a simple approach to reformulate the probabilistic chance constraints into tractable constraints. First, for each time step of the SMPC prediction horizon a probabilistic occupancy grid is computed, where each cell of the grid represents the probability that the cell is occupied by a TV. This leads to the formulation of chance constraints to avoid collisions between the EV and TV. A tractable expression of the chance constraint is found by deriving a binary grid in order to clearly identify the admissible road grid cells and by finding a convex hull in which the EV can operate. Finally, we solve the optimal control problem of the SMPC. In the following subsections we describe the method in detail, starting with the probabilistic occupancy grid. Explicitly denoting the current time step k for states is omitted due to clarity. Prediction steps are indicated by the index h.

A. Probabilistic Grid

The environment is represented by a grid $\mathcal{G}$, i.e., an evenly spaced field of cells $\boldsymbol{c}_{i,j} \in \mathcal{G}$ with $\boldsymbol{c}_{i,j} = (c_i, c_j)$. Each cell has dimensions l_x and l_y accounting for length and width, respectively, and is identified in a 2D space by two indices i and j. The level of approximation depends on the size of the cell.

For every prediction step h an individual grid is generated, which is later used for collision avoidance. Here, the *Probabilistic Grid* (PG) is inspired by OGs but it is defined differently compared to standard OG literature. The PG, represented by a matrix $\mathcal{P}$, consists of elements $p_{i,j}$ which describe the occupancy probability. However, the probability value $p_{i,j}$ does not necessarily correspond to the exact probability that cell $\boldsymbol{c}_{i,j}$ is occupied by a TV. This is necessary as the proposed method later combines grids of individual TVs.

For simplicity, in the following we assume a Gaussian probability distribution for the TV motion prediction in order to demonstrate the method. However, it is possible to apply arbitrary probability distributions within the proposed method. The probability distribution, in this case a 2D Gaussian distribution, is used to assign probability values to the matrix $\mathcal{P}$. The probability density function is given by

$$f_{c^{\text{TV}}}\left(\boldsymbol{c}\right)=\frac{\exp\left(-\frac{1}{2}\left(\boldsymbol{c}-\boldsymbol{c}_h^{\text{TV}}\right)^\top\boldsymbol{\Sigma}_h^{-1}\left(\boldsymbol{c}-\boldsymbol{c}_h^{\text{TV}}\right)\right)}{\sqrt{(2\pi)^2|\boldsymbol{\Sigma}_h|}} \tag{4}$$

where $\boldsymbol{c}_h^{\text{TV}}$ is the cell corresponding to the estimated TV center at prediction step h and Σ_h is a covariance matrix.

The uncertainty $\boldsymbol{\Sigma}_h$ in (4) is propagated by using a recursive technique, described in [7],

$$\boldsymbol{\Sigma}_{h+1}=\left(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}\right)\boldsymbol{\Sigma}_h\left(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}\right)^\top+\boldsymbol{G}\boldsymbol{\Sigma}_w\boldsymbol{G}^\top \tag{5}$$

given the assumed TV prediction model (2), the initial condition $\boldsymbol{\Sigma}_0=\boldsymbol{0}$, and the uncertainty covariance matrix $\boldsymbol{\Sigma}_w$.

In order to consider the TV dimensions, we first compute probability values for each cell according to (4), displayed in Figure 1a. The black point represents $\boldsymbol{c}_h^{\text{TV}}$, i.e., the center of gravity (CoG) of the TV, the green dashed line represents the normal distribution on the xp-plane passing through $\boldsymbol{c}_h^{\text{TV}}$, while the red dashed line represents the normal distribution on the yp-plane. Then, we start to expand the maximum value computed with (4) along both dimensions x and y in order to cover the vehicle area. The final result is displayed in Figure 1b, where the distance between the two green dashed lines equals the width of the vehicle, and the distance between the red dashed lines equals the length. Note that Figure 1b does not show a probability density function, as the maximal value was artificially expanded.

In the following, the PG is extended for the case of multiple vehicles.

B. Multiple Target Vehicles and Maneuver Probabilities

The presented PG can easily be extended to scenarios with more than one TV by computing a PG for each TV and then adding up the probability values of the different grids for each cell. If a certain area on the road can potentially be occupied by more than one vehicle in the future, its probability to be occupied increases. This yields

