Probabilistic Constraint Tightening Techniques for Trajectory Planning with Predictive Control

Nathan Goulet[a,*], Qian Wang[b], Beshah Ayalew[a]

[a]*Department of Automotive Engineering, Clemson University, 4 Research Dr., Greenville, 29607, SC, USA*
[b]*Autonomous Driving Solutions (ADS), Huawei, Shanghai, China*

Abstract

In order for automated mobile vehicles to navigate in the real world with minimal collision risks, it is necessary for their planning algorithms to consider uncertainties from measurements and environmental disturbances. In this paper, we consider analytical solutions for a conservative approximation of the mutual probability of collision between two robotic vehicles in the presence of such uncertainties. Therein, we present two methods, which we call unitary scaling and principal axes rotation, for decoupling the bivariate integral required for efficient approximation of the probability of collision between two vehicles including orientation effects. We compare the conservatism of these methods analytically and numerically. By closing a control loop through a model predictive guidance scheme, we observe through Monte-Carlo simulations that directly implementing collision avoidance constraints from the conservative approximations remains infeasible for real-time planning. We then propose and implement a convexification approach based on the tightened collision constraints that significantly improves the computational efficiency and robustness of the predictive guidance scheme.

Keywords: probabilistic collision criteria, constraint tightening, chance constraints, motion trajectory planning, predictive control

*Corresponding Author

Email addresses: `ngoulet@clemson.edu` (Nathan Goulet), `beshah@clemson.edu` (Beshah Ayalew)

1. Introduction

A core undertaking for safely guiding an autonomously controlled vehicle (ACV) within the real world amongst other dynamic agents, objects or object vehicles (OVs), is that of devising real-time feasible collision free motion plans. Guidance and trajectory planning algorithms for an ACV operating in dynamic environments need to observe/track and predict the motion and extent of neighboring OVs [1]. Starting with available measurements such as sparse laser point clouds [2], OVs can be tracked by treating each as an extended object [3, 4]. With knowledge of an OV's shape and motion states (for example fusing with camera images and other sensors), it can be represented by an estimated geometric/spatial shape to formulate enforceable collision avoidance constraints in the motion trajectory planning problem for the ACV [5, 6]. However, the problem is still quite challenging to solve in real-time due to the following principal difficulties: 1) The planning problem involves uncertainties due to modeling errors, sensor imperfections and/or environmental disturbances; and; 2) Since the desired feasible field for planning maneuvers is defined to be outside of the area occupied by OVs and other (road or workspace) boundaries, the collision avoidance problem is generally non-convex.

The first difficulty involving uncertainties is often handled by considering the known/assumed bounds (non-deterministic uncertainty case) or their probability distributions (probabilistic case) [7]. In the non-deterministic case [8], the worst case of the uncertainty is considered in the trajectory planning problem leading often to a very conservative solution. In the probabilistic case [9], the conservatism can be relaxed by specifying probabilistic collision criteria which are often posed as chance/probabilistic collision avoidance constraints with a given threshold. However, the evaluations of these collision criteria are often computationally intractable due to the multivariate integrals involved in the probabilistic collision avoidance criteria. Nevertheless, for a specified confidence threshold, a solution can be sought by solving approximate deterministic trajectory planning problems with either a sampling-based method (Monte Carlo Simulations) [9] or with constraint-tightening methods [10, 11].

In constraint-tightening methods, the original probabilistic collision avoidance (chance) constraint is replaced by a deterministic constraint that is a function of the confidence threshold and the probability distribution of the uncertainty. The key step in these methods is to determine the form of this

function. With assumptions of uncorrelated Gaussian distributions for the state observations, an explicit function is easy to obtain by factorizations into univariate integrals [11]. For the case of correlated state uncertainties, [12] derived an approximate explicit function for probabilistic collision evaluation of small-sized objects (radius smaller than 1 m). However, this approximation does not work for agents of larger geometric sizes, e.g. ACVs operating in public road traffic. Our prior work in [10] provided a framework to handle agents of non-negligible sizes with 2D rectangular axes-aligned shapes. Therein, the computations of the mutual probability of collision between OVs of non-negligible sizes is approximated analytically via coordinate rotations. In this paper, we will refer to this method as principal axes rotation. A drawback of this framework is that it does not account for the relative orientation of agents and the associated uncertainty. This limitation will be addressed in this paper by drawing from a similar approach proposed in [13] that accounts for heading state uncertainty assuming independence of the heading state from the translational states. Therein, the mutual probability of collision is approximated via a linear coordinate transformation to achieve unit covariance. In this paper, we will refer to this latter method as unitary scaling. Both principal axes rotation and unitary scaling methods attempt to achieve the same goal of constraint tightening via conservative approximations of the collision avoidance criteria. In this paper, we characterize the conservatism introduced by each method to identify guidance on their respective preferred use scenarios.

Approaches to address the second difficulty (non-convexity of planning space) vary by the choice of the planning method. Commonly cited sampling-based planning methods like the A* algorithm [14] and RRT* algorithm [15] use polygonal models [9, 16], which are a disjunction of linear constraints, or algebraic models like circles, ellipses [17] and hyper ellipses [18]. These sampling-based methods suffer from the computational burden of the large number of samples that may be needed for sufficient resolution of the relevant geometry for real-time guidance in complex traffic. Alternatives to these planning methods are mathematical constrained-optimization based planning methods like model predictive control (MPC) [19] that solve on-line guidance optimizations subject to the constraints of the measured and model predicted evolution of traffic. While polygonal representations of objects can be adopted with these methods, they are less convenient since the disjunction of linear constraints that would be needed to represent multi-vehicle traffic often lead to non-smoothness and/or discontinuity in the solution space. This

often result in Disjunctive Linear Programming problems [20], which require specific solvers to find a solution. On the other hand, if one finds analytical approximations for the mutual probability of collision criteria between objects, it is possible to directly implement those as tightened constraint functions in the motion trajectory planning task (as in [13]). However, this may still result in excessive execution time and non-robust behavior, as we will show later in Section 3.2 with an MPC-based planner. For this reason, we further propose convexification steps on top of the analytical approximations that improve execution time while still guaranteeing the probability of collision threshold is not violated by the planner.

The main contributions of this paper can be summarized as follows:

- Detailed discussions on the analytical approximation of the mutual probability of collision avoidance criteria between two agents.

- Compare two methods for decoupling the bivariate integral that appears in the collision avoidance criteria.

- Demonstration of the shortcomings of a chance-constrained nonlinear MPC-based motion trajectory planner that directly implements the approximated probabilistic collision avoidance criteria as tightened constraints.

- Propose a constraint convexification method that significantly enhances the computational speed and robustness of predictive planners that use the constraint-tightening approximations.

The rest of the paper is organized as follows. Section 2 derives the two methods for analytical approximation of the probability of collision criteria and characterizes the associated conservatism in each. Section 3 first outlines the chance-constrained nonlinear MPC-based motion-trajectory planner and demonstrates its performance with a direct implementation of the results from Section 2. Then, it proposes the convexification method and demonstrates the associated improvements. Section 4 presents the conclusions of the paper and identifies some avenues for further work.

2. Conservative Approximation of Probabilistic Collision Avoidance Criteria

Figure 1 shows a schematic of the overall process for evaluating probabilistic collision avoidance constraints. We assume that there are measurements
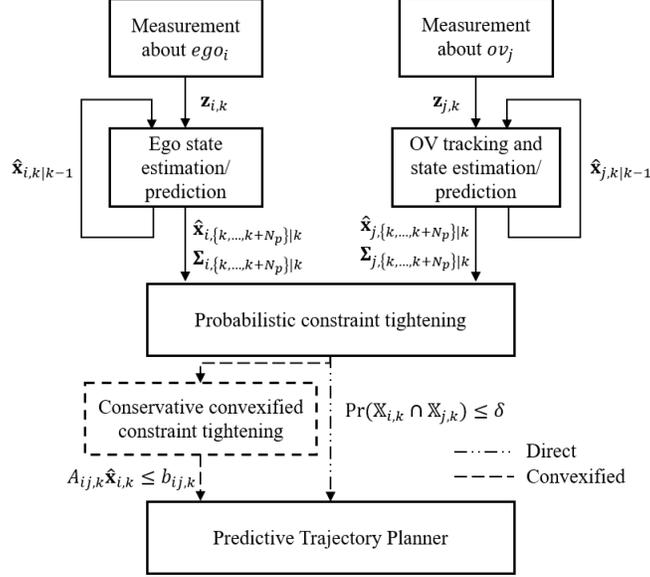
Figure 1: Probabilistic collision avoidance constraint tightening framework.

$\mathbf{z}_j$ available for each object vehicle, ov_j, in the field of view of the ego vehicle, ego_i. We also assume that each ov_j is tracked from these measurements and we can obtain an estimation of the current state and prediction of the future states of each ov_j and ego_i. As object vehicle tracking, and state estimation and prediction are not a primary contribution of this paper we omit further discussion of that here. The reader is referred to [21, 22, 23] and references therein. We note that estimated/predicted state vector $\hat{\mathbf{x}}_j$ and associated covariance matrices Σ_j are assumed to be available for each ov_j over the time horizon N_p. Using this information about the state and uncertainty of ov_j, along with the estimated and predicted state trajectory and covariance of ego_i, the probabilistic constraint tightening block in Figure 1 conducts the main computations we detail in the next subsection. In Section 3, we will first overview how to directly implement the probabilistic constraint $\Pr\left(\mathbb{X}_{i,k} \cap \mathbb{X}_{j,k}\right) \leq \delta$ within a predictive trajectory planner (dash-dotted line in Figure 1), where $\mathbb{X}$ is the 2D space occupied by a vehicle considering its geometric shape and δ is the probability of collision threshold. Then, we will present our conservative convexification method (dashed lines in Figure 1). Either the direct probabilistic constraint or the convexified constraint $(A_{ij,k}\hat{x}_{i,k} \leq b_{ij,k})$ may be applied over the predicted state trajectories with
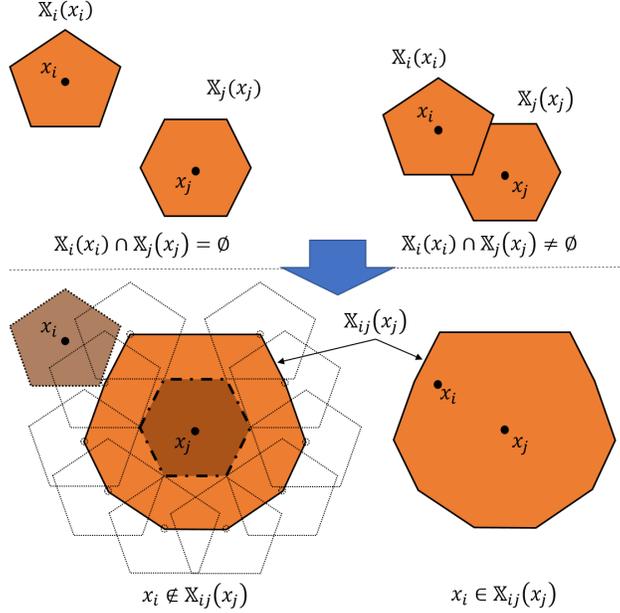
Figure 2: Example of a collision free condition (left) and collision condition (right) between an ego vehicle (pentagonal here) ego_i and object vehicle (hexagonal) ov_j.

the time indexed by k as in Figure 1.

We will begin by formulating a conservative approximation (upper bound) on the probability of collision between ego_i and ov_j. For our purposes, hereafter, x represents the 2D position state for the centroid of the geometric shape for a given vehicle, unless specifically noted otherwise.

