Probabilistic Weapon Engagement Zones for a Turn Constrained Pursuer

Grant Stagg* and Cameron K. Peterson†
Brigham Young University, Provo, Utah, 84602, USA

Isaac Weintraub‡
Air Force Research Laboratory, Wright-Patterson AFB, Ohio, 45433, USA

Curve–straight probabilistic engagement zones (CSPEZ) quantify spatial regions an evader should avoid to mitigate capture risk against a turn-rate limited pursuer performing a curve–straight path with uncertain parameters (position, heading, velocity, range, and maximum turn rate). This research presents strategies and algorithms for generating evader paths that minimize capture risk from a pursuer when such uncertainty exists. We begin by deriving an analytic solution for the deterministic curve–straight engagement zone (CSBEZ). We then extend this model into a probabilistic framework by conducting a sensitivity analysis using Monte Carlo sampling, linearization, quadratic approximation, and neural network regression to account for parameter uncertainty. The accuracy, memory usage, and computational cost of these methods are evaluated in simulation. Finally, we incorporate CSPEZ constraints into a path planning algorithm to generate safe trajectories that explicitly consider pursuer uncertainty.

I. Introduction

Navigating hostile or contested areas is a common requirement in many mission scenarios, particularly for unmanned aerial vehicle (UAV) reconnaissance. These vehicles must often operate in environments where detection or engagement by adversaries presents a significant risk to the platform. To ensure mission success, it is essential to generate safe and efficient trajectories [1, 2] that minimize the need for abrupt evasive actions during execution. A key challenge arises when knowledge of the adversarial environment, such as the unknown positions or capabilities of the threat, is unknown or incomplete. If not properly accounted for, this uncertainty can cause UAVs to underestimate or overestimate the risks associated with specific paths. Therefore, robust planning approaches must explicitly incorporate this uncertainty to generate viable and survivable routes.

In this work, we use engagement zones (EZ) to model adversarial threats. EZs are regions an evader must avoid to guarantee that there will be no engagement with a pursuer. Past work has defined EZs using basic cardioid shapes [3–5]. These works created path planning algorithms for an evader to avoid engagements [3], accounting for two EZs [4], and many EZs [5]. All these methods use the simple cardioid-shaped EZ that directly represents the capabilities of the pursuer. They also rely on knowing the pursuer's parameters exactly, not accounting for the uncertainty inherent in adversarial environments.

Unlike the simple cardioid-shaped EZs, the researchers in [6] created the concepts of basic engagement zones (BEZ) that are defined using the properties of differential games [7]. They provided a BEZ between a pursuer and target agent, where the goal of the pursuer is to capture the target. Assuming the pursuer could instantaneously change its heading (infinite turn rate), the BEZ model was leveraged to create a path plan, using optimal control theory, around a single BEZ. In a previous work [8], we extended their work to account for uncertainty in the evader and pursuer parameters called probabilistic engagement zones (PEZs). We used a linear approximation to propagate uncertainty through the BEZ equation, then developed a path planning algorithm that uses the linear PEZ approximation as a constraint. Our work used the same model as [6], assuming the pursuer had an infinite turn-rate. The infinite-turn rate model is an overly conservative representation of real-world pursuer capabilities.

In recent work, the BEZ was extended to use a Dubin's vehicle model [9] for the pursuer [10]. Two BEZ variants were introduced to better represent the pursuer's motion capabilities: the curve-only BEZ (CBEZ), in which the pursuer

*PhD Student, Electrical and Computer Engineering, Brigham Young University.
†Associate Professor, Electrical and Computer Engineering, Brigham Young University, and AIAA Senior Member.
‡Senior Engineer, Control Science Center, Air Force Research Laboratory, Wright-Patterson AFB OH 45433, AIAA Lifetime Associate Fellow

follows a single arc of constant curvature to intercept the evader, and the curve-straight BEZ (CSBEZ), in which the pursuer executes a minimum-radius turn (i.e., at maximum turn rate) followed by a straight segment to complete the interception. Their work provided an analytic solution for the CBEZ and a semi-analytic solution for the CSBEZ. Both formulations require precise knowledge of pursuer parameters, position, heading, maximum turn-rate, range, and speed. In this paper, we extend that work to account for uncertain pursuer parameters.