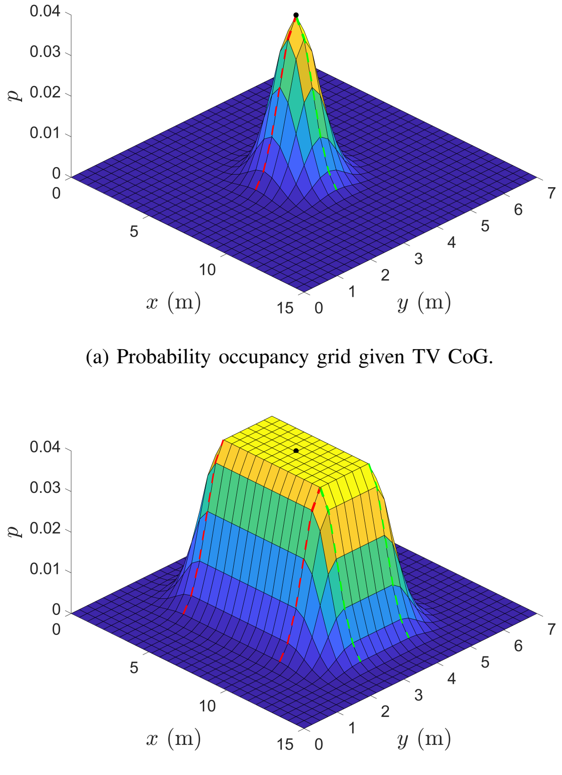
$$p_{i,j}=\sum_{n_{\text{v}}=1}^{N_{\text{v}}}p_{i,j,n_{\text{v}}} \tag{6}$$

where N_{v} is the number of TVs localized in the detection range of the EV.

We additionally consider N_{m} possible TV maneuvers. This is achieved by computing a PG for each maneuver and weigh it with the probability that this maneuver is actually performed, resulting in

$$p_{i,j}=\sum_{n_{\text{v}}=1}^{N_{\text{v}}}\sum_{n_{\text{m}}=1}^{N_{\text{m}}}\varepsilon_{n_{\text{v}},n_{\text{m}}}p_{i,j,n_{\text{v}},n_{\text{m}}} \tag{7}$$

where $\varepsilon_{n_{\text{v}},n_{\text{m}}}$ is the probability that the maneuver n_{m} is performed by TV n_{v}. A resulting example grid with two TVs is shown in Figure 2.



(a) Probability occupancy grid given TV CoG.

(b) Probability occupancy grid taking into account the TV dimensions.

Fig. 1: Illustration for building the TV probability occupancy grid. The black point represents $\boldsymbol{c}_h^{\text{TV}}$.

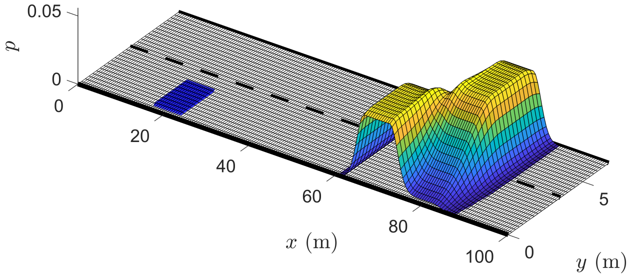


Fig. 2: Probabilistic Grid of a prediction step. The EV is on the left in blue and two TVs are displayed on the right.

In the following, the PG, considering all surrounding TVs, is adapted to be included in an SMPC optimization problem.

C. Binary Grid

In SMPC the probabilistic chance constraint is required to be reformulated into a deterministic expression. We consider the chance constraint

$$\Pr\left(\boldsymbol{\xi}^{\text{EV}}\in\Xi^{\text{safe}}\right)\geq\beta \tag{8}$$

where Ξ^{safe} is the set of state which guarantees collision avoidance. The safety constraint $\boldsymbol{\xi}^{\text{EV}}\in\Xi^{\text{safe}}$ is required to be satisfied with probability β, where β is a tunable risk parameter. Calculating a tractable, deterministic representation of the chance constraint can be computationally expensive and also challenging for non-Gaussian probability distributions.

To tackle this problem, we reformulate the chance constraints for the SMPC by utilizing the presented PG. We transform the probabilistic grid $\mathcal{P}$ into a *Binary Grid* (BG), represented by a matrix $\mathcal{B}$ with elements $b_{i,j}$, by imposing a probability threshold p_{th} similar to the risk parameter β,

$$b_{i,j} = \begin{cases} 1, & \text{if } p_{i,j} \geq p_{\text{th}}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The value 1 indicates that a certain cell $\boldsymbol{c}_{i,j}$ is considered to be occupied and therefore inadmissible. The parameter p_{th} is a trade-off between risk and conservatism: the lower the threshold, the more conservative the controller. By transforming the probabilistic grid $\mathcal{P}$ into the binary grid $\mathcal{B}$, we make a clear distinction between admissible and inadmissible space for the EV.

Now, the chance constraint in (8) can be expressed as a standard hard constraint

$$\boldsymbol{\xi}^{\text{EV}} \in \Xi^{\text{adm}} \tag{10}$$

where Ξ^{adm} includes all admissible cells in the BG. The constraint (10) can be handled in a general optimization problem.

This approach resembles the use of chance constraints in other SMPC approaches, as cells with low occupancy probability are neglected, depending on the threshold p_{th}. We then obtain constraints (10) that can be directly handled by a solver. In the next subsection we show how to derive a linear inequality description of the constraint in (10).