2.1. General Formulation of the Probabilistic Collision Avoidance Criteria

We can state that a collision occurs between ego_i and ov_j, if $\mathbb{X}_i(x_i) \cap \mathbb{X}_j(x_j) \neq \emptyset$ (at any time k, where the subscript k is omitted in the following to reduce clutter). This condition is depicted on the right-hand side of Figure 2. Any arbitrary shape may be considered for the vehicles at this point. The criteria for the probability of ego_i entering a collision state with ov_j to not exceed a threshold δ is then:

$$\Pr\left(\mathbb{X}(x_i) \cap \mathbb{X}(x_j) \neq \emptyset\right) \leq \delta. \tag{1}$$

Determining the intersection of two arbitrary shapes, $\mathbb{X}_i(x_i) \cap \mathbb{X}_j(x_j)$, analytically can be cumbersome. One common approach is to lump the geometric

shapes of the ego_i and ov_j into a combined shape $\mathbb{X}_{ij}(x_j)$ centered at x_j as the Minkowski sum [24] of the underlying shapes (i.e. $\mathbb{X}_i \oplus \mathbb{X}_j = \mathbb{X}_{ij}$). Then ego_i can simply be considered as a point located at x_i. See the lower portion of Figure 2. The collision probability threshold criteria becomes:

$$\Pr(x_i \in \mathbb{X}_{ij}(x_j)) \leq \delta; \tag{2}$$

where the probability of collision between ego_i and ov_j is defined by [21]:

$$\Pr(x_i \in \mathbb{X}_{ij}(x_j)) = \iint I_c(x_i, x_j)\, p_{x_i x_j}(x_i, x_j)\, dx_i dx_j. \tag{3}$$

Here, $p_{x_i x_j}$ is the joint probability density function (PDF) of the random variables x_i and x_j for the position states, and $I_c(x_i, x_j)$ is a collision indicator function defined as:

$$I_c(x_i, x_j) = \begin{cases} 1 & \text{if } x_i \in \mathbb{X}_{ij}(x_j), \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Now using the definition of the joint PDF, we can rewrite $p_{x_i x_j}(x_i, x_j) = p_{x_i}(x_i|x_j)\, p_{x_j}(x_j)$. This, with the definition of I_c, allows us to replace the inner indefinite integral of (3) with a definite integral over the shape of $\mathbb{X}_{ij}(x_j)$ to obtain the general formulation for the probability of collision:

$$\Pr(x_i \in \mathbb{X}_{ij}(x_j)) = \int \left[\int_{x_i \in \mathbb{X}_{ij}(x_j)} p_{x_i}(x_i|x_j)\, dx_i \right] p_{x_j}(x_j)\, dx_j. \tag{5}$$

2.2. Approximation of the Probability of Collision for Gaussian Uncertainty and Car-like Rectangular Geometries

The above general form in (5) is typically quite difficult to evaluate efficiently for vehicles of non-negligible sizes and arbitrary shapes using onboard computing resources. To expedite computation of the probability of collision online some simplifying assumptions need to be invoked. We make two assumptions here:

Assumption 1: OVs are rectangular in shape and their position states follow Gaussian distributions. This assumption aligns with common practical autonomous vehicle guidance schemes that use lidar- or vision-based perception techniques with bounding boxes for object detection and tracking [22, 25] along with Gaussian noise assumptions. Following these practices:

7

- Both ego_i and ov_j are described as car-like rectangular shapes;

- The PDFs of the position x and heading ϕ states of ego_i and ov_j are independent and available from the respective estimation routines as Gaussians.

We designate the normal PDF with $\mathcal{N}(\mu, \Sigma)$, where μ is the mean and Σ the covariance. In the discussion that follows, we use the state variable in the subscript to differentiate which mean or covariance is intended. We note that the assumption of rectangular shapes is not restrictive, as other geometries may be bounded by a rectangle or divided into multiple sub-regions, each of which is bounded by a rectangle. Further, the motions of the vehicles may have a coupled (interactive) dynamics in traffic for which interaction-aware planning methods have been proposed elsewhere [26, 27]. While this is not the focus of the present paper, we note that one could estimate the uncertainty in the joint states/behaviors of ego_i and ov_j as in [28] and extend the decoupling approach detailed in this paper for bivariate distributions to higher dimensions and proceed to utilize the proposed constraint tightening framework. This extension is left as a possible future research direction.

When we consider both ego_i and ov_j to be rectangular, instead of mere points or circles, it is necessary to consider the orientation relative to the coordinate frame as well. We can incorporate heading information in our modeling with the following approach. Suppose ϕ_i and ϕ_j are the respective heading angles of the two vehicles relative to a road fixed frame (see top of Figure 3). We rotate the frame of reference to align with the ego vehicle and define the relative heading angle $\phi_{ji} = \phi_j - \phi_i$ with mean $\mu_{\phi_j} - \mu_{\phi_i}$ and co-variance $\Sigma_{\phi_j} + \Sigma_{\phi_i}$ (see middle of Figure 3). Let $\mathbb{X}_{ij}(x_j, \phi_{ji})$ be the combined geometric description located at the 2D position state x_j and constructed according to the relative orientation ϕ_{ji} (see bottom of Figure 3). Applying Baye's rule, the probability of collision can be formulated as:

$$\Pr\left(x_i \in \mathbb{X}_{ij}(x_j)\right) = \int p_{\phi_{ji}}(\phi_{ji})\Pr\left(x_i \in \mathbb{X}_{ij}(x_j, \phi_{ji})\,|\phi_{ji}\right)d\phi_{ji}. \qquad (6)$$

Proposition 1: Given independent Gaussians x_i and x_j, the probability of collision conditioned on relative heading angle ϕ_{ji} is:

$$\Pr\left(x_i \in \mathbb{X}_{ij}(x_j, \phi_{ji})\,|\phi_{ji}\right) = \int_{x_{ij}\in\mathbb{X}_{ij}(0,\phi_{ji})} \mathcal{N}_{x_{ij}}\left(\mu_{x_{ij}}, \Sigma_{x_{ij}}\right)dx_{ij}, \qquad (7)$$
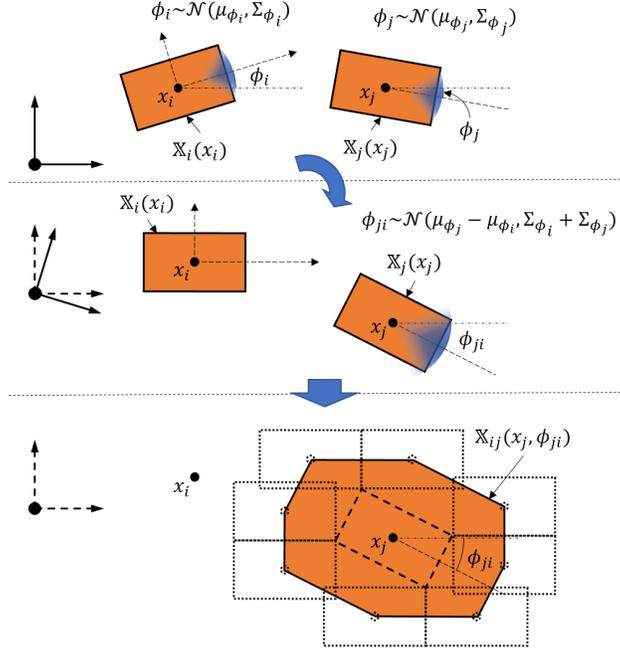
8

Figure 3: Depiction of heading angles ϕ_i and ϕ_j (top), the rotation to align the reference frame with ego_i and the resulting relative heading angle ϕ_{ji} (middle), and the combined shape $\mathbb{X}_{ij}(x_j, \phi_{ji})$ (bottom).

where $x_{ij} = x_i - x_j$ is the relative position state variable with mean $\mu_{x_{ij}} = \mu_{x_i} - \mu_{x_j}$, and covariance $\Sigma_{x_{ij}} = \Sigma_{x_i} + \Sigma_{x_j}$.

Proof: Proposition 1 was originally proved in [21]. ■

When considering heading deviations, it is still cumbersome to compute the collision probability defined by (6) and (7) (for real time use), due to the double integral and the fact that the integration bounds in (7) depend on ϕ_{ji} and are irregular. This leads us to our next pragmatic assumption.

Assumption 2: We use conservative embedding of ov_j within a circumscribed rectangle aligned with the axes of ego_i considering a specified range of the relative heading angle ϕ_{ji}.

This assumption is utilized in such a way that the combined geometric description $\mathbb{X}_{ij}(0, \phi_{ji})$ remains rectangular, thereby simplifying the evaluation of the integral in (7). There are, however, two remaining difficulties: first is the consideration of uncertainty in relative heading angle, and second is the coupling of the uncertainty in the two dimensions of the position space.

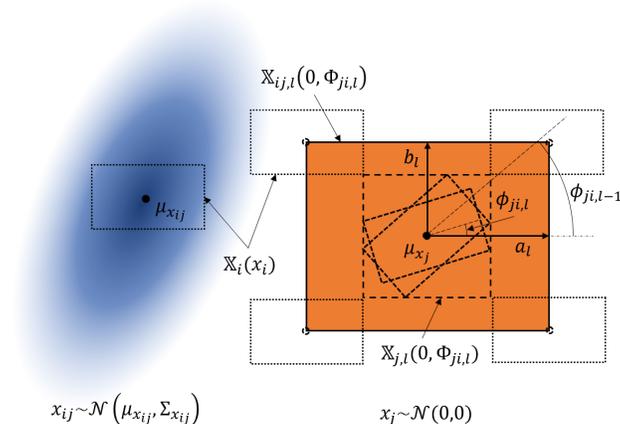To address the first difficulty, we adopt a method proposed in [13]. We

9

Figure 4: Visualization of the conservative embedding of ov_j within an ego_i axes-aligned circumscribing rectangle for ϕ_{ji} interval l.

consider the support of the relative heading ϕ_{ji} and discretize it into n_ϕ intervals $\Phi_{ji,l} = [\phi_{ji,l-1}, \phi_{ji,l}], l \in \{1, \ldots, n_\phi\}$, from $\phi_{ji,0} = \mu_{\phi_{ji}} - \pi/2$ to $\phi_{ji,n_\phi} = \mu_{\phi_{ji}} + \pi/2$. As ϕ_{ji} is Gaussian, and by definition has an infinite support, we define $\Phi_{ji,0} = [-\infty, \phi_{ji,0}]$ and $\Phi_{ji,n_\phi+1} = [\phi_{ji,n_\phi}, \infty]$, thus $\sum_{l=0}^{n_\phi+1} \Pr(\phi_{ji} \in \Phi_{ji,l}) = 1$. For each heading angle interval $\Phi_{ji,l}, l \in \{0, \ldots, n_\phi+1\}$, the combined shape $\mathbb{X}_{ij}(x_j, \Phi_{ji,l})$ is constructed by a Minkowski sum of the shape of ego_i and the rectangle that circumscribes the convex hull of the shape of ov_j over the interval (see Figure 4). We denote the half-length and half-width of $\mathbb{X}_{ij}(0, \Phi_{ji,l})$ by a_l and b_l, respectively.

An upper bound on the probability of collision, considering the n_ϕ discrete intervals of ϕ_{ji}, can then be obtained as:

$$\Pr(x_i \in \mathbb{X}_{ij}(x_j)) \leq \sum_{l=0}^{n_\phi+1} \Pr(\phi_{ji} \in \Phi_{ji,l}) \Pr(x_i \in \mathbb{X}_{ij}(x_j, \Phi_{ji,l}) \,|\, \Phi_{ji,l}). \quad (8)$$

The upper bound in (8) is significantly easier to evaluate than (6) as the coupled integration over ϕ_{ji} and the 2D deviation state x_{ij} is replaced by a (probability) weighted summation of sub-integrations over only the 2D position states.

The second difficulty remains as the collision probability given a specific relative heading angle interval still requires bivariate integration over the 2D position states. To overcome this, we seek a transformation of the

10

deviation state variable into a reference frame that decouples/factorizes the integration of the bivariate Gaussian distribution into univariate integrals. Specifically, we are interested in obtaining a transformation applied to the deviation variables that results in a diagonalized covariance matrix. As such a transformation may result in an irregular integration region $\mathbb{X}'_{ij}(0, \Phi_{ji,l})$, we will also determine the rectangular region $\mathbb{X}''_{ij}(0, \Phi_{ji,l})$ that conservatively embeds $\mathbb{X}'_{ij}(0, \Phi_{ji,l})$. We explore two methods that accomplish this, unitary scaling (US) and principal axes rotation (PA).

2.2.1. Decoupling Method 1 – Unitary Scaling

US normalizes the covariance of the deviation state (i.e. $\Sigma_{x'_{ij}} = T\Sigma_{x_{ij}}T^T = I$, where I is the 2 by 2 identity matrix), in order to allow decoupling in any direction [13, 29]. One transformation matrix that accomplishes this is given by $T_0 = \Sigma_{x_{ij}}^{-1/2}$ (see Appendix B for a discussion on how to compute T_0). The transformation T_0, as depicted in the middle left hand portion of Figure 5, can be shown to be composed from three linear elementary operations: rotation, scaling and shear; or $T_0 = RSH$, where R is a rotation matrix, S a scaling matrix, and H a shear matrix (see more below). Scaling does not alter the shape, i.e. it maintains a rectangular integration region. Rotation rotates the rectangular integration region with respect to the coordinate axes, however, it does not alter the shape. Taking advantage of the fact that we can decouple x'_{ij} in any direction, we can undo the rotation using the transformation R^{-1} to align $\mathbb{X}'_{ij}(0, \Phi_{ji,l})$ with the coordinate axes. See the bottom left of Figure 5. (With a slight abuse of notation, we designate the axes of the new reference frame by 1 and 2, however, it should be noted that this is not the same frame as the principle axes 1 and 2 mentioned later for Method 2.) Conversely, shear does alter the shape of $\mathbb{X}'_{ij}(0, \Phi_{ji,l})$. As shear maintains parallelism, the (tightest) bounding box $\mathbb{X}''_{ij}(0, \Phi_{ji,l})$ in the transformed coordinate frame contains either: the two edges along the length of the transformed combined shape $\mathbb{X}'_{ij}(0, \Phi_{ji,l})$, we will call this case 1 (shear occurs along axis 1 – depicted on the left of Figure 5); or the two edges along the width, which we call case 2 (shear occurs along axis 2 – not depicted).