Parallel work in addressing path planning under uncertainty in adversarial settings has explored the use of stochastic differential games [11–13]. In particular, [12] applies path integral control, a sampling-based technique, to solve zero-sum stochastic differential games. This approach uses random sampling and requires substantial computational resources to operate in real time. The researchers in [13] use optimal control to solve the stochastic differential game for the turn-rate limited pursuer and evader. These methods are primarily reactive, offering strategies for agents after the engagement has begun. They use a single objective, avoid capture, and do not account for higher-level mission objectives. Our work provides constraints that can be used in conjunction with higher-level mission goals to ensure safety while accounting for uncertainty.

In this paper, we extend the prior research on CSBEZ to provide safer trajectory planning for UAVs operating in adversarial environments. Our contributions include: an analytic solution for that CSBEZ that was originally introduced in [10]; an extension of the CSBEZ to account for uncertainty; and a path optimization algorithm that accounts for this uncertainty when generating low-risk trajectories. Specifically, we model uncertainty in the pursuer's position, heading, turn rate, range, and velocity through the introduction of the curve-straight probabilistic engagement zone (CSPEZ). We propose four methods for solving the CSPEZ. Two are based on local approximations: linear CSPEZ (LCSPEZ) and quadratic CSPEZ (QSPEZ). One is an approximation approach used as a baseline comparative: Monte Carlo CSPEZ (MCSPEZ). And the last is neural network-based approximation (NNCSPEZ), which employs a multi-layer perceptron trained on simulated data. We provide a numerical comparison of all these and describe the trade-offs in using each method. Finally, we also implement a path optimization algorithm that incorporates the CSPEZ approximations as constraints, ensuring the planned trajectories remain safe under uncertainty in the pursuer parameters.

The remainder of the paper is organized as follows. Section II presents background information and defines the CSBEZ. In Section III, we extend this model to incorporate uncertainty in the pursuer's parameters, introducing the turn-straight probabilistic engagement zone (CSPEZ) and describing the four approximate solution methods. Section IV details path planning algorithms that leverage the CSPEZ formulations. Experimental results are presented in Section V, and conclusions are drawn in Section VI.

II. Curve-Straight Basic Engagement Zone (CSBEZ)

The curve straight basic engagement zone (CSBEZ) provides the spatial region an evader needs to avoid capture by a pursuer that is turn-rate constrained and is executing a curve-straight path. It is defined using known parameters, specifically the pursuer's initial position P, heading ψ_P, range R (equivalent to maximum flight time $R = v_P t$), minimum turn radius a, and the evader's initial position E, speed v_E, and heading ψ_E. To determine if a point is in the turn-constrained BEZ we need to determine if the pursuer can intercept the evader by traveling less than a distance R, provided the evader follows its current heading.

The geometry of the engagement zone can be determined by considering the path of the evader. If the evader travels along its current heading for the amount of time the pursuer travels its max range it would end at the point $F = E + \nu R \boldsymbol{v}_E$, where $\nu = v_E/v_P$ is the speed ratio and $\boldsymbol{v}_E$ is a unit vector pointing in the direction of the evader's heading. We assume the pursuer takes a curve-straight (CS) path to intercept the evader at F, which consists of a single arc (either left or right) of angle θ with a radius of a starting at initial position P ending at point G, then a straight-line path starting at G and ending at F. This path length is $\theta a + \|F - G\|_2$. If this path length is less than R the evader is inside the EZ.

To find the path length, we need to determine the point G at which the pursuer will leave the "turn" portion of the path and head straight towards point F. The point G occurs when F is along the tangent line of the circle, which can be found geometrically using the center point of the turn radius. Below we will find this tangent line from the left turn radius, which we define as G_ℓ. The center point C_ℓ for the left turn radius is

$$C_\ell(P, \psi_P, a) = P + a \begin{bmatrix} -\sin(\psi_P) \\ \cos(\psi_P) \end{bmatrix}. \tag{1}$$