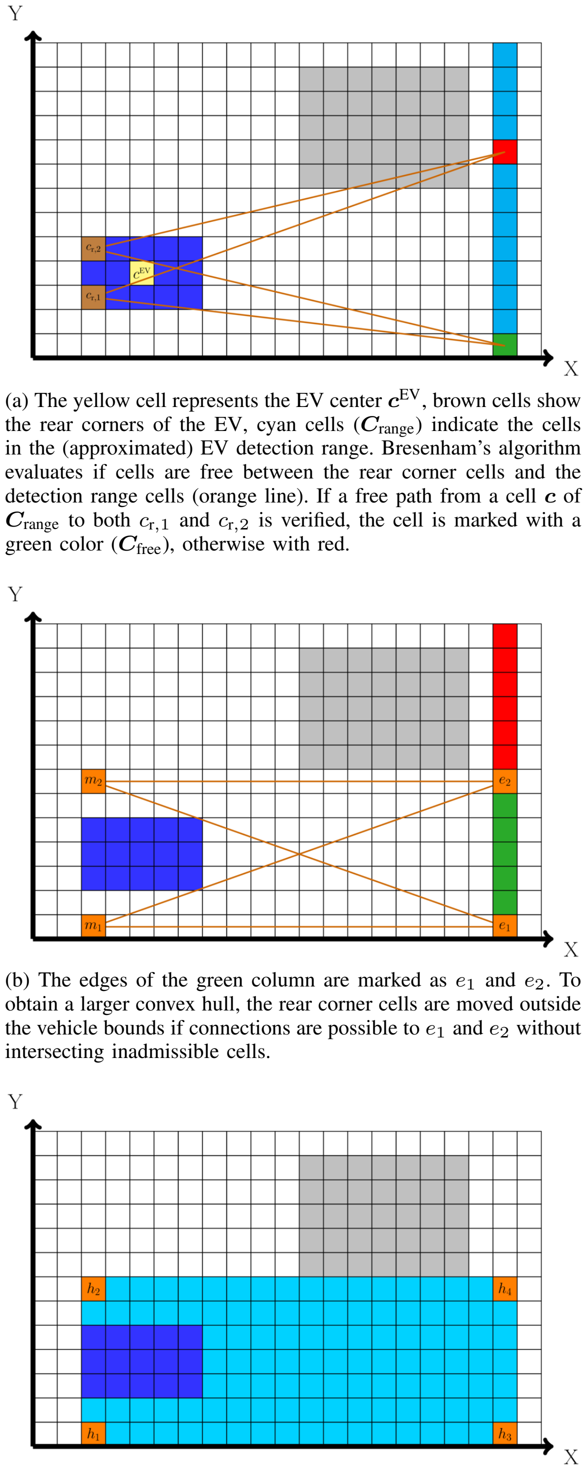
D. Convex Admissible Safe State Space

In order to obtain a fast SMPC framework, it is not sufficient to define a deterministic constraint reformulation, as seen above, it is also beneficial to find a linear inequality description of the safety constraint. This is achieved by applying Bresenham's line algorithm [18].

Given two points in a grid, which are connected by a straight line, Bresenham's line algorithm yields all cells which are touched by the connecting line. Even if it was developed in the field of computer graphics to select the bitmaps of an image, it is applicable here - instead of bitmaps we have grid cells.

The goal here is to find a convex hull for all admissible cells of the BG, applying Bresenham's algorithm. The convex hull, consisting of valid cells, can be described by linear constraints.

The basic steps are shown in Figure 3, where the blue cells represent the EV vehicle and the gray cells represent the inadmissible space according to (9). The scenario is a straight road with direction of motion along the x-axis. First, valid cells in the EV detection range are determined (Figure 3a). Then, using Bresenham's algorithm, it is checked whether the cells on the detection range boundary allow straight connecting lines to the EV without intersecting occupied cells. Second, the area of valid cells around the EV is enlarged (Figure 3b), resulting in a convex hull (Figure 3c). The detailed approach is described in Algorithm 1.



(a) The yellow cell represents the EV center $\boldsymbol{c}^{\text{EV}}$, brown cells show the rear corners of the EV, cyan cells ($\boldsymbol{C}_{\text{range}}$) indicate the cells in the (approximated) EV detection range. Bresenham's algorithm evaluates if cells are free between the rear corner cells and the detection range cells (orange line). If a free path from a cell $\boldsymbol{c}$ of $\boldsymbol{C}_{\text{range}}$ to both $c_{\text{r},1}$ and $c_{\text{r},2}$ is verified, the cell is marked with a green color ($\boldsymbol{C}_{\text{free}}$), otherwise with red.

(b) The edges of the green column are marked as e_1 and e_2. To obtain a larger convex hull, the rear corner cells are moved outside the vehicle bounds if connections are possible to e_1 and e_2 without intersecting inadmissible cells.

(c) The light-blue area displays the convex hull, the vertices are marked with orange cells.

Fig. 3: Illustration of the steps in Algorithm 1.