To efficiently implement US, we can reduce the number of matrix operations by taking advantage of the decomposition of T_0 into RSH (see Appendix C for how to compute R, S, and H for both case 1 and case 2). Instead of applying the transformation T_0 and rotating the reference frame by R^{-1}, we
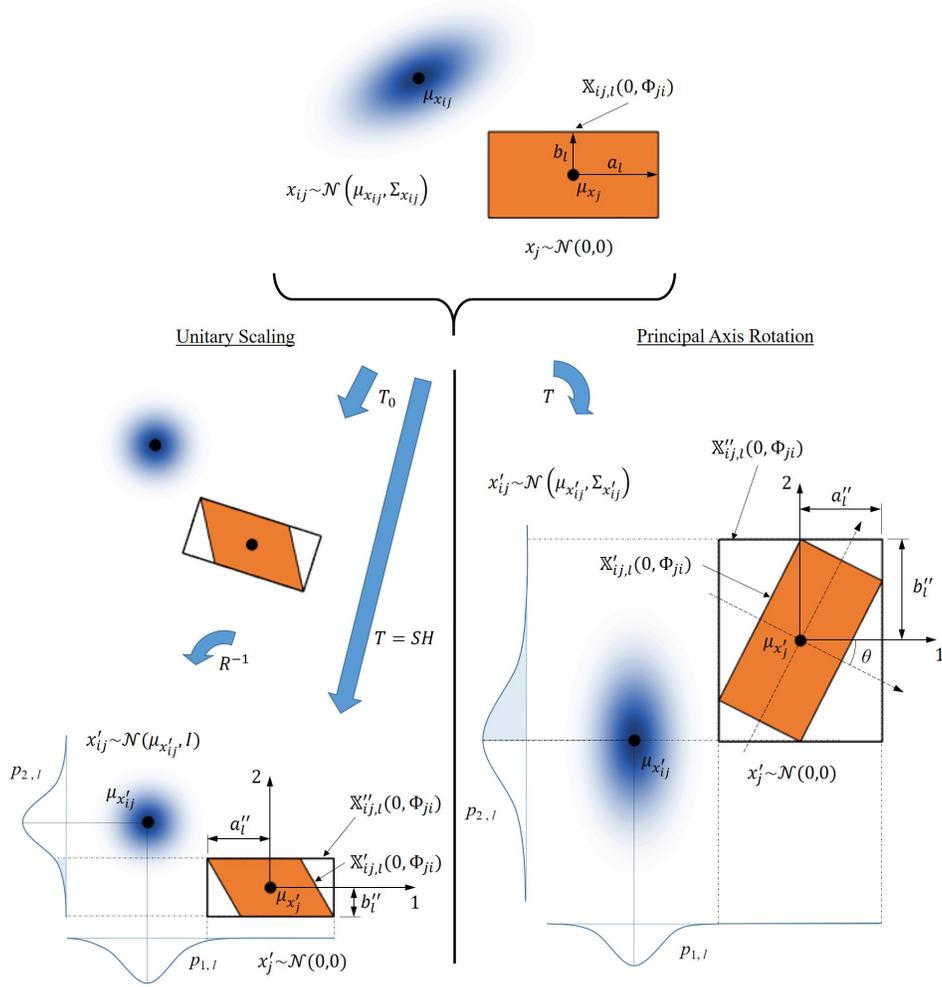
Figure 5: Visualization of the decoupling methods, unitary scaling (left), principal axis rotation (right).

can simply apply the transformation $T = SH$, which takes the form:

$$
SH = \begin{cases} \begin{bmatrix} s_1 & s_1h \\ 0 & s_2 \end{bmatrix} & \text{for case 1,} \\[2ex] \begin{bmatrix} s_1 & 0 \\ s_2h & s_2 \end{bmatrix} & \text{for case 2.} \end{cases} \tag{9}
$$

Based on the assumed forms of R, S, and H, it is possible to solve for s_1, s_2,

12

and h as functions of the eigenvalues λ_c, $c \in \{1, 2\}$, and components of the first eigenvector v_{1c}, $c \in \{1, 2\}$ of the original covariance matrix $\Sigma_{x_{ij}}$. The complete derivation for s_1, s_2, and h may be found in Appendix C.1 for case 1 and Appendix C.2 for case 2, while the relationships between the elements of a covariance matrix Σ and its eigenvalues and eigenvectors may be found in Appendix A.

2.2.2. Decoupling Method 2 – Principal Axes Rotation

The second method attempts to decouple the bivariate Gaussian distribution of the deviation state by a rotation to align the coordinate axes with the principal axes 1 and 2 of the uncertainty. The transformation is then $T = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, where the angle θ to the principal axes is the direction of an eigenvector of the covariance matrix $\Sigma_{x_{ij}}$ [30]. The transformed covariance is then $\Sigma_{x'_{ij}} = T\Sigma_{x_{ij}}T^T = D$, where D is the diagonal matrix composed of the eigenvalues of the original covariance matrix $\Sigma_{x_{ij}}$. Let θ correspond to the eigenvector associated with the first eigenvalue along the diagonal of D. The reader is directed to Appendix A or our paper [10] for detailed equations for this method. The right half of Figure 5 depicts this rotation.

After the transformation for either decoupling method, a rectangle $\mathbb{X}''_{ij}(0, \Phi_{ji,l})$ aligned with axes 1 and 2 is found that circumscribes the transformed integration region $\mathbb{X}'_{ij}(0, \Phi_{ji,l})$ (depicted for both methods in Figure 5). We can then solve for the half-length and half-width of the bounding box of the transformed shape, a''_l and b''_l, respectively, with the following relationship:

$$\begin{bmatrix} a''_l \\ b''_l \end{bmatrix} = \mathrm{abs}\,(T) \begin{bmatrix} a_l \\ b_l \end{bmatrix}, \tag{10}$$

where we use $\mathrm{abs}\,(\cdot)$ to describe the elementwise absolute value function. Note that T is the rotation matrix for PA and $T = SH$ for US. The bivariate integral in (7) becomes a product of two univariate integrals:

$$\Pr\left(x_i \in \mathbb{X}_{ij}(0, \Phi_{ji}) \,|\phi_{ji,l}\right) \leq \prod_{c=1}^{2} \int_{\underline{x}'_{c,l}}^{\overline{x}'_{c,l}} \mathcal{N}_{x'_c}\left(\mu_{x'_c}, \sigma^2_{x'_c}\right) dx'_c, \tag{11}$$

where $x'_{ij} = \left[x'_1, x'_2\right]^T$ is the transformed relative position state, and $\overline{x}'_{c,l}$ and $\underline{x}'_{c,l}$ are the upper and lower bounds of the conservative transformed shape $\mathbb{X}''_{ij,l}(0, \Phi_{ji,l})$ in the c-direction, $c \in \{1, 2\}$ ($\pm a''_l$ along axis 1 and $\pm b''_l$ along

13

axis 2). For future use, we will denote the integral (or the factors) on the right-hand side of (11) as $\Pr\left(x'_c \in \left[\underline{x}'_{c,l}, \overline{x}'_{c,l}\right]\right)$. Using the result in (11) and the definition of the cumulative distribution function (CDF) of the normal distribution, we can update the upper bound in (8) as:

$$\Pr\left(x'_{ij} \in \mathbb{X}''_{ij}(0)\right) = \sum_{l=0}^{n_\phi+1} \left[\Pr\left(\phi_{ji} \in \Phi_{ji,l}\right) \prod_{c=1}^{2} \Psi\left(\frac{x'_c - \mu_{x'_c}}{\sigma_{x'_c}}\right) \Bigg|_{\underline{x}'_{c,l}}^{\overline{x}'_{c,l}}\right], \quad (12)$$

where $\Psi(z)$ is the CDF of the standard normal and $\sigma_{x'_c}$ is the standard deviation. With (12), we have now derived an analytical solution for an upper bound on the mutual probability of collision between ego_i and ov_j.

2.3. Comparison of Decoupling Methods

We now investigate the shape contours for different (constant) probabilities of collision approximated with the two decoupling methods. To obtain the true probability of collision for comparison purposes, we utilize Monte-Carlo Simulations. Specifically, we estimate the true probability of collision by sampling n_{MS} instances of the position and heading states of the ego_i and ov_j (both with mean $\mu_{x,\phi}$ and joint covariance $\Sigma_{x,\phi}$) and then calculating the indicator function (4) and the probability of collision by:

$$\Pr\left(x_i \in \mathbb{X}_{ij}\right) = \frac{1}{n_{MS}} \sum_{k=0}^{n_{MS}} I_c\left(\begin{bmatrix} x_{i,k} \\ \phi_{i,k} \end{bmatrix}, \begin{bmatrix} x_{j,k} \\ \phi_{j,k} \end{bmatrix}\right) \qquad (13)$$

Figure 6 shows the collision probability contours approximated using the unitary scaling case 1 (US1), principal axes (PA) rotation, and Monte-Carlo Simulations (MS) methods. We see that both approximation methods (US1, PA) are conservative as the true probability contours are contained within the contours of the approximations.

In the following, we seek to characterize the level of conservatism for both decoupling methods. This can be very important to consider for online motion planning tasks, as the choice of decoupling method could mean the difference between finding a feasible path in the available computing time or not. Substituting the respective definitions of T for US and PA into (10)
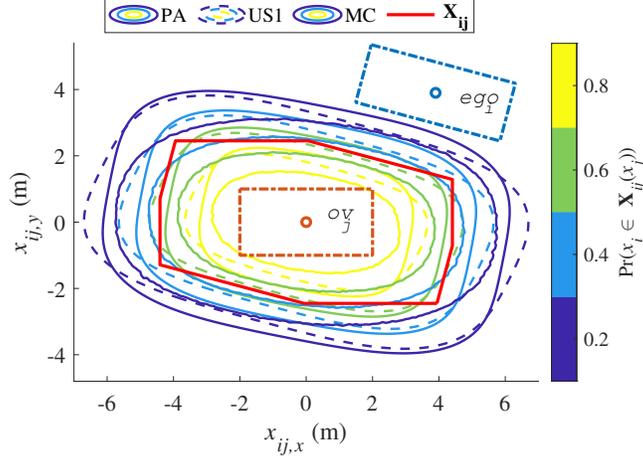
14

Figure 6: Contour of collision probabilities between ego_i and ov_j using unitary scaling case 1 (US1), principle axes rotation (PA), and Monte-Carlo simulations (MC); $n_{MS} = 10,000$ samples were used for the MS method; for the approximation methods $n_\phi = 20$ was used.

yields:

$$
\begin{bmatrix} a'' \\ b'' \end{bmatrix} = \begin{cases} \begin{bmatrix} |s_1|\, a + |s_1 h|\, b \\ |s_2|\, b \end{bmatrix} & \text{for US case 1,} \\[2ex] \begin{bmatrix} |s_1|\, a \\ |s_2 h|\, a + |s_2|\, b \end{bmatrix} & \text{for US case 2,} \\[2ex] \begin{bmatrix} |\cos\theta|\, a + |\sin\theta|\, b \\ |\sin\theta|\, a + |\cos\theta|\, b \end{bmatrix} & \text{for PA.} \end{cases}
\tag{14}
$$

To compare the two decoupling methods, we emphasize that both cases of US scale the original dimensions, whereas PA does not. As scaling does not result in any conservatism, we will compare a'' and b'' of PA to $a''/|s_1|$ and $b''/|s_2|$ of either US case. We do this both analytically and numerically.

First for PA, it is easy to show geometrically, or by taking derivatives of (14) with respect to θ, that both a'' and b'' are bounded in the set $\left[\min\{a, b\}, \sqrt{a^2 + b^2}\right]$. If $b < a$, then the minimum of a'' occurs at the angles $\theta \in \{\pi/2 \pm k\pi\}$ for all $k \in \mathbb{I}_{\geq 0}$, and the minimum of b'' occurs at $\theta \in \{\pm k\pi\}$ for all $k \in \mathbb{I}_{\geq 0}$, where $\mathbb{I}_{\geq 0}$ is the set of positive integers including 0. If $a > b$, then the angles obtaining the minimum of a'' and b'' are each shifted by $\pi/2$. The maximum of a'' occurs at $\theta \in \{\tan^{-1}(b/a) \pm k\pi\}$ for all
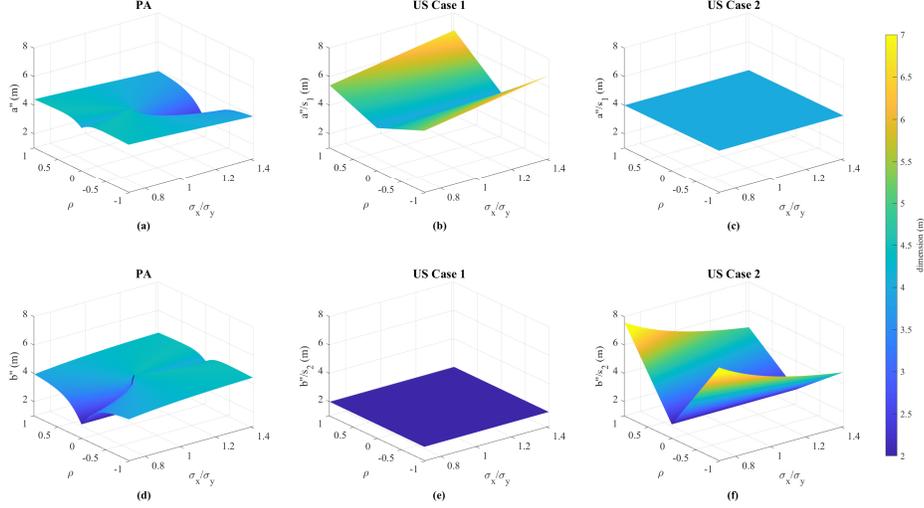
15

Figure 7: Plot of a'' (top row) and b'' (bottom row), as a function of correlation coefficient ρ and σ_x/σ_y for principal axis rotation (left), unitary scaling case 1 (center), and unitary scaling case 2 (right), with original dimension $a = 4$ and $b = 2$.