To aid in finding G_ℓ we define four vectors that are functions of the pursuer's parameters (P, ψ_P, and a) and F:

$$\begin{aligned} \boldsymbol{v}_{1,\ell}(F, P, \psi_P, a) &= F - C_\ell(P, \psi_P, a) \\ \boldsymbol{v}_{2,\ell}(F, P, \psi_P, a) &= F - G_\ell(F, P, \psi_P, a) \\ \boldsymbol{v}_{3,\ell}(F, P, \psi_P, a) &= G_\ell(F, P, \psi_P, a) - C_\ell(P, \psi_P, a) \\ \boldsymbol{v}_{4,\ell}(F, P, \psi_P, a) &= P - C_\ell.(P, \psi_P, a) \end{aligned} \tag{2}$$

These provide the vectors (a) from the left turn radius center point to the end of the intercept point, (b) the tangent line to the intercept point, (c) the left turn radius center to the tangent point, and (d) the left turn radius center to the pursuer's initial point. All of these vectors are shown in Figure 1.

We drop the function notation on the vectors for clarity, but for future reference, it will be important to remember these are functions of the pursuer's parameters. Of there vectors two are known ($\boldsymbol{v}_{1,\ell}$ and $\boldsymbol{v}_{4,\ell}$) and two depend on the tangent point G_ℓ: ($\boldsymbol{v}_{2,\ell}$, $\boldsymbol{v}_{3,\ell}$. The vector $\boldsymbol{v}_{1,\ell}$ points from the center of the left-turn radius to the goal location. To solve for the tangent point, we will solve for $\boldsymbol{v}_{3,\ell}$, which is the vector from C_ℓ to G_ℓ. We know that the straight path the pursuer takes after it stops turning ($\boldsymbol{v}_{2,\ell}$) is tangent to the circle. This means $\boldsymbol{v}_{2,\ell}$ and $\boldsymbol{v}_{3,\ell}$ are perpendicular:

$$\boldsymbol{v}_{2,\ell} \cdot \boldsymbol{v}_{3,\ell} = 0. \tag{3}$$

We also note that $\boldsymbol{v}_{1,\ell}$ is a linear combination of $\boldsymbol{v}_{2,\ell}$ and $\boldsymbol{v}_{3,\ell}$

$$\boldsymbol{v}_{2,\ell} = \boldsymbol{v}_{1,\ell} - \boldsymbol{v}_{3,\ell}. \tag{4}$$

Substituting this into the orthogonality condition, we obtain a dot product expression involving $\boldsymbol{v}_{1,\ell}$ and $\boldsymbol{v}_{3,\ell}$. Since $\boldsymbol{v}_{3,\ell}$ lies on a circle of radius a, we can enforce the constraint $\|\boldsymbol{v}_{3,\ell}\| = a$. This yields the following relation:

$$\boldsymbol{v}_{1,\ell} \cdot \boldsymbol{v}_{3,\ell} = a^2. \tag{5}$$

The vector $\boldsymbol{v}_{3,\ell}$ can be decomposed into a component parallel to $\boldsymbol{v}_{1,\ell}$, which contributes to dot product, and a component perpendicular to $\boldsymbol{v}_{1,\ell}$, which does not. The orthogonal component can be used to ensure $\|\boldsymbol{v}_{3,\ell}\|_2 = a$. This decomposition takes the form

$$\boldsymbol{v}_{3,\ell} = \alpha \boldsymbol{v}_{1,\ell} + \beta \boldsymbol{v}_{1,\ell}^\perp, \tag{6}$$

where $\boldsymbol{v}_{1,\ell}^\perp$ is a unit vector perpendicular to $\boldsymbol{v}_{1,\ell}$. There are two possible perpendicular vectors, one formed by a 90 degree clockwise rotation and the other by a counter-clockwise rotation. To ensure $\boldsymbol{v}_{3,\ell}$ is pointing the correct direction we choose the clockwise perpendicular vector for $\boldsymbol{v}_{1,\ell}^\perp$:

$$\boldsymbol{v}_{1,\ell}^\perp = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \frac{\boldsymbol{v}_{1,\ell}}{\|\boldsymbol{v}_{1,\ell}\|}. \tag{7}$$