Algorithm 1 is used for each time step h of the prediction horizon N. It results in a linear inequality description of a

Algorithm 1 Admissible Safe State Space search

```
Input: $\boldsymbol{\xi}_h^{\text{EV}}$, $\mathcal{B}_h$
Identify EV center $\boldsymbol{c}^{\text{EV}}$ and select rear corners $c_{\text{r},1}$ and $c_{\text{r},2}$ (as in Figure 3a)
Select admissible cells in the detection range of the EV and store them in a matrix $\boldsymbol{C}_{\text{range}}$ (cyan column in Figure 3a)
for each cell $\boldsymbol{c}$ in $\boldsymbol{C}_{\text{range}}$ do
    Verify free path from $\boldsymbol{c}$ to $c_{\text{r},1}$, and from $\boldsymbol{c}$ to $c_{\text{r},2}$
    if free paths are confirmed then
        Store $\boldsymbol{c}$ in matrix $\boldsymbol{C}_{\text{free}}$
    end if
end for
Select edge cells $e_1$ and $e_2$ of $\boldsymbol{C}_{\text{free}}$ (as in Figure 3b)
Define $m_1$ as $c_{\text{r},1}$ and $m_2$ as $c_{\text{r},2}$
Move down $m_1$ as long as free path exists between $m_1$, $e_1$ and $m_1$, $e_2$
Move up $m_2$ as long as free path exists between $m_2$, $e_1$ and $m_2$, $e_2$
Select vertices of the convex hull $h_1$, $h_2$, $h_3$, $h_4$
Output: Convert vertex-representation into linear inequality representation (11)
```

convex hull, i.e.,

$$\boldsymbol{A}_h^{\text{safe}} \boldsymbol{\xi}_h^{\text{EV}} \leq \boldsymbol{b}_h^{\text{safe}}. \tag{11}$$

This safety constraint is now included in an MPC optimal control problem. Note that there is no guarantee that an admissible convex hull is found for each prediction step h within the prediction horizon N. It is, however, assumed that a convex hull can always be found for $h = 1$. If no admissible convex hull is found at prediction step h, the convex hull of step $h - 1$ is used.

E. Model Predictive Controller

Given the convex hull representing the admissible EV states, we can formulate a tractable SMPC optimization problem, which is denoted as a standard MPC optimization problem. The optimization problem is given by

$$\min_{\boldsymbol{U}} \sum_{h=0}^{N-1} \left(\left\| \Delta \boldsymbol{\xi}_h^{\text{EV}} \right\|_{\boldsymbol{Q}}^2 + \left\| \boldsymbol{u}_h^{\text{EV}} \right\|_{\boldsymbol{R}}^2 \right) + \left\| \Delta \boldsymbol{\xi}_N^{\text{EV}} \right\|_{\boldsymbol{S}}^2 \tag{12a}$$

$$\text{s.t.} \quad \boldsymbol{\xi}_{h+1}^{\text{EV}} = f\left(\boldsymbol{\xi}_h^{\text{EV}}, \boldsymbol{u}_h^{\text{EV}}\right), \quad h \in \mathbb{N}, \tag{12b}$$

$$\boldsymbol{\xi}_{h+1}^{\text{TV}} = \boldsymbol{A}\boldsymbol{\xi}_h^{\text{TV}} + \boldsymbol{B}\boldsymbol{u}_h^{\text{TV}} + \boldsymbol{G}\boldsymbol{w}_h^{\text{TV}}, \quad h \in \mathbb{N}, \tag{12c}$$

$$\boldsymbol{u}_h^{\text{TV}} = \boldsymbol{K}\left(\boldsymbol{\xi}_h^{\text{TV}} - \boldsymbol{\xi}_{h,\text{ref}}^{\text{TV}}\right), \quad h \in \mathbb{N}, \tag{12d}$$

$$\boldsymbol{u}_h^{\text{EV}} \in \mathcal{U}_h, \quad h = 0, \dots, N-1, \tag{12e}$$

$$\boldsymbol{\xi}_h^{\text{EV}} \in \Xi_h, \quad h = 1, \dots, N, \tag{12f}$$

$$\boldsymbol{A}_h^{\text{safe}} \boldsymbol{\xi}_h^{\text{EV}} \leq \boldsymbol{b}_h^{\text{safe}}, \quad h = 1, \dots, N, \tag{12g}$$

with the EV input $\boldsymbol{U} = (\boldsymbol{u}_0, \dots, \boldsymbol{u}_{N-1})^\top$, $\|z\|_{\boldsymbol{Z}}^2 = z^\top \boldsymbol{Z} z$ and $\Delta \boldsymbol{\xi}_h^{\text{EV}} = \boldsymbol{\xi}_h^{\text{EV}} - \boldsymbol{\xi}_{h,\text{ref}}^{\text{EV}}$ with maneuver-dependent EV reference $\boldsymbol{\xi}_{h,\text{ref}}^{\text{EV}}$, weighting matrices $\boldsymbol{Q}, \boldsymbol{S} \in \mathbb{R}^{4\times4}$ and $\boldsymbol{R} \in \mathbb{R}^{2\times2}$, EV prediction model $f\left(\boldsymbol{\xi}_h^{\text{EV}}, \boldsymbol{u}_h^{\text{EV}}\right)$ according to (1), and the input and state constraints $\mathcal{U}_h$ and Ξ_h. The TV model for the prediction is according to (2), (3).