$k \in \mathbb{I}_{\geq 0}$, while the maximum of b'' occurs at $\theta \in \{\tan^{-1}(a/b) \pm k\pi\}$ for all $k \in \mathbb{I}_{\geq 0}$. Further, if we look at the area of the transformed integration region for PA, $a'' b''$, we can see that it is bounded in $[ab, 0.5(a^2 + b^2) + ab]$, where the minimum area occurs at the angles $\theta \in \{\pm k\pi/2\}$ for all $k \in \mathbb{I}_{\geq 0}$, and the maximum area occurs at the angle $\theta \in \{\pi/4 \pm k\pi/2\}$ for all $k \in \mathbb{I}_{\geq 0}$.

Now investigating US, it is easy to see from (14) that $b''/|s_2| = b$ for case 1 and $a''/|s_1| = a$ for case 2, i.e. there is no conservatism in the respective directions. Therefore, all of the conservatism can be lumped into $a''/|s_1|$ for case 1 and $b''/|s_2|$ for case 2. Specifically, the conservatism is proportional to the shear h, therefore we seek to analyze how h (case 1 (C.4c) and case 2 (C.7c) in Appendix C) behaves with respect to λ_1, λ_2, v_{11}, and v_{12}. As we chose unit eigenvectors, we know $v_{11}, v_{12} \in [0, 1]$ and $v_{11}^2 + v_{12}^2 = 1$. The denominator of h, for either case, is then a weighted combination of the eigenvalues, while the numerator is bounded with respect to v_{11} and v_{12}. Therefore, it is of interest to understand how h behaves with respect to the eigenvalues λ_1 and λ_2. In theory, if the eigenvalues tend towards infinity, it can be shown that h tends towards either $\pm v_{12}/v_{11}$ or $\mp v_{11}/v_{12}$ for either case (see Appendix C). As both are unbounded this means the conservatism of US is theoretically unbounded in one direction. However, in practice, the

spectral radius of $\Sigma_{x_{ij}}$ is (or the eigenvalues are) typically bounded meaning practical bounds on h can be computed.

Recalling that the eigenvalues and eigenvectors are a function of the standard deviations σ_x and σ_y, as well as the correlation coefficient ρ, we present plots of the dimensions of $\mathbb{X}_{ij}''$ as a function of ρ and σ_x/σ_y in Figure 7. As presented analytically, Figure 7 (a) and (d) support that a'' and b'' for PA are bounded, and therefore the corresponding conservatism is bounded. Then referencing Figure 7 (b) and (e), we see how $b''/|s_2|$ remains constant while $a''/|s_1|$ increases, apparently unbounded, as $|\rho|$ and σ_x/σ_y increase for US case 1. Similar behavior is observed for the opposite dimensions in Figure 7 (c) and (f) for US case 2.

The key observations are the following: in unstructured environments, where ego_i may need to navigate around an obstacle in any direction, the PA decoupling method is beneficial as the conservatism is easily bounded in any direction. However, in applications where only one direction is of relative importance, US guarantees minimal conservatism in one or the other direction. For example, if the application is autonomous driving on a multi-lane highway, US case 1 will guarantee minimal lateral conservatism allowing ego_i to pass (or be passed by) OVs in adjacent lanes. On the other hand, on a single lane highly populated road where the ego_i is following an OV using adaptive cruise control (ACC), US case 2 is beneficial as it allows for minimal conservatism in the longitudinal direction, and therefore accommodate longitudinally dense traffic.

3. Application in Closed-Loop Trajectory Planning

3.1. Nonlinear MPC-based Motion Planning Formulation

To demonstrate the use of the above constraint tightening methods, we formulate a chance constrained nonlinear MPC (CC-MPC) that solves the trajectory planning problem:

$$\min_{\mathbf{x}_i,\mathbf{u}} \mathbb{E}\left[(\mathbf{y}-\mathbf{r})^T\mathbf{Q}(\mathbf{y}-\mathbf{r}) + \mathbf{u}^T\mathbf{R}\mathbf{u}\right] \tag{15a}$$

Subject to:

$$\dot{x}_i = f\left(x_i, u, w\right), x_i \in \mathcal{X}, u \in \mathcal{U}, w \in \mathcal{W} \tag{15b}$$

$$y_i = Cx_i + \nu, \nu \in \mathcal{V} \tag{15c}$$

$$x_i\left(0\right) = \hat{x}_i\left(t\right) \tag{15d}$$

$$\Pr\left(\mathbb{X}\left(x_i\right) \cap \mathbb{X}\left(x_j\right) \neq \emptyset\right) \leq \delta \;\; \forall j \in \{1, ..., n_{ov}\} \tag{15e}$$

$$\mathbb{E}\left[c_2\left(x_i, u\right)\right] \geq 0. \tag{15f}$$

The problem in (15a) minimizes the expected error in output tracking, where $\mathbf{y}$, $\mathbf{r}$, and $\mathbf{u}$ are, respectively, the outputs, references to be tracked, and inputs augmented over the prediction horizon. Let $\mathbf{Q}$ and $\mathbf{R}$ be block diagonal matrices composed of the symmetric positive semi-definite weighting matrices Q_k and R_k at each time step $k \in \{1, \ldots, n_p\}$. The constraints include: 1) the nonlinear motion dynamics model and the feasible state and input bounded sets, $\mathcal{X}$ and $\mathcal{U}$, respectively, in (15b), with process noise w sampled from the set $\mathcal{W}$; 2) the output model in (15c), with measurement noise ν sampled from the set $\mathcal{V}$; 3) the initial condition in (15d); 4) the probabilistic collision avoidance constraints in (15e), and 5) nonlinear expectation constraints are modeled by (15f).

Here the ego vehicle state $x_i = \begin{bmatrix} s & y_e & \phi & v & a & \gamma \end{bmatrix}^T$ includes the position state (longitudinal position s and lateral position y_e), heading angle ϕ, velocity v, acceleration a, yaw rate γ, and the inputs include desired acceleration a_d and yaw rate γ_d. We assume full state measurement, i.e. C is the 6 by 6 identity matrix, however, the optimal control problem only tracks y_e and v references. The kinematics are governed by a particle motion model in the Frenet frame, with the dynamics associated with vehicle mass, moment of inertia, drivetrain, and steering system lumped into first order lag dynamics for a and γ. For a detailed discussion of the complete nonlinear motion model of (15b) and nonlinear constraints of (15f), including friction limits, the reader is directed to our prior work in [19]. We also defer the discussion of the transformation of the stochastic CC-MPC problem above into a deterministic one (with collision constraints tightened with our approach) to references elsewhere [31, 32, 33].

We utilize an unscented Kalman filter (UKF) to estimate the mean current state of ego_i and the idea of the most likely measurement [34] to propagate the covariance over the prediction horizon. The predicted ov_j states include $x_j = \begin{bmatrix} s & y_e & v & \phi \end{bmatrix}^T$. For estimating and predicting x_j and its co-
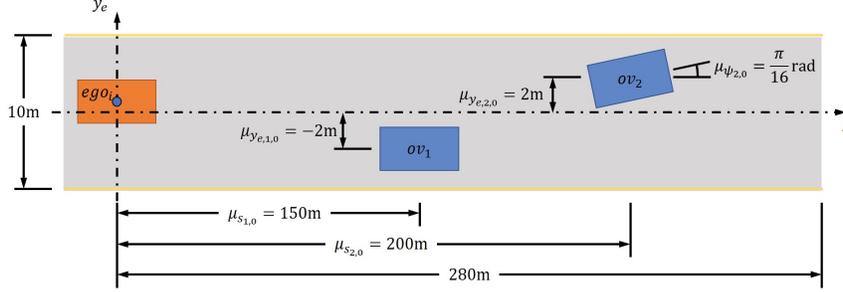
18

Figure 8: Illustrative scenario.

variance we use a UKF coupled with the nonlinear motion model from [19].

3.2. Direct Implementation of the Probabilistic Constraint

For a direct implementation of the probabilistic constraint, we replace the general probability on the left-hand side of (15e) with the analytical upper bound on the probability of collision in (12). To simplify calculations within the MPC, we utilize the prior time step plan of x_i and the prediction of x_j to precompute $\Sigma_{x_{ij}}$, ϕ_{ji}, and $\Sigma_{\phi_{ji}}$. This allows us to calculate the integration bounds a_l'' and b_l'' for every l relative heading range $\Phi_{ji,l}$ and each time step prior to the MPC call. We solve the optimal control problem (OCP) using the MPCTools [35] interface with CasADi [36] and the IPOPT solver [37].

To illustrate how the constraint tightening framework and CC-MPC perform, we set up an example simulation scenario with an autonomously controlled ego vehicle driving down a road and avoiding two stationary object vehicles within its path. The specific roadway with obstacles is presented in Figure 8. The roadway is 280m long and 10m wide. The ego vehicle ego_i starts at an initial position state randomly perturbed about the means $\mu_{s_{i,0}} = 0$m, $\mu_{y_{e,i,0}} = 0.5$m, and $\mu_{\phi_{i,0}} = 0$rad, with variances of $\Sigma_{s_{i,0}} = 0.1\text{m}^2$, $\Sigma_{y_{e,i,0}} = 0.1\text{m}^2$, and $\Sigma_{\phi_{i,0}} = 0.01\text{rad}^2$. Further, ego_i has an initial velocity of $v_{i,0} = 17$ m/s. The CC-MPC controlling the ego vehicle tracks the lane centerline with $r_{y_e} = 0$m and a reference velocity $r_v = 20$m/s. The plant model for the ego_i is the same as that used within the CC-MPC. The stationary OVs are then located as follows: ov_1 has a mean initial position state of $\mu_{s_{1,0}} = 150$m, $\mu_{y_{e,1,0}} = -2$m, and $\mu_{\phi_{1,0}} = 0$rad; while, ov_2 has a mean initial position state of $\mu_{s_{2,0}} = 200$m, $\mu_{y_{e,2,0}} = 2$m, and $\mu_{\phi_{2,0}} = \pi/16$rad. The initial position states of both stationary OVs are randomly perturbed about the means with the same variance as the ego_i.
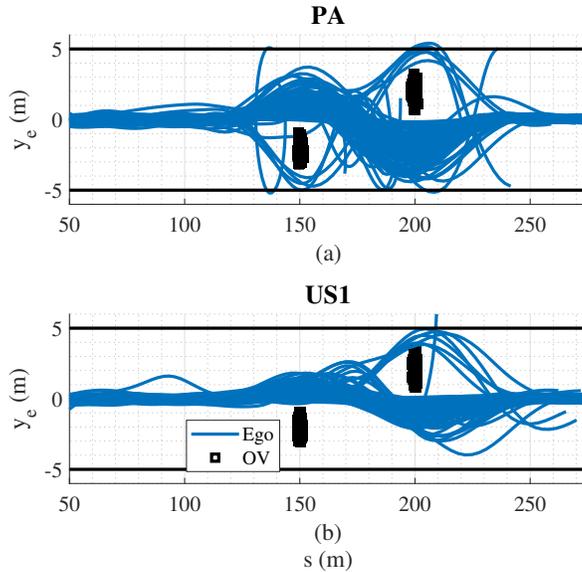
19

Figure 9: Simulation trajectories for direct implementation of the tightened probabilistic collision constraint using (a) PA decoupling and (b) US1 decoupling.

The constraint tightening framework and CC-MPC is updated every 0.15 simulation seconds, and the CC-MPC is discretized with 0.15s time step with a horizon of 6s, for a total of 40 time steps in the horizon. We choose a probability of collision threshold of $\delta = 0.001$ and discretize the relative heading space with $n_\phi = 20$. For comparison of the two decoupling methods described in Section 2.2, 100 simulations each using US1 and PA decoupling methods were completed. The results for the 100 simulations using each decoupling method are plotted within Figure 9.