To solve for α we use the dot product relationship defined in Equation (5). Because we decomposed $\boldsymbol{v}_{3,\ell}$ into a parallel and orthogonal components, only the parallel component contributes to the dot product:

$$\boldsymbol{v}_{1,\ell} \cdot \boldsymbol{v}_{3,\ell} = \alpha \boldsymbol{v}_{1,\ell} \cdot \boldsymbol{v}_{1,\ell} + \beta \boldsymbol{v}_{1,\ell} \cdot \boldsymbol{v}_{1,\ell}^\perp = \alpha \|\boldsymbol{v}_{1,\ell}\|_2^2 = a^2 \tag{8}$$

which yields

$$\alpha = \frac{a^2}{\|\boldsymbol{v}_{1,\ell}\|_2^2}. \tag{9}$$

We will now use the fact that $\|\boldsymbol{v}_{3,\ell}\|_2 = a$ to solve for the perpendicular component of $\boldsymbol{v}_{3,\ell}$. Squaring both sides gives:

$$a^2 = \boldsymbol{v}_{3,\ell} \cdot \boldsymbol{v}_{3,\ell} \tag{10}$$

Substituting the decomposition $\boldsymbol{v}_{3,\ell} = \alpha \boldsymbol{v}_{1,\ell} + \beta \boldsymbol{v}_{1,\ell}^\perp$, we expand the dot product:

$$\begin{aligned} a^2 &= (\alpha \boldsymbol{v}_{1,\ell} + \beta \boldsymbol{v}_{1,\ell}^\perp) \cdot (\alpha \boldsymbol{v}_{1,\ell} + \beta \boldsymbol{v}_{1,\ell}^\perp) && (11) \\ &= \alpha^2 \boldsymbol{v}_{1,\ell} \cdot \boldsymbol{v}_{1,\ell} + 2\alpha\beta \boldsymbol{v}_{1,\ell} \cdot \boldsymbol{v}_{1,\ell}^\perp + \beta^2 \|\boldsymbol{v}_{1,\ell}^\perp\|^2 && (12) \end{aligned}$$

We now simplify the expression. Since $\boldsymbol{v}_{1,\ell}$ and $\boldsymbol{v}_{1,\ell}^{\perp}$ are orthogonal, the cross term vanishes, $\boldsymbol{v}_{1,\ell} \cdot \boldsymbol{v}_{1,\ell}^{\perp} = 0$ and because $\boldsymbol{v}_{1,\ell}^{\perp}$ is a unit vector, we have $\|\boldsymbol{v}_{1,\ell}^{\perp}\|^2 = 1$. Equation (10) simplifies to:

$$a^2 = \alpha^2 \|\boldsymbol{v}_{1,\ell}\|^2 + \beta^2 \tag{13}$$

Substituting in the value of $\alpha = \dfrac{a^2}{\|\boldsymbol{v}_{1,\ell}\|^2}$, we get:

$$a^2 = \frac{a^4}{\|\boldsymbol{v}_{1,\ell}\|^2} + \beta^2. \tag{14}$$

Solving for β yields:

$$\beta = \sqrt{a^2 - \frac{a^4}{\|\boldsymbol{v}_{1,\ell}\|^2}}. \tag{15}$$

Using the parallel (α) and perpendicular (β) components $\boldsymbol{v}_{3,\ell}$ is

$$\boldsymbol{v}_{3,\ell}(F, P, \psi_P, a) = \frac{a^2}{\|\boldsymbol{v}_{1,\ell}\|^2}\boldsymbol{v}_{1,\ell} + \sqrt{a^2 - \frac{a^4}{\|\boldsymbol{v}_{1,\ell}\|^2}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \frac{\boldsymbol{v}_{1,\ell}}{\|\boldsymbol{v}_{1,\ell}\|}. \tag{16}$$

Now that we have the vector starting at the left center point C_ℓ and ending at the tangent point G_ℓ we can solve for the tangent point as

$$G_\ell(F, P, \psi_P, a) = C_\ell + \boldsymbol{v}_{3,\ell}(F, P, \psi_P, a). \tag{17}$$

Using the tangent point G_ℓ, we can compute the total length of the pursuer's curve-straight path to the evader's projected position F. This path consists of two segments: an arc from the pursuer's initial position P to the tangent point G_ℓ, followed by a straight-line segment from G_ℓ to F. To compute the arc length, we first determine the angle swept out by the pursuer during the turn. This angle is measured from the vector pointing from the center of the left-turn circle to the pursuer's initial position, $\boldsymbol{v}_{4,\ell}$, to the vector pointing from the center to the tangent point, $\boldsymbol{v}_{3,\ell}$. Because the turn is counter-clockwise, we use the two-dimensional cross and dot products to compute the angle via the atan2 function:

$$\theta_{ccw}(F, P, \psi_P, a) = \operatorname{atan2}(\boldsymbol{v}_{4,\ell} \times \boldsymbol{v}_{3,\ell}, \boldsymbol{v}_{4,\ell} \cdot \boldsymbol{v}_{3,\ell}). \tag{18}$$