In this optimization problem, the chance constraint (8) is reformulated into the tractable safety constraint (12g) according to (11).

IV. Results

This SMPC algorithm is now applied to an autonomous driving scenario. In the following, we first provide the general setup that has been used for the simulations. Then, we present the results of a first simulation, a highway scenario where an EV overtakes two TVs, to demonstrate the method. Eventually, we show the results of a second simulation with multiple TVs to analyze the computational cost obtained with the proposed grid-based SMPC method.

A. Simulation Setup

The presented method has been implemented in MATLAB® using the MPC routine developed in [19] as a base implementation. Each vehicle has been modeled as a rectangle with length and width of 6 m and 2 m. Simulations are run with sampling time $\Delta t = 0.2\,\text{s}$. The scenario is a two-lane highway with a lane width $l_{\text{lane}} = 3.5\,\text{m}$. Cell dimensions are $l_x = 0.5\,\text{m}$ and $l_y = 0.25\,\text{m}$.

We consider the EV model

$$\dot{x} = v \cos(\psi + \alpha), \tag{13a}$$

$$\dot{y} = v \sin(\psi + \alpha), \tag{13b}$$

$$\dot{\psi} = \frac{v}{l_\text{r}} \sin(\alpha), \tag{13c}$$

$$\dot{v} = a, \tag{13d}$$

$$\alpha = \arctan\left(\frac{l_\text{r}}{l_\text{r} + l_\text{f}} \tan(\delta_\text{f})\right) \tag{13e}$$

where x and y represent the longitudinal and lateral position of the vehicle CoG, v and a denote the longitudinal velocity and acceleration, α denotes the body slip angle, δ_f is the steering angle of the front wheels, l_r and l_f are the distances from the vehicle CoG to the rear and front axles, respectively. We denote with $\boldsymbol{\xi}^{\text{EV}} = [x, y, \psi, v]^\top$ and $\boldsymbol{u}^{\text{EV}} = [\delta_\text{f}, a]^\top$ the state and the control input of the EV. Model (13) is discretized by the forward-Euler method with sampling time Δt.

The EV employs two simple policies to decide on a reference lane $y_{h,\text{ref}}^{\text{EV}}$: 1) if the actual lane is occupied 20 m in front of it, the EV moves to the nearest free lane, otherwise it keeps its actual lane, 2) when it passes a TV and the longitudinal distance between their CoGs is larger than 15 m, the EV will overtake the TV by positioning itself in front of the TV.

The EV is subject to the following constraints $\boldsymbol{u}^{\text{EV}} \in \mathcal{U}$,

$$y \in [1\ ,\ 6]\,\text{m}, \tag{14a}$$

$$\delta_\text{f} \in [-3\ ,\ 3]\,\text{deg}, \tag{14b}$$

$$a \in [-5\ ,\ 5]\,\text{m/s}^2, \tag{14c}$$

in addition to the safety inequality constraints (11).

We consider a time-discrete point-mass TV prediction model for (2) with

$$\boldsymbol{A} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0.5\,(\Delta t)^2 & 0 \\ \Delta t & 0 \\ 0 & 0.5\,(\Delta t)^2 \\ 0 & \Delta t \end{bmatrix}. \quad (15)$$

The selected TV controller matrix values are $[k_{12}, k_{21}, k_{22}] = [-1, -0.8, -2.2]$. We assume Gaussian noise $\boldsymbol{w}_k^{\text{TV}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_w)$ with covariance matrix $\boldsymbol{\Sigma}_w = \text{diag}\,(1,1,1,1)$ and disturbance matrix $\boldsymbol{G} = \text{diag}\,(0.05, 0.067, 0.013, 0.03)$. These choices for the TVs are similar to [11].

Here, only the two most likely TV maneuvers are considered: a lane keeping (LK) and a lane changing (LC) maneuver with constant longitudinal velocity where each of them is weighted with a probability value for the respective maneuver being executed, as shown in Sec. III-B. Note that more maneuvers could be considered. Here, we randomly assign a probability in the range of [0.8 , 1] to one of the predicted maneuvers. The second maneuver is given a probability such that the sum equals one.

The SMPC has a prediction horizon $N = 20$, weighting matrices $\boldsymbol{Q} = \text{diag}\,(0, 2, 0.5, 0.1)$ and $\boldsymbol{R} = \text{diag}\,(0.1, 1)$, and a probability threshold $p_{\text{th}} = 0.15$.

Algorithm 1 is used to find a convex hull at each time step h of the prediction horizon N. If at a generic time step h_t a convex hull is not found, we consider the one calculated at the preceding time step $h_t - 1$. Therefore, Algorithm 1 is based on the assumption that a convex hull can always be found at step $h = 0$.