From Figure 9 we can see that the CC-MPC, using either decoupling method, does not always find a solution that keeps the vehicle traveling down the roadway. When comparing the trajectories for PA and US1 decoupling, in Figure 9 (a) and (b), respectively, we can see that the US1 results in more robust behavior. Specifically, US1 only has one instance where a solution is not found, while PA results in several. Furthermore, with either decoupling method the CC-MPC does not always pass the OVs on the same side. The reason US1 decoupling performs more robustly is because it is less conservative along the relevant (lateral) dimension resulting in additional free space for planning, as discussed earlier in Section 2.3. The statistics of the max
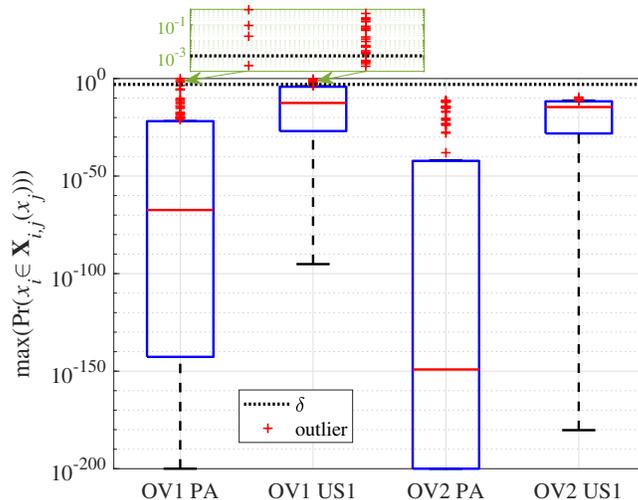
20

Figure 10: Box plot of max observed probability of collision between the ego_i and ov_1 and ov_2 for each simulation with direct implementation of the tightened probabilistic constraint.

probability of collision in Figure 10 support this notion as the medians (red lines) and quartiles (upper and lower extents of the blue box) for US1 decoupling are greater than those for PA. We note that there are outliers which exceed the probability threshold δ in Figure 10, as a solution is not always found.

Another drawback is that the implementation of the approximated probabilistic collision avoidance constraint with either method is still computationally expensive. The simulations presented here were run on a modern laptop with an Intel® Core™ i7-7820HQ 2.9 GHz quad core processor with 16.0GB RAM. The median computation time for the CC-MPC with PA decoupling over the 9,300 calls of the solver was 0.142s. This is on the border of the update rate of 0.15s, resulting in the CC-MPC not meeting real-time requirements more than 45% of the time. This is improved with US1 decoupling, where the median is 0.126s and real-time requirements are not met with 30% of the CC-MPC calls. To improve the robustness and computational efficiency of our CC-MPC we develop convexified tightened collision avoidance constraints as described below.
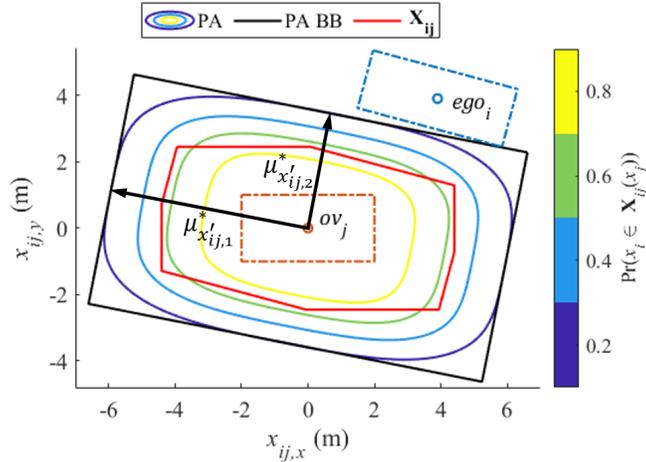
21

Figure 11: Depiction of the tightened constraint bounding box for the principal axis approach. The statistics match those in Figure 6.

3.3. Convexified Tightened Constraint

We perform the convexification of the probabilistic collision avoidance constraint in 2 steps: first, we seek a conservative approximation of the probability contour in the position space of ego_i for the threshold δ with each ov_j; second, we seek convex bounds on the mean position state that guarantee the threshold δ is not violated.

3.3.1. Tightened constraint bounding box

For a given threshold δ, it remains difficult to invert the right-hand side in (12) to find the corresponding contour. We simplify this task by finding the length and width of a bounding box circumscribing the contour (see Figure 11) as tightly as possible, thereby remaining conservative but not too much so. To do this, we implement a search along axes aligned with those used in the decoupling approximations.

Now, to find the tightened constraint bounding box we perform five steps:

1. Discretize the relative heading space into n_ϕ intervals $\Phi_{ji,l}$, where $l \in \{1, \ldots, n_\phi\}$, and calculate $\Pr(\phi_{ji} \in \Phi_{ji,l})$, a_l'', and b_l'' for all l.
2. Calculate $\Pr\left(x_1' \in \left[\underline{x}_{1,l}', \overline{x}_{1,l}'\right]\right)$ assuming $\mu_{x_1'} = 0$ for all $l \in \{1, \ldots, n_\phi\}$.

3. Implement a gradient-based probability threshold search (PTS) algorithm presented in Algorithm 1 in Appendix D, to find $\mu^*_{x'_2}$, the extent along axis 2 where $\Pr\left(x'_{ij} \in \mathbb{X}''_{ij}\left(0\right)\right) = \delta$.

4. Repeat steps 2 and 3 by switching the coordinates (i.e. $\mu_{x'_2} = 0$, and search for the corresponding $\mu^*_{x'_1}$ where $\Pr\left(x'_{ij} \in \mathbb{X}''_{ij}\left(0\right)\right) = \delta$).

The resulting tightened constraint bounding box in the transformed coordinate frame is of size $2\mu^*_{x'_1}$ by $2\mu^*_{x'_2}$. We note here that steps 2 and 3 in the above iterative algorithm are independent. This provides the opportunity to use different decoupling methods to find $\mu^*_{x'_1}$ and $\mu^*_{x'_2}$. Therefore, to reduce conservatism it is ideal to use US case 1 to find $\mu^*_{x'_2}$ and US case 2 to find $\mu^*_{x'_1}$. To obtain constraints expressed in the original reference frame the inverse transformations for the respective methods may be applied to the points defining the bounding box in the transformed coordinates.

3.3.2. Convex bounds

When directly constraining the mean position state of ego_i to be outside the tightened constraint bounding box found above (Figure 11 for a given threshold δ), a disjunctive or mixed-integer program results requiring specialized solvers. In our prior work, to avoid posing the problem in this way, we utilized elliptical or hyper-elliptical collision avoidance constraints, as they are continuous and differentiable and can be readily handled via nonlinear optimization solvers. There are, however, two drawbacks to that method: 1) the feasible set for the position states $\begin{bmatrix} s & y_e \end{bmatrix}^T$ are non-convex; and, 2) the major and minor axes of a hyper-ellipse (fourth order) must be enlarged by a factor of at least $2^{1/4}$, in order to circumscribe the minimum bounding box [38].

Here, we choose a third approach: at each time step in the horizon, the lateral deviation y_e is upper and lower constrained by linear functions of s, as depicted in Figure 12. More specifically, the upper and lower constraints on y_e are defined as:

$$\underline{\alpha}s + \underline{\beta} \leq y_e \leq \overline{\alpha}s + \overline{\beta}, \tag{16}$$

where $\overline{\alpha}$ and $\underline{\alpha}$ are the slopes of the upper and lower bounds, respectively, on y_e at time step k, while $\overline{\beta}$ and $\underline{\beta}$ are the respective intercepts. We further constrain the longitudinal position s to be within an upper bound $\overline{s}$ and lower bound $\underline{s}$ to guarantee the constraint (16) performs as intended.
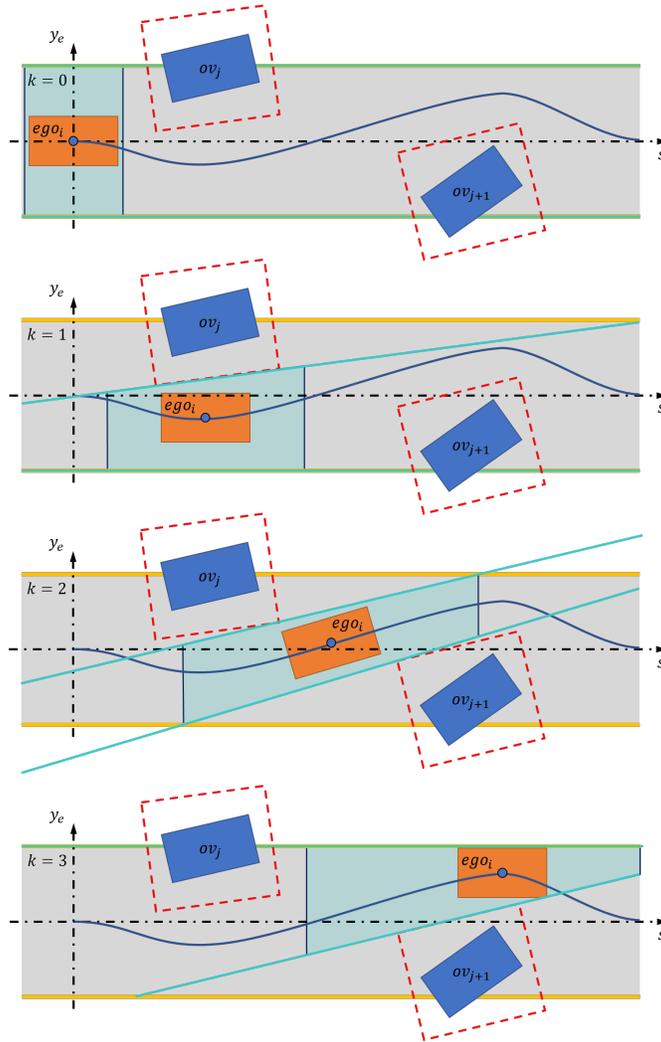
Figure 12: Example evolution of the convex constraint space (shaded teal) over the prediction horizon. The blue spline is the predicted trajectory, red dashed lines are the bounding boxes, yellow lines are the road boundary, and the teal lines are the linear y_e bounds.

In order to implement this method for deterministic constraint tightening we make two assumptions: 1) the solution to the MPC problem does not deviate significantly from one solver call to another, and 2) if the prior plan is shifted assuming a constant velocity between prediction time steps, as presented in [39], the error in the position states is negligible. The first assumption allows us to use the prior solution of the MPC problem to define

the slope and intercept of the boundary line at each time step in the prediction horizon. Whereas the second assumption allows for an efficient method of shifting the plan if the time step in the horizon is larger than the update rate of the MPC.

Prior to defining the slopes and intercepts, we will introduce some useful notations. Let $\mathbb{B}_j$ denote the ordered set of four (s, y_e) points that define the tightened constraint bounding box of ov_j. Moving forward we will drop the OV index j to reduce clutter. Each point in $\mathbb{B}$ is designated by b_p, $p \in \{1, \ldots, 4\}$. An edge of the tightened constraint bounding box is defined by two adjacent points in $\mathbb{B}$ and designated by $B_p = \{b_p, b_q\}$, where $q = p + 1$ if $p < 4$ and $q = 1$ if $p = 4$.

To find $\overline{\alpha}$, $\underline{\alpha}$, $\overline{\beta}$, and $\underline{\beta}$ we begin by sorting whether a given ov_j should be considered for the upper or lower bound. For the purposes of this paper, if the geometric center of ov_j is above the lane centerline ($y_{e,j} > 0$), then it is considered in the upper bound on y_e. While, if $y_{e,j}$ is below the lane centerline $y_{e,j} \leq 0$, it is considered in the lower bound on y_e. Once the OVs are sorted, a simple set of rules is used to calculate $\overline{\alpha}$, $\underline{\alpha}$, $\overline{\beta}$, and $\underline{\beta}$. There are three possible conditions based on s_i; we will refer to Figure 12 to describe each condition. We will use the subscript s (or y_e) when referring to those respective values of elements in $\mathbb{B}$. Next we will overview how to calculate $\overline{\alpha}$ and $\overline{\beta}$. An analogous method may be used to calculate $\underline{\alpha}$ and $\underline{\beta}$, however, we will highlight minor changes as necessary.

The first condition arises if $\min \mathbb{B}_s \leq s_i \leq \max \mathbb{B}_s$, as is the case at $k = 1$ for the upper y_e bound in Figure 12 (or $k = 3$ for the lower). To calculate $\overline{\alpha}$ and $\overline{\beta}$, we begin by determining the set $\{B_p\}$ of edges where $s_i \in [\min B_{p,s}, \max B_{p,s}]$. Then $\overline{\alpha}$ and $\overline{\beta}$ are the slope and intercept, respectively, of the edge B_p in that set satisfying $\arg\min_p \{B_{p,y_e}\}$ (or $\arg\max_p \{B_{p,y_e}\}$ for $\underline{\alpha}$ and $\underline{\beta}$).