The path length is the sum of the arc length and the Euclidean distance from the tangent point to the final position

$$L_\ell(F, P, \psi_P, a) = a\theta_{ccw}(F, P, \psi_P, a) + \|G_\ell(F, P, \psi_P, a) - F\|_2. \tag{19}$$

If we need to instead compute the shortest right-straight path, then the only difference is which perpendicular vector we choose when constructing $\boldsymbol{v}_{3,r}$. We compute the clockwise angle instead of the counterclockwise angle, and choose the right center point. In the right-straight case

$$C_r(P, \psi_P, a) = P + a \begin{bmatrix} \sin(\psi_P) \\ -\cos(\psi_P) \end{bmatrix}, \tag{20}$$

$$\boldsymbol{v}_{3,r} = \frac{a^2}{\|\boldsymbol{v}_{1,r}\|^2}\boldsymbol{v}_{1,r} + \sqrt{a^2 - \frac{a^4}{\|\boldsymbol{v}_{1,r}\|^2}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \frac{\boldsymbol{v}_{1,r}}{\|\boldsymbol{v}_{1,r}\|}, \tag{21}$$

and

$$\theta_{cw}(F, P, \psi_P, a) = \operatorname{atan2}\left(-\boldsymbol{v}_{4,r} \times \boldsymbol{v}_{3,r}, \boldsymbol{v}_{4,r} \cdot \boldsymbol{v}_{3,r}\right). \tag{22}$$

Using this angle, we define the shortest right-turn length as

$$L_r(F, P, \psi_P, a) = a\theta_{cw}(F, P, \psi_P, a) + \|G_r(F, P, \psi_P, a) - F\|_2. \tag{23}$$

The overall shortest turn-straight length is

$$L(F, P, \psi_P, a) = \min(L_r(F, P, \psi_P, a), L_\ell(F, P, \psi_P, a)). \tag{24}$$

If F is to the right of the Pursuer's heading, then the right-straight path will be shorter than the left-straight path. We define the EZ function as

$$z(E, \psi_E, P, \psi_P, a, R, \nu) = L(E + \nu R \boldsymbol{v}_{\psi_E}, P, \psi_P, a) - R. \tag{25}$$

The CSBEZ consists of all the points where the minimum turn-straight path length required to reach the evader's future position, after the pursuer has traveled its maximum range, is less than the pursuer's range. The CSBEZ is defined as

$$\mathcal{Z} = \{E | z(E, \psi_E, P, \psi_P, a, R, \nu) \leq 0\}. \tag{26}$$

Figure 1 shows both the reachable set and the CSBEZ. Also shown are the points and vectors that were used to compute the left-straight path length.

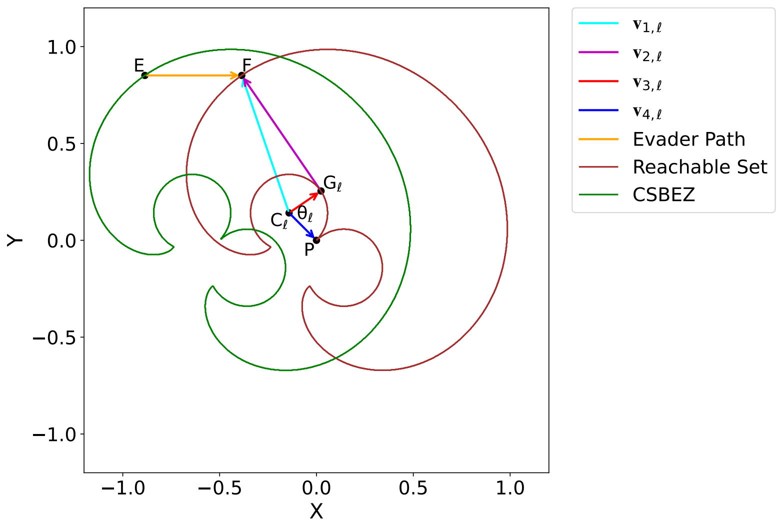

Fig. 1 This figures shows the reachable set of the purser in brown. The CSBEZ is shown in green. The evader starting E and final F positions are shown as well as the pursuer's starting position P, the tangent point G_ℓ where the pursuer switches from a curved path to a straight one, and the center of the left turn radius C_ℓ. The four vectors used to find the path length are also shown as $v_{1,\ell}, v_{2,\ell}, v_{3,\ell}$, and $v_{4,\ell}$.