B. Overtaking Scenario

This scenario consists of a straight two-lane road with two TVs. The center of the right lane is set to $1.75\,\text{m}$, the left lane to $5.25\,\text{m}$. The EV is positioned on the left lane with initial state $\boldsymbol{\xi}^{\text{EV}} = [10, l_{\text{ref}}, 0, 26]$, where $l_{\text{ref}} = 5.25\,\text{m}$. The two TVs start with the initial states $\boldsymbol{\xi}^{\text{TV}_1} = [40, 27, 5.25, 0]$, and $\boldsymbol{\xi}^{\text{TV}_2} = [90, 27, 1.75, 0]$. TV_1 is positioned on the left lane, TV_2 on the right one. A probability value 0.8 is assigned to the LK maneuver and 0.2 to the LC maneuver for both TVs. In the simulation the TVs follow the maneuver with the higher probability. Therefore, both TVs will actually proceed along their lane.

Figures 4, 5, and 6 show the simulation results. Figure 4 illustrates the vehicle motion, Figure 5 displays the EV velocity and steering angle, while Figure 6 shows the distance between the EV and the two TVs. At the beginning the EV accelerates to reach its reference velocity of $30\,\text{m/s}$, as shown in the first plot of Figure 5. When the lane is occupied by TV_1 $20\,\text{m}$ in front of the EV, the EV starts a LC maneuver, moving to the right lane, as shown in Figure 4a. In Figure 6a it can be seen that once the maneuver is completed, the longitudinal distance between EV and TV_1 is approximately $16\,\text{m}$. As soon as it passes TV_1 and the distance between their CoGs is larger than $15\,\text{m}$, the reference lane for the EV changes and the EV starts to move towards the left lane to finish overtaking TV_1. Figure 4b shows a screenshot of this phase. When the EV has completed the maneuver, i.e., when it lies completely in the left lane, the longitudinal distance with respect to TV_1 is about $17\,\text{m}$. Once it reaches and passes TV_2, the EV performs a new LC maneuver by moving to the right lane again. Once the EV lies on the right lane, the relative distance between the CoGs of the two vehicles is approximately $21\,\text{m}$. This last phase is represented in Figure 4c.

For the presented scenario, and with the given policies, we can see that there is no deceleration by the EV while performing LC maneuvers, but only changes in the steering angle ψ. Collisions are avoided. As the simulated (not the predicted) TV motion is deterministic in this scenario, one simulation is sufficient for the proposed method. For the presented grid-based SMPC method, additional simulations with identical initialization result in the same behavior, in contrast to SMPC methods based on sampling.

Given this setup and the policy to compute the reference lane for the EV, the MATLAB® solver *fmincon* always finds a feasible solution. Therefore, bounds on control signal and space constraints are respected. However, it is important to mention that these results are obtained with the choice $p_{\text{th}} = 0.15$, and no further research has been conducted on different values. The effect of varying risk parameters is studied in other SMPC works, e.g., [7], [11].