The second possible condition occurs if ego_i is within Δs, a fixed look-ahead/behind distance, of the tightened constraint bounding box. Specifically, if $s_i \geq \max \mathbb{B}_s$ and $s_i - \Delta s \leq \max \mathbb{B}_s$ or $s_i \leq \min \mathbb{B}_s$ and $s_i + \Delta s \geq \min \mathbb{B}_s$ (both depicted at $k = 2$ in Figure 12, the former as the upper bound and the latter as the lower). We then calculate $\overline{\alpha}$ and $\overline{\beta}$ as follows:

$$\overline{\alpha} = \frac{\overline{y}_e - b_{p,y_e}}{s_i \mp \Delta s - b_{p,s}} \tag{17a}$$

$$\overline{\beta} = \overline{y}_e - \overline{\alpha}\left(s_i \mp \Delta s\right), \tag{17b}$$

25

where $\overline{y}_e$ is the upper roadway boundary. For $\underline{\alpha}$ and $\underline{\beta}$ replace $\overline{y}_e$ in (17) with the lower roadway boundary $\underline{y}_e$.

The last possible condition occurs when the tightened constraint bounding box is beyond the look ahead/behind distance, i.e. when $s_i + \Delta s < \min \mathbb{B}_s$ or $s_i - \Delta s > \max \mathbb{B}_s$. When this occurs, the constraint is defined by the roadway boundaries, therefore $\overline{\alpha} = 0$ and $\overline{\beta} = \overline{y}_e$ ($\underline{\beta} = \underline{y}_e$). This condition is depicted at time steps $k = 0$ or 3 for the upper bound in Figure 12 ($k = 0$ and 1 for the lower).

We note that the above algorithm for determining a convex constraint space is relatively simple, and the problem of decomposing the obstacle-free space into convex sets is an intriguing research question on its own with many solutions (e.g. [40, 41]). However, our primary focus within this paper is on guaranteeing the probability of collision threshold is not violated (accomplished in Section 3.3.1) and on the added performance benefits of the convexified tightened constraint.

3.3.3. Results

In this section, we compare the performance of our CC-MPC with the convexified tightened collision avoidance constraint to that with the direct implementation of the approximated constraint. To this end, first the same simulation setup as presented in Section 3.2 was used. As before, 100 simulations for each of the PA and US decoupling methods were completed. Here, when utilizing US decoupling, we combine US case 1 to find $\mu^*_{x'_2}$ and US case 2 to find $\mu^*_{x'_2}$, as noted in Section 3.3.1. Figure 13 shows the trajectories for the 100 simulations for both PA and US decoupling. When comparing this to Figure 9, we see that the convexified tightened constraint has significantly improved the robustness of our CC-MPC. Specifically, the CC-MPC was able to find a solution for all simulations. The ego_i also passes each ov_j on the same side in every simulation, which is required based on the formulation of the convexified tightened constraint.

The trajectories in Figure 13 show improved robustness, however, we are also interested in whether the probability of collision threshold is not violated. The boxplot of the maximum observed probability of collision between each OV is presented in Figure 14. From the figure, we notice that the probability of collision threshold is never violated. The maximum observed probability of collision was approximately 0.0008, which occurred between the ego_i and ov_1 with the US decoupling method. When comparing the two decoupling
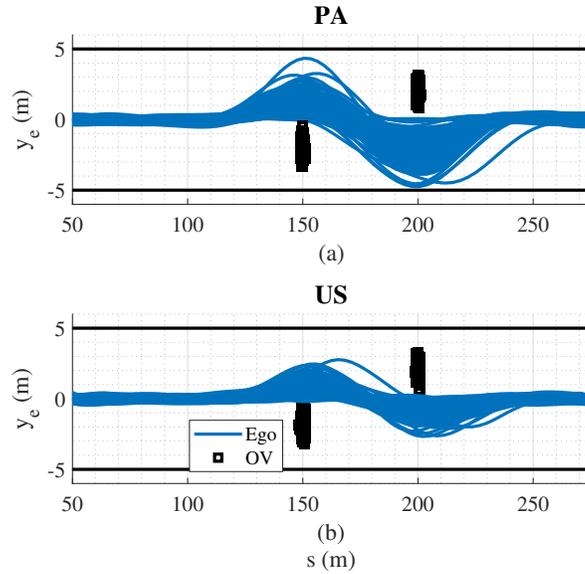
Figure 13: Simulation trajectories for CC-MPC with convexified tightened collision constraints using (a) PA decoupling and (b) US decoupling.
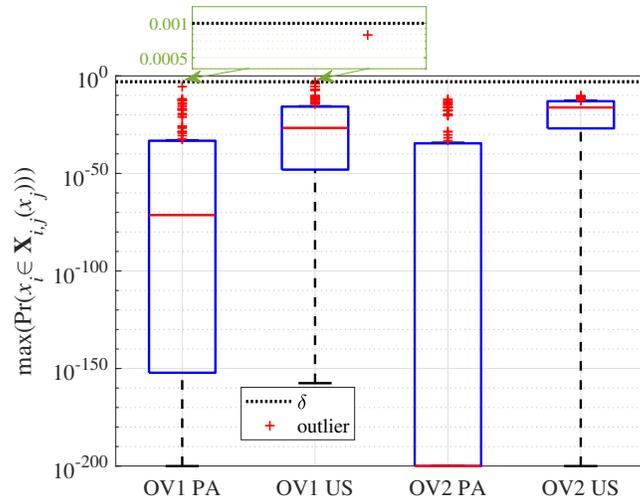

Figure 14: Box plot of max observed probability of collision between the ego_i and ov_1 and ov_2 for each simulation with the convexified tightened collision constraint.

methods, it is easy to see in Figure 14 that, as expected, the US decoupling method is significantly less conservative than the PA decoupling method.
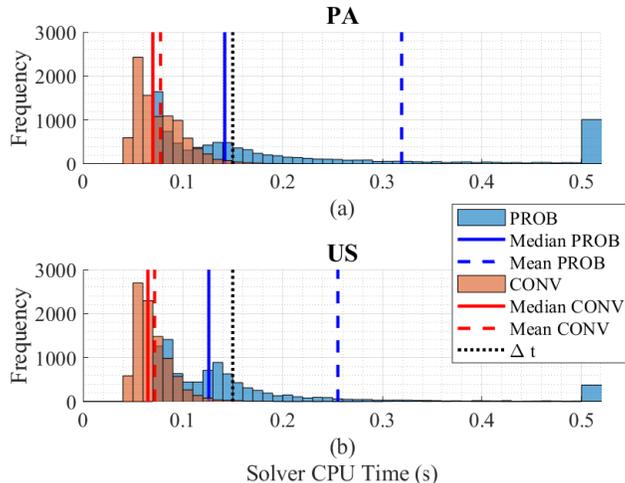
Figure 15: Histogram of MPC computation time for (a) PA and (b) US decoupling; where Δt is the simulation update rate, PROB denotes the direct implementation, and CONV denotes the convexified tightened constraint.

This is reflected in Figure 13, where the US trajectories deviate less from the centerline, traveling closer to the obstacles. Now comparing Figure 14 with Figure 10, it is evident that the overall probability of collision distributions is shifted downward for the convexified constraint, i.e. the constraint convexification has resulted in safer driving on average.

Next, it is important to determine if the reduction in conservatism comes at the cost of increased computation time. Figure 15 presents the distributions of the computation times for both CC-MPC implementations and both decoupling methods. The distribution is significantly shifted to the left (or computation times are on average lower) for the convexified constraint. Specifically, using the convexified constraint, the median computation times are reduced to 0.07s for the PA decoupling case and 0.065s for the US decoupling case. With the convexified constraint, the real-time requirement (Δ_t) is exceeded only in 1.85% of the MPC calls for the PA case and in 1.31% of the MPC calls for the US case.

The improved computation time does come at the cost of increased precomputations for finding the dimensions of the tightened constraint bounding box. The computation statistics for finding the bounding box dimensions are presented in Table 1. There is a negligible difference in the computation times for the two decoupling methods. Further, the magnitude of the addi-

Table 1: Computation Time Statistics for Finding the Tightened Constraint Bounding Box Dimensions

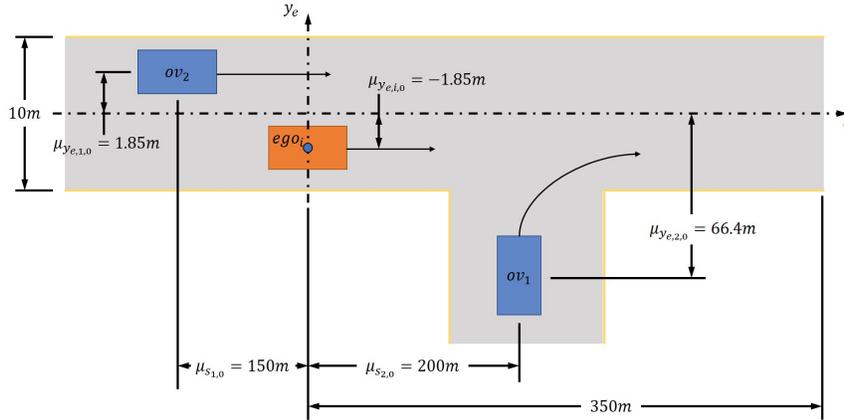| Computation Time (ms)	PA	US
Median	3.999	4.035
Upper Quartile	4.670	4.692

Figure 16: Scenario to illustrate performance with moving obstacles. The OV references are $r_{v,1} = 8$ m/s and $r_{v,2} = 22$ m/s for speed and $r_{y_e,1} = -1.85$ m and $r_{y_e,2} = 1.85$ m for lateral error, respectively, while the ego references are $r_{v,i} = 20$ m/s and $r_{y_e,i} = -1.85$ m.

tional precomputation time is significantly lower than the reduction in the CC-MPC execution time. We do note that the presented computation time is only for finding one set of tightened constraint bounding box dimensions. This process must be repeated for each step of the horizon, however, each computation is independent and can therefore be parallelized. Thus, the con-vexified tightened constraint is a viable solution to improving the robustness and reducing the execution time of the presented CC-MPC.

Lastly, in order to show the applicability of the constraint handling method to more complex scenarios, we will briefly present results for a second scenario depicted in Figure 16. Therein, the ego vehicle is traveling in the right lane, with ov_2 passing ego_i on the left, while ov_1 is entering the roadway from the right in front of ego_i. The mean initial positions (of the randomize simulations) are shown in Figure 16, while the initial velocities are as follows: $v_{i,0} = 17$ m/s, $v_{1,0} = 15$ m/s, and $v_{2,0} = 20$ m/s. The OVs utilize the intelligent driver model (IDM) for speed tracking and a linear feedback controller
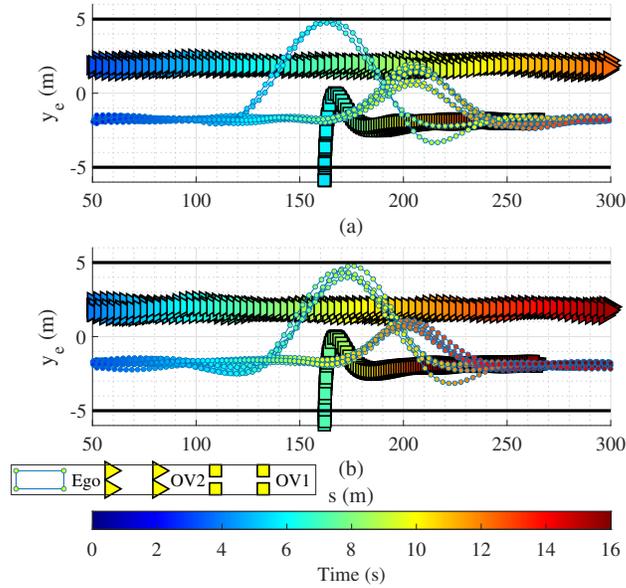
Figure 17: Simulation trajectories for CC-MPC with convexified tightened collision constraints and scenario 2 using (a) PA and (b) US decoupling.

for y_e and ψ (the reference heading error $r_\psi = 0$ rad).

Figure 17 shows select simulated trajectories with the CC-MPC and the two decoupling methods. For clarity only representative trajectories are shown. In general, there are two main types of responses that ego_i exhibits, either: speeding up and passing ov_1 before returning to the left and allowing ov_2 to pass; or slowing down and allowing ov_2 to pass prior to passing ov_1. Despite the varied responses, the probability of collision criteria are still met as can be seen in Figure 18. When comparing to Figure 14, it is evident that with moving obstacles the maximum probability of collision distributions are shifted upwards as the dynamic environment includes additional uncertainties with respect to OV state predictions that are not captured with the simplified models used here. Still, the thresholds were not violated in any of the simulations.