III. Curve-Straight Probabilistic Engagement Zone (CSPEZ)

The above section found the CSBEZ if the pursuer's parameters are known. However, in adversarial environments, enemy parameters are often unknown or uncertain. The CSPEZ finds the probability that the evader is in the true CSBEZ, when accounting for uncertainty in the pursuer's parameters. We do this through four uncertainty propagation techniques that we will describe in the subsequent sections: Monte Carlo CSPEZ (MCCSPEZ), linearized CSPEZ (LCSPEZ), quadratic CSPEZ (QCSPEZ), and neural network CSPEZ (NNCSPEZ).

For this work, we assume the agent has a prior belief of the pursuer's parameters represented as a probability distribution. We combine all the pursuer parameters into a single vector $\Theta_P = [x_P, y_P, \psi_P, a, R, \nu_P]^\top$ and assume a Gaussian distribution $\Theta_P \sim \mathcal{N}(\mu_{\Theta_P}, \Sigma_{\Theta_P})$ on their distributions with mean $\mu_{\Theta_P} = [\mu_{x_P}, \mu_{y_P}, \mu_{\psi_P}, \mu_a, \mu_R, \mu_{\nu_P}]^\top$ and covariance

$$\Sigma_{\Theta_P} = \begin{bmatrix} \Sigma_P & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} \\ \mathbf{0}_{1\times2} & \sigma^2_{\psi_P} & 0 & 0 & 0 \\ \mathbf{0}_{1\times2} & 0 & \sigma^2_a & 0 & 0 \\ \mathbf{0}_{1\times2} & 0 & 0 & \sigma^2_R & 0 \\ \mathbf{0}_{1\times2} & 0 & 0 & 0 & \sigma^2_{\psi_P} \end{bmatrix}. \tag{27}$$

Note that the vector of mean values is the mean value of each individual pursuer parameters and Σ_P is the covariance of the pursuer's position. The other diagonal elements are the variances of the remaining pursuer parameters. We assume the x and y values of the pursuer's position are correlated, and that there is no correlation between the rest of the pursuer's parameters. If there was a correlation between any of these parameters the off-diagonal elements of Σ_{Θ_P} would be non-zero. The probability density function (PDF) of Θ_P is given by

$$f_{\Theta_P}(\tau) = \frac{1}{\sqrt{(2\pi)^6 \, |\Sigma_{\Theta_P}|}} \exp\left(-\frac{1}{2} \left(\tau - \mu_{\Theta_P} \right)^\top \Sigma_{\Theta_P}^{-1} \left(\tau - \mu_{\Theta_P} \right) \right). \tag{28}$$

We also combine the evader's parameters into a single vector $\Theta_E = [x_E, y_E, v_E]^\top$. We assume the evader's parameters are known for this work, however, extending these methods to account for uncertainty in evader parameters would be straightforward and interesting.

The CSBEZ, Equation (25), can be written as a function of the pursuer and evader parameters $z(\Theta_P, \Theta_E)$. Given a mean and covariance of the pursuer parameters $(\mu_{\Theta_P}, \Sigma_{\Theta_P})$, we wish to find the probability that an evader is in the true CSBEZ. Note that this equation operates under a specific evader configuration Θ_E and the result will change as these parameters change. Because the evader is in the CSBEZ when $z(\Theta_P, \Theta_E) \leq 0$, the probability the evader is inside the CSBEZ is the same as finding the probability of the function being less then zero. This can be solved by integrating the PDF of Θ_P over the region where $z(\Theta_P, \Theta_E) \leq 0$,

$$P_{CSPEZ}(\mu_{\Theta_P}, \Sigma_{\Theta_P}, \Theta_E) = P(z(\Theta_P, \Theta_E) \leq 0) = \int_{\{\tau | z(\tau, \Theta_E) \leq 0\}} f_{\Theta_P}(\tau) d\tau. \tag{29}$$