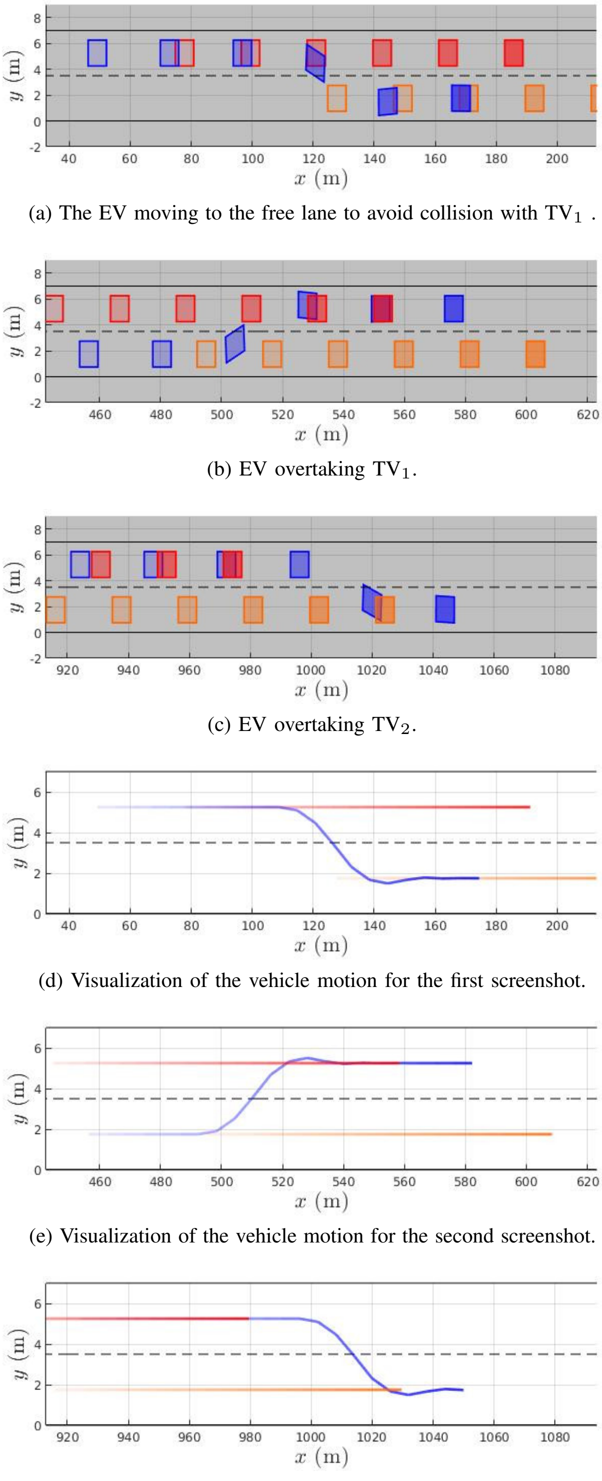
C. Computational cost evaluation

To evaluate the computational cost of the algorithm, we use the following more complex setup. The general setup is the one shown in Section IV-B. For each simulation, the EV is randomly positioned on one of the two lanes and a random reference lane is assigned. The same applies to each TV. The first TV is positioned in front of the EV with a longitudinal distance of $40\,\text{m}$. If more TVs are simulated, these are positioned every $50\,\text{m}$. For each TV, one probability value is sampled in the range [0.8 , 1] and assigned randomly to one of the two maneuvers, LC or LK, and the second value is assigned to the other maneuver such that the sum of the probabilities equals one. The actual behavior of a TV follows the maneuver with higher probability. We simulated three scenarios with one, two, and three TVs, and each of them has been run 10 times on a standard desktop computer with an Intel i5 processor (3.3 GHz).

Table I shows the mean μ and the standard deviation σ of the algorithm computation time per iteration. By increasing the number of TVs in the EV detection range

TABLE I: Mean and standard-deviation per algorithm iteration for scenarios with a different number of TVs .

# of TVs	μ (s)	σ (s)
1 TV	0.45	0.48
2 TVs	0.44	0.43
3 TVs	0.42	0.36



(a) The EV moving to the free lane to avoid collision with TV$_1$.

(b) EV overtaking TV$_1$.

(c) EV overtaking TV$_2$.

(d) Visualization of the vehicle motion for the first screenshot.

(e) Visualization of the vehicle motion for the second screenshot.

(f) Visualization of the vehicle motion for the third screenshot.

Fig. 4: Screenshot of the simulation during the overtaking maneuvers. Blue boxes represent the EV, red and orange boxes represent TV$_1$ and TV$_2$, respectively. Fading boxes show past states.

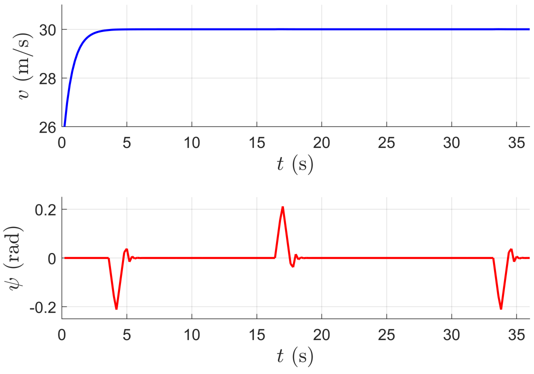
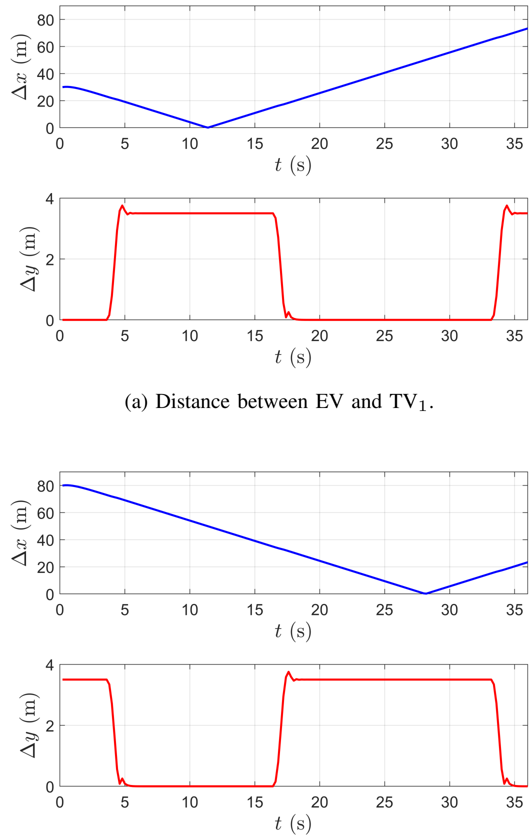


Fig. 5: EV longitudinal velocity and steering.



(a) Distance between EV and TV$_1$.

(b) Distance between EV and TV$_2$.

Fig. 6: Longitudinal and lateral distance between EV and TVs in absolute value terms.

for the scenario, the computation cost mean μ remains almost constant, while the standard deviation σ shows larger variations. The computational cost of the algorithm is mainly due to the complexity of the nonlinear EV model (13), and not dependent on the number of TVs in the scenario. The computational effort generating the PG and calculating the convex hull is comparatively small, as this is done prior to solving the optimization problem. This is a major advantage over for example [11], where the computation time increases significantly with an increasing number of TVs.