4. Conclusion

In this paper we have derived analytical methods for the efficient approximation of an upper bound on the mutual probability of collision between
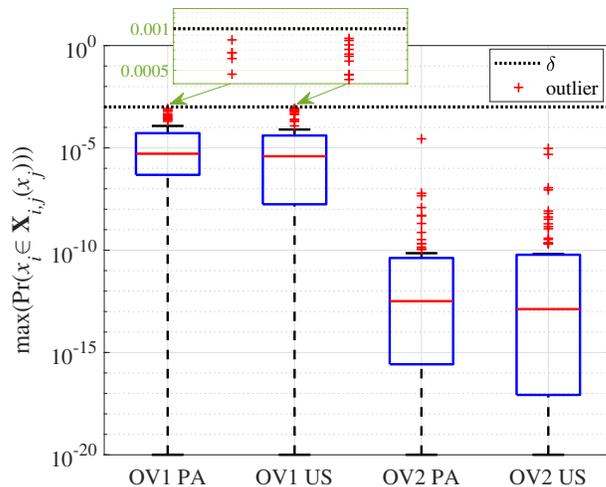
30

Figure 18: Box plot of max observed probability of collision between the ego_i and ov_1 and ov_2 for scenario 2 with the convexified tightened collision constraint.

robotic vehicles. This included a detailed discussion of two separate decoupling methods (principal axes rotation, and unitary scaling cases 1 and 2) and the associated conservatism of each method. We have drawn the following important insights on the utility of either method: principal axis rotation is ideal for unstructured environments as the conservatism introduced is easily bounded in all directions; unitary scaling case 1 is ideal for environments where agents are traveling in parallel and passing each other, because the conservatism is only introduced in the longitudinal direction; lastly, unitary scaling case 2 is ideal for environments where agents are following each other as conservatism is only introduced in the lateral direction.

The effectiveness of the analytical approximation methods was then evaluated within a chance-constrained model predictive controller (CC-MPC) motion planner formulation. Therein, we highlighted shortcomings in the robustness and computational efficiency of using a direct implementation of the tightened probability of collision constraints in the CC-MPC. We then proposed a method for convexification of the tightened constraints which showed promising results for significantly improving robustness and reducing computation times for online motion planning in complex traffic.

There are important future extensions that can be pursued on the application of these approximation methods for mutual collision probabilities. These include the following: Verifying the performance when implemented

on a higher fidelity plant/vehicle model; Using the approximations for clustering object vehicles in traffic to enable further convexification of the navigation space [21]; Improving the algorithm presented here for determining the slope and intercept of the convexified tightened constraint; Exploring methods/extensions to the constraint building process to consider unstructured environments and roadways with multiple lanes; Investigating methods to account for the inherent coupling between neighboring vehicle's states with an additional layer of uncertainty decoupling.

Appendix A. Calculating $\Sigma_{x'_{ij}}$ and θ

In order to implement principal axis rotation it is necessary to solve for the covariance $\Sigma_{x'_{ij}}$ along the principal axes and the angle of rotation θ to align the reference frame with the principal axes. To accomplish this, we first solve for the eigenvalues of the original covariance $\Sigma_{x_{ij}}$, as $\Sigma_{x'_{ij}} = D$, where D is the diagonal matrix of eigenvalues. Assuming the covariance matrix is of the form:

$$\Sigma_{x_{ij}} = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}, \tag{A.1}$$

we can compute the eigenvalues λ_c, $c \in \{1, 2\}$, by solving $\left|\Sigma_{x_{ij}} - \lambda_c I\right| = 0$, resulting in:

$$\lambda_{1/2} = \frac{\left(\sigma_x^2 + \sigma_y^2\right) \mp \sqrt{\left(\sigma_x^2 + \sigma_y^2\right)^2 - 4\sigma_x^2\sigma_y^2\left(1 - \rho^2\right)}}{2}. \tag{A.2}$$

Then, the diagonal matrix $D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$.

Now that we have the covariance along the principal axes, it is necessary to solve for the rotation angle θ. We first solve for the matrix of eigenvectors $V = \begin{bmatrix} \vec{v}_1 & \vec{v}_2 \end{bmatrix}$, where $\vec{v}_c = \begin{bmatrix} v_{c1} & v_{c2} \end{bmatrix}^T$, for all $c \in \{1, 2\}$, is the unit eigenvector corresponding to the cth eigenvalue. Knowing that $\left(\Sigma_{x_{ij}} - \lambda_c I\right)\vec{v}_c = 0$ and $v_{c1}^2 + v_{c2}^2 = 1$, as we assume unit eigenvectors, we can solve for v_{c1} and v_{c2} as follows:

$$v_{c2} = \pm\left[\left(\frac{\sigma_x\sigma_y\rho}{\sigma_x^2 - \lambda_c}\right)^2 + 1\right]^{-1/2}; \tag{A.3a}$$

$$v_{c1} = -\frac{\sigma_x\sigma_y\rho v_{c2}}{\sigma_x^2 - \lambda_c}. \tag{A.3b}$$

Further, as the eigenvectors are orthogonal, it can be shown that $v_{21} = \mp v_{12}$ and $v_{22} = \pm v_{11}$. The angle of rotation θ_c is then:

$$\theta_c = -\tan^{-1}\frac{v_{c2}}{v_{c1}}. \tag{A.4}$$

Substituting (A.3b) in to (A.4) and simplifying yields:

$$\theta_c = -\tan^{-1}\frac{\lambda_c - \sigma_x^2}{\sigma_x\sigma_y\rho}. \tag{A.5}$$

For our implementation we rotate to align with the first principal axis or θ_1. The rotation applied to x_{ij} and $\mathbb{X}_{ij}\left(0, \Phi_{ji,l}\right)$ is then:

$$T_c = \begin{bmatrix} \cos\theta_c & -\sin\theta_c \\ \sin\theta_c & \cos\theta_c \end{bmatrix}, \tag{A.6}$$

where:

$$\cos\theta_c = \frac{\sigma_x\sigma_y\rho}{\sqrt{\left(\lambda_c - \sigma_x^2\right)^2 + \left(\sigma_x\sigma_y\rho\right)^2}}; \tag{A.7a}$$

$$\sin\theta_c = \frac{\lambda_c - \sigma_x^2}{\sqrt{\left(\lambda_c - \sigma_x^2\right)^2 + \left(\sigma_x\sigma_y\rho\right)^2}}. \tag{A.7b}$$

Lastly, as we perform the inverse rotation in order to bring the tightened constraint bounding box back to the original reference frame for use within our motion planning algorithm, we note:

$$T_c^{-1} = \begin{bmatrix} \cos-\theta_c & -\sin-\theta_c \\ \sin-\theta_c & \cos-\theta_c \end{bmatrix}. \tag{A.8}$$

Appendix B. Unitray Scaling - Calculating T_0

In order to efficiently compute the transformation matrix $T_0 = \Sigma_{x_{ij}}^{-1/2}$, we can take advantage of the fact that the covariance matrix $\Sigma_{x_{ij}}$ is symmetric. Due to symmetry, it is possible to orthogonally diagonalize the covariance $\Sigma_{x_{ij}} = VDV^{-1} = VDV^T$ [42]. See Appendix A for how to compute D and V. The square root by diagonalization is then simply $\Sigma_{x_{ij}}^{1/2} = VD^{1/2}V^T$,

where $D^{1/2} = \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{bmatrix}$. To obtain $T_0 = \left(VD^{1/2}V^T \right)^{-1}$, we take the inverse:

$$T_0 = \frac{1}{\left(v_{11}v_{22} - v_{21}v_{12} \right)^2 \lambda_1^{\frac{1}{2}} \lambda_2^{\frac{1}{2}}} A, \tag{B.1}$$

where:

$A =$

$$\begin{bmatrix} v_{12}^2 \lambda_1^{\frac{1}{2}} + v_{22}^2 \lambda_2^{\frac{1}{2}} & -v_{11}v_{12}\lambda_1^{\frac{1}{2}} - v_{21}v_{22}\lambda_2^{\frac{1}{2}} \\ -v_{11}v_{12}\lambda_1^{\frac{1}{2}} - v_{21}v_{22}\lambda_2^{\frac{1}{2}} & v_{11}^2 \lambda_1^{\frac{1}{2}} + v_{21}^2 \lambda_2^{\frac{1}{2}} \end{bmatrix}. \tag{B.2}$$

Appendix C. Unitary Scaling - Decomposing T_0

As mentioned previously, the transformation T_0 can be decomposed into three elementary operations: rotation, scaling, and shear. From a matrix perspective this means T_0 may be decomposed into $T_0 = RSH$, where the rotation matrix R, scaling matrix S, and shear matrix H are defined as:

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}; \tag{C.1a}$$

$$S = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}; \tag{C.1b}$$

$$H = \begin{cases} \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} & \text{for case 1,} \\ \begin{bmatrix} 1 & 0 \\ h & 1 \end{bmatrix} & \text{for case 2.} \end{cases} \tag{C.1c}$$

Multiplying R, S, and H, and setting the elements of RSH equal to those of $T_0 = \begin{bmatrix} t_{11} & t_{12} \\ t_{12} & t_{22} \end{bmatrix}$ yields a system of 4 equations. We first detail the solution for case 1, followed by that for case 2.

Appendix C.1. Case 1 - Longitudinal Conservatism

The system of equations for case 1 is:

$$t_{11} = s_1 \cos \alpha; \tag{C.2a}$$

$$t_{12} = s_1 h \cos \alpha - s_2 \sin \alpha; \tag{C.2b}$$

$$t_{12} = s_1 \sin \alpha; \tag{C.2c}$$

$$t_{22} = s_1 h \sin \alpha + s_2 \cos \alpha, \tag{C.2d}$$

with 4 unknowns, α, s_1, s_2, and h. Solving this yields:

$$\alpha = \tan^{-1} \frac{t_{12}}{t_{11}}; \tag{C.3a}$$

$$s_1 = \sqrt{t_{11}^2 + t_{12}^2}; \tag{C.3b}$$

$$s_2 = \frac{t_{11} t_{22} - t_{12}^2}{\sqrt{t_{11}^2 + t_{12}^2}}; \tag{C.3c}$$

$$h = \frac{t_{12} (t_{22} + t_{11})}{t_{12}^2 + t_{11}^2}. \tag{C.3d}$$

Recalling we seek the transformation matrix $T = SH$, it is beneficial to obtain (C.3b)-(C.3d) in terms of the eigenvalues and eigenvector elements. Substituting the definitions of t_{11}, t_{12}, and t_{22} from (B.1) and (B.2) and simplifying yields:

$$s_1 = \left(\frac{v_{12}^2 \lambda_1 + v_{11}^2 \lambda_2}{\lambda_1 \lambda_2} \right)^{\frac{1}{2}}; \tag{C.4a}$$

$$s_2 = \left(v_{12}^2 \lambda_1 + v_{11}^2 \lambda_2 \right)^{\frac{1}{2}}; \tag{C.4b}$$

$$h = \frac{v_{11} v_{12} (\lambda_2 - \lambda_1)}{v_{12}^2 \lambda_1 + v_{11}^2 \lambda_2}. \tag{C.4c}$$

Now to see if this equation is unbounded we will look at the limits of (C.4c) with respect to the eigenvectors and eigenvalues. We know that the eigenvector elements are bounded by $[-1, 1]$ and $v_{11}^2 + v_{12}^2 = 1$, as we assumed unit eigenvectors. It is easy to see that if $v_{11} = \pm 1$ and $v_{12} = 0$, or vice versa, $h = 0$. Therefore, we are interested in the limits with respect to the eigenvalues. Taking the limit of (C.4c) as $\lambda_2 \to \pm\infty$ and using L'Hôpital's rule yields v_{12}/v_{11}, which is the tangent of the eigenvector and therefore unbounded. Similarly, taking the limit as $\lambda_1 \to \pm\infty$ yields $-v_{11}/v_{12}$, which is

the negative of the cotangent of the eigenvector and also unbounded. Although, the shear is theoretically unbounded, from a practical perspective, if you can characterize and bound the spectral radius of $\Sigma_{x_{ij}}$, the shear, and conservatism is therefore bounded.

Appendix C.2. Case 2 - Lateral Conservatism

The system of equations for case 2 is:

$$t_{11} = s_1 \cos \alpha - s_2 h \sin \alpha; \tag{C.5a}$$
$$t_{12} = -s_2 \sin \alpha; \tag{C.5b}$$
$$t_{12} = s_1 \sin \alpha + s_2 h \cos \alpha; \tag{C.5c}$$
$$t_{22} = s_2 \cos \alpha. \tag{C.5d}$$

Solving the systems results in:

$$\alpha = \tan^{-1} -\frac{t_{12}}{t_{22}}; \tag{C.6a}$$

$$s_1 = \frac{t_{11}t_{22} - t_{12}^2}{\sqrt{t_{12}^2 + t_{22}^2}}; \tag{C.6b}$$

$$s_2 = \sqrt{t_{12}^2 + t_{22}^2}; \tag{C.6c}$$

$$h = \frac{t_{12}\left(t_{22} + t_{11}\right)}{t_{12}^2 + t_{22}^2}. \tag{C.6d}$$

As we did with case 1, we substitute in to (C.5b)-(C.5d) from (B.1) and (B.2) and simplify to obtain:

$$s_1 = \left(v_{11}^2 \lambda_1 + v_{12}^2 \lambda_2\right)^{\frac{1}{2}}; \tag{C.7a}$$

$$s_2 = \left(\frac{v_{11}^2 \lambda_1 + v_{12}^2 \lambda_2}{\lambda_1 \lambda_2}\right)^{\frac{1}{2}}; \tag{C.7b}$$

$$h = \frac{v_{11}v_{12}\left(\lambda_2 - \lambda_1\right)}{v_{11}^2 \lambda_1 + v_{12}^2 \lambda_2}. \tag{C.7c}$$

Comparing (C.4c) of case 1 and (C.7c) of case 2, we can see the only difference is the eigenvalues are weighted by the opposite elements of the eigenvector in the denominator. The results for analyzing the limits with respect to v_{11} and v_{12}, therefore, still hold. However, the limit of h as $\lambda_1 \rightarrow \pm\infty$ is now $-v_{12}/v_{11}$, and the limit of h as $\lambda_2 \rightarrow \pm\infty$ is v_{11}/v_{12}. Again, this is unbounded.

Appendix D. Probability Threshold Search Algorithm

The probability threshold search (PTS) algorithm is used within step 3 of the process in Section 3.3.1 for finding the tightened constraint minimum bounding box. PTS (Algorithm 1) is based on Newton's method.

Algorithm 1 Probability Threshold Search (PTS) Algorithm

Input: Threshold δ, maximum iterations $\overline{N}_i$, tolerance ε, initial guess $\mu_{x'_{2,0}}$,
$\left\{ \Pr\left(\phi_{ji} \in \Phi_{ji,l} \right), a''_l, b''_l \right\} \forall l \in \{1, ..., n_\phi\}$, $\Sigma_{x'_{ij}}$, and $\Pr\left(x'_1 \in \left[\underline{x}'_{1,l}, \overline{x}'_{1,l} \right] \right)$

Output: Distance $\mu^*_{x'_2}$, Probability $\Pr\left(x'_{ij} \in \mathbb{X}''_{ij}(0) \right)$, Iteration number i

1: **Initialize** $\mu^*_{x'_2} = \mu_{x'_{2,0}}$, $i = 1$, $\underline{\mu}_{x'_2} = 0$, $\overline{\mu}_{x'_2} = 1e5$

2: **while** $i \le \overline{N}_l$ **do**

3:     Calculate $\Pr\left(x'_{ij} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \mu^*_{x'_2} \end{bmatrix}, \Sigma_{x'_{ij}} \right) \in \mathbb{X}''_{ij}(0) \right)$

4:     Calculate deviation to δ: $dP = \Pr\left(x'_{ij} \in \mathbb{X}''_{ij}(0) \right) - \delta$

5:     **if** $|dP| \le \varepsilon$ **then**

6:         **break while**

7:     **end if**

8:     **if** $dP < 0$ and $\mu^*_{x'_2} < \overline{\mu}_{x'_2}$ **then**

9:         $\overline{\mu}_{x'_2} = \mu^*_{x'_2}$

10:    **end if**

11:    **if** $dP > 0$ and $\mu^*_{x'_2} > \underline{\mu}_{x'_2}$ **then**

12:        $\underline{\mu}_{x'_2} = \mu^*_{x'_2}$

13:    **end if**

14:    Calculate the derivative $\frac{\partial}{\partial \mu^*_{x'_2}} \Pr\left(x'_{ij} \in \mathbb{X}''_{ij}(0) \right)$

15:    **if** $\mu^*_{x'_2} - dP / \frac{\partial}{\partial \mu^*_{x'_2}} \Pr\left(x'_{ij} \in \mathbb{X}''_{ij}(0) \right) > \overline{\mu}_{x'_2}$ **then**

16:        $\mu^*_{x'_2} = \left(\overline{\mu}_{x'_2} + \underline{\mu}_{x'_2} \right)/2$

17:    **else if** $\mu^*_{x'_2} - dP / \frac{\partial}{\partial \mu^*_{x'_2}} \Pr\left(x'_{ij} \in \mathbb{X}''_{ij}(0) \right) < \underline{\mu}_{x'_2}$ **then**

18:        $\mu^*_{x'_2} = \left(\overline{\mu}_{x'_2} + \underline{\mu}_{x'_2} \right)/2$

19:    **else**

20:        $\mu^*_{x'_2} = \mu^*_{x'_2} - dP / \frac{\partial}{\partial \mu^*_{x'_2}} \Pr\left(x'_{ij} \in \mathbb{X}''_{ij}(0) \right)$

21:    **end if**

22:    $i = i + 1$

23: **end while**

References

[1] S. Lefèvre, D. Vasquez, C. Laugier, A survey on motion prediction and risk assessment for intelligent vehicles, ROBOMECH Journal 1 (1) (2014) 1–14.

[2] S. Wender, K. Dietmayer, 3D vehicle detection using a laser scanner and a video camera, IET Intell. Transport Syst. 2 (2) (2008) 105–1112.

[3] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang, S. Godsill, Overview of Bayesian sequential monte carlo methods for group and extended object tracking, Digital Signal Processing 25 (2014) 1–16.

[4] K. Granström, M. Baum, S. Reuter, Extended object tracking: introduction, overview and applications, Journal of Advances in Information Fusion 12 (2) (2016) 139–174.

[5] C. Goerzen, Z. Kong, B. Mettler, A survey of motion planning algorithms from the perspective of autonomous UAV guidance, Journal of Intell. and Robotic Syst. 57 (1-4) (2010) 65.

[6] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, E. Frazzoli, A survey of motion planning and control techniques for self-driving urban vehicles, IEEE Trans. on Intell. Vehicles 1 (1) (2016) 33–55.

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

[8] Y. Kuwata, T. Schouwenaars, A. Richards, J. How, Robust constrained receding horizon control for trajectory planning, in: AIAA Guidance, Navigation, and Control Conf. and Exhibit, 2005.

[9] L. Blackmore, H. Li, B. Williams, A probabilistic approach to optimal robust path planning with obstacles, in: 2006 American Control Conference, 2006, pp. 7 pp.–.

[10] Q. Wang, B. Ayalew, Constraint tightening for the probabilistic collision avoidance of multi-vehicle groups in uncertain traffic, in: 2017 IEEE Conf. on Control Technol. and Appl. (CCTA), 2017, pp. 2188–2195.

[11] A. Carvalho, Y. Gao, S. Lefevre, F. Borrelli, Stochastic Predictive Control of Autonomous Vehicles in Uncertain Environments, International Symposium on Advanced Vehicle Control (AVEC) (June) (2014).

[12] N. E. Du Toit, J. W. Burdick, Probabilistic collision checking with chance constraints, IEEE Trans. on Robotics 27 (4) (2011) 809–815.

[13] J. Hardy, M. Campbell, Contingency planning over probabilistic obstacle predictions for autonomous road vehicles, IEEE Trans. on Robotics 29 (4) (2013) 913–929. doi:10.1109/TRO.2013.2254033.

[14] A. Kushleyev, M. Likhachev, Time-bounded lattice for efficient planning in dynamic environments, in: 2009 IEEE International Conf. on Robotics and Automation, 2009, pp. 1662–1668.

[15] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, J. P. How, Real-time motion planning with applications to autonomous urban driving, IEEE Trans. on Control Syst. Technol. 17 (5) (2009) 1105–1118.

[16] J. Nilsson, P. Falcone, M. Ali, J. Sjöberg, Receding horizon maneuver generation for automated highway driving, Control Engineering Practice 41 (2015) 124–133.

[17] U. Rosolia, F. Braghin, A. Alleyne, E. Sabbioni, NLMPC for real time path following and collision avoidance, SAE International Journal of Passenger Cars-Electronic and Electrical Syst. 8 (2015-01-0313) (2015) 401–405.

[18] M. S. Menon, A. Ghosal, Obstacle avoidance for hyper-redundant snake robots and one dimensional flexible bodies using optimization, in: 14th world congress in mechanism and machine science, 2015.

[19] T. Weiskircher, Q. Wang, B. Ayalew, Predictive guidance and control framework for (semi-)autonomous vehicles in public traffic, IEEE Trans. on Control Syst. Technol. 25 (6) (2017) 2034–2046. doi:10.1109/TCST.2016.2642164.

[20] E. Balas, Disjunctive programming, in: P. Hammer, E. Johnson, B. Korte (Eds.), Discrete Optimization II, Vol. 5 of Annals of Discrete Mathematics, Elsevier, 1979, pp. 3–51.

[21] Q. Wang, B. Ayalew, A probabilistic framework for tracking the formation and evolution of multi-vehicle groups in public traffic in the presence of observation uncertainties, IEEE Trans. on Intell. Transp. Syst. 19 (2) (2018) 560–571. doi:10.1109/TITS.2017.2750073.

[22] A. Petrovskaya, S. Thrun, Model based vehicle detection and tracking for autonomous urban driving, Autonomous Robots 26 (2-3) (2009) 123–139.

[23] D. D. Yoon, G. G. M. N. Ali, B. Ayalew, Data Association and Fusion Framework for Decentralized Multi-Vehicle Cooperative Perception, in: Proc. of the ASME 2019 IDETC & CIE Conf., 2019.

[24] S. M. LaValle, The Configuration Space, in: Planning Algorithms, Cambridge University Press, 2006, Ch. 4, pp. 155–168.

[25] S. Sivaraman, M. M. Trivedi, Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis, IEEE Trans. on Intell. Transp. Syst. 14 (4) (2013) 1773–1795.

[26] M. Sefati, J. Chandiramani, K. Kreiskoether, A. Kampker, S. Baldi, Towards tactical behaviour planning under uncertainties for automated vehicles in urban scenarios, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017, pp. 1–7.

[27] M. Rodrigues, A. McGordon, G. Gest, J. Marco, Autonomous navigation in interaction-based environments—a case of non-signalized roundabouts, IEEE Transactions on Intelligent Vehicles 3 (4) (2018) 425–438.

[28] B. Zhou, W. Schwarting, D. Rus, J. Alonso-Mora, Joint multi-policy behavior estimation and receding-horizon trajectory planning for automated urban driving, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 2388–2394.

[29] R. A. Paielli, H. Erzberger, Conflict probability estimation for free flight, Journal of Guidance, Control, and Dynamics 20 (3) (1997) 588–596.

[30] C. Constable, D. C. Agnew, Multivariate Distributions, in: Geographical Data Analysis: Statistics, 2008, Ch. 4.

[31] T. A. N. Heirung, J. A. Paulson, J. O'Leary, A. Mesbah, Stochastic model predictive control — how does it work?, Computers & Chemical Engineering 114 (2018) 158–170.

[32] M. Lorenzen, F. Dabbene, R. Tempo, F. Allgöwer, Constraint-Tightening and Stability in Stochastic Model Predictive Control, IEEE Trans. on Automatic Control 62 (7) (2017) 3165–3177.

[33] B. Kouvaritakis, M. Cannon, Model Predictive Control Classical, Robust and Stochastic, 1st Edition, Springer International Publishing, 2016.

[34] N. E. Du Toit, J. W. Burdick, Robot motion planning in dynamic, uncertain environments, IEEE Trans. on Robotics 28 (1) (2012) 101–115.

[35] M. Risbeck, J. Rawlings, MPCTools: Nonlinear model predictive control tools for CasADi (Octave interface) (2016).
URL https://bitbucket.org/rawlings-group/octave-mpctools

[36] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, M. Diehl, CasADi: software framework for nonlinear optimization and optimal control, Mathematical Programming Computation 11 (1) (2019) 1–36.

[37] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical Programming 106 (1) (2006) 25–57.

[38] Q. Wang, Predictive Maneuver Planning and Control of an Autonomous Vehicle in Multi-Vehicle Traffic with Observation Uncertainty, Doctor of philosophy, Clemson University (2017).
URL https://tigerprints.clemson.edu/all_dissertations/2044

[39] N. Goulet, B. Ayalew, Distributed maneuver planning with connected and automated vehicles for boosting traffic efficiency, IEEE Trans. on Intell. Transp. Syst. (2021) 1–15doi:10.1109/TITS.2021.3096878.

[40] R. A. Brooks, Solving the find-path problem by good representation of free space, IEEE Trans. on Syst., Man, and Cybernetics SMC-13 (2) (1983) 190–197.

[41] R. Deits, R. Tedrake, Computing Large Convex Regions of Obstacle-Free Space Through Semidefinite Programming, Springer International Publishing, Cham, 2015, pp. 109–124.

[42] D. C. Lay, S. R. Lay, J. J. McDonald, 7.1 Diagonalization of Symmetric Matrices, in: Linear Algebra and its Applications, 5th Edition, Pearson Education, Inc., College Park, MD, 2016, Ch. 7, pp. 463–470.