D. Discussion

In Section III-C we introduced the *probability threshold* parameter p_{th}, which allows to transform the PG into a BG, in order to obtain a deterministic approximation of the chance constraints (8). This parameter is a trade-off between conservatism and risk. By setting a low value for p_{th}, a high number of cells will be considered occupied. At a certain step, the road can seem fully occupied, resulting in a conservative maneuver for the EV. On the other hand, a high value of p_{th} will reduce the number of occupied cells considered by the algorithm and, at certain step of the prediction horizon, the road can seem free. This can yield a more aggressive controller and potentially a collision between the vehicles.

Therefore, the parameter p_{th} has to be chosen by evaluating a trade-off between conservatism and risk for the planning algorithm, depending on the kind of distribution one decides to adopt. It can also be beneficial to apply a time-varying probability threshold, adapting to different situations. In general, selecting a suitable threshold is challenging, similar to choosing a risk parameter in other SMPC approaches. Note that the risk parameter and the considered probability values do not perfectly represent true probabilities.

In the simulations a nonlinear EV model is used to increase the accuracy of the EV predictions, whereas a simpler, linear TV model is used in combination with noise, as the TV behavior is subject to uncertainty. However, applying a linearized EV prediction model allows to solve a QP problem, given the safety constraint (11), a quadratic cost function, as well as linear input and state constraints. This is highly beneficial when a fast algorithm is needed that still considers stochastic behavior of surrounding vehicles.

V. Conclusion

We presented a novel and simple approach to apply SMPC for trajectory planning in uncertain environments, by using a probabilistic grid. This allows efficiently planning ego vehicle trajectories, while considering stochastic behavior of surrounding objects. The proposed method scales well with an increasing number of objects considered, here shown for three vehicles, and can handle arbitrary probability distributions of future object motion. It is still of interest to obtain simulation results for more complex scenarios and target vehicle maneuvers, as well as combine the proposed approach with occupancy grids.

While we applied the proposed method to autonomous vehicles, other applications are possible. It is especially interesting to apply the grid-based SMPC approach to three-dimensional applications, e.g., trajectory planning for robots.

References

[1] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168, Baden-Baden, Germany, June 2011.

[2] C. Katrakazas, M. Quddus, W. Chen, and L. Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442, 2015.

[3] B. Gutjahr, L. Gröll, and M. Werling. Lateral vehicle trajectory optimization using constrained linear time-varying mpc. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1586–1595, June 2017.

[4] A. Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems*, 36(6):30–44, Dec 2016.

[5] G. Schildbach, L. Fagiano, C. Frei, and M. Morari. The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica*, 50(12):3009 – 3018, 2014.

[6] T. Brüdigam, J. Teutsch, D. Wollherr, and M. Leibold. Combined robust and stochastic model predictive control for models of different granularity, 2020. arXiv: 2003.06652.

[7] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli. Stochastic predictive control of autonomous vehicles in uncertain environments. In *12th International Symposium on Advanced Vehicle Control*, Tokyo, Japan, 2014.

[8] D. Lenz, T. Kessler, and A. Knoll. Stochastic model predictive controller with chance constraints for comfortable and safe driving behavior of autonomous vehicles. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 292–297, Seoul, South Korea, June 2015.

[9] G. Schildbach and F. Borrelli. A dynamic programming approach for nonholonomic vehicle maneuvering in tight environments. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 151–156, June 2016.

[10] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli. Scenario model predictive control for lane change assistance and autonomous driving on highways. *IEEE Intelligent Transportation Systems Magazine*, 9(3):23–35, Fall 2017.

[11] T. Brüdigam, M. Olbrich, M. Leibold, and D. Wollherr. Combining stochastic and scenario model predictive control to handle target vehicle uncertainty in autonomous driving. In *21st IEEE International Conference on Intelligent Transportation Systems*, Maui, USA, 2018.

[12] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, Cambridge, Mass., 2005.

[13] C. Coue, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessiere. Bayesian occupancy filtering for multi-target tracking: an automotive application. *International Journal of Robotic Research - IJRR*, 25, 01 2006.

[14] M.K. Tay, K. Mekhnacha, M. Yguel, C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière. *Probabilistic Reasoning and Decision Making in Sensory-Motor Systems*, chapter The Bayesian Occupation Filter, pages 77–98. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[15] M. Saval-Calvo, L. Valdés, J. Castillo-Secilla, S. Cuenca-Asensi, A. Martínez-Álvarez, and J. Villagra. A review of the bayesian occupancy filter. *Sensors*, 17, 03 2017.

[16] S. Steyer, G. Tanzmeister, and D. Wollherr. Grid-based environment estimation using evidential mapping and particle tracking. *IEEE Transactions on Intelligent Vehicles*, 3(3):384–396, Sep. 2018.

[17] S. Steyer, C. Lenk, D. Kellner, G. Tanzmeister, and D. Wollherr. Grid-based object tracking with nonlinear dynamic state and shape estimation. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–20, 2019.

[18] J. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4:25–30, 1965.

[19] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control.* Springer-Verlag, London, 2017.