This integral has no analytic solution because of the coupling between the dimensions of Θ_P and must be approximated. Note that this integral is evaluated for a specific configuration of the pursuer, characterized by a given mean and covariance, and for a fixed set of evader parameters. As a result, a new integral must be computed for each unique combination of pursuer distribution and evader parameters. For example, if this probability is used as a constraint in a path planning algorithm, the integral must be re-evaluated at multiple points along the path, using the position and heading defined by the trajectory. In the following sections we present four methods for approximating this integral: MCCPEZ, LCSPEZ, QCSPEZ, NNCSPEZ.

A. Monte Carlo Probabilistic Engagement Zone (MCCSPEZ)

The MCCSPEZ builds from our previous work in [8], and relies on using Monte Carlo integration to perform the integral in Equation (29). However, instead of using the simpler, unlimited turn-rate BEZ, we use the CSBEZ model. MCCSPEZ works by first drawing N_m random samples from the distribution of pursuer parameters $\Theta_P^i \sim \mathcal{N}(\mu_{\Theta_P}, \Sigma_{\Theta_P}), i = 1, \ldots, N_m$. We then evaluate the CSBEZ function for each random sample, given the evader parameters, and count the number that are less than zero. The MCCPEZ is then

$$P_{MCCSPEZ}(\mu_{\Theta_P}, \Sigma_{\Theta_P}, \Theta_E) = P(z(\Theta_P, \Theta_E) \leq 0) \approx \frac{1}{N_m} \sum_{i=0}^{N_m} \mathbb{1}\left(-z(\Theta_P^i, \Theta_E) \right), \tag{30}$$

where $\mathbb{1}(z)$ is an indicator function defined as

$$\mathbb{1}(z) = \left\{ \begin{array}{ll} 0 & z < 0 \\ 1 & z \geq 0 \end{array} \right\}. \tag{31}$$

As the number of random samples increases, the MCCSPEZ approximation will improve. One downside of the MCCSPEZ method is the large number of samples needed for accurate approximations. Another is that if a gradient-based path optimization algorithm is being used, the Jacobian of the MCCSPEZ is needed. However, because of the flat indicator function, the Jacobian is zero everywhere except for the switching point, and at the switching point it is undefined.

B. Linearized Probabilistic Engagement Zone (LCSPEZ)

The LCSPEZ method overcomes the large sample size, and Jacobian problems of the MCCSPEZ, at the expense of accuracy. This method relies on creating a linear model of the CSPEZ funtion and using the linearization to model the integral in Equation (29) has a well defined solution. A similar method for the simple BEZ was presented in [8].

We create the linear model using a first-order Taylor series approximation centered around the mean value of the pursuer parameters. We do this by finding the Jacobians of the CSPEZ function with respect ot the pursuer parameters,

$$z(\Theta_P + \boldsymbol{\delta}_P, \Theta_E) \approx z(\Theta_P, \Theta_E) + \boldsymbol{\delta}_P \frac{\partial z}{\partial \Theta_P}, \tag{32}$$

where $\partial z / \partial \Theta_P$ is the Jacobian of z with respect to Θ_P. The Jacobians can be found analytically or through automatic differentiation. In this work, we used the JAX automatic differentiation library [14]. Using this model we can compute the mean and variance of the output of the CSBEZ function as

$$\mu_z(\mu_{\Theta_P}, \Theta_E) = z(\mu_{\Theta_P}, \Theta_E) \tag{33}$$

and

$$\sigma_z^2(\mu_{\Theta_P}, \Sigma_{\Theta_P}, \Theta_E) = \frac{\partial z}{\partial \Theta_P}\left(\mu_{\Theta_P}, \Theta_E\right) \Sigma_{\Theta_P} \frac{\partial z}{\partial \boldsymbol{\Theta}_P}\left(\mu_{\Theta_P}, \Theta_E\right)^\top . \tag{34}$$

Using this we then find the LCSPEZ probability as

$$P_{LCSPEZ}(\mu_{\Theta_P}, \Sigma_{\Theta_P}, \Theta_E) = P(z(\Theta_P, \Theta_E) \leq 0) \approx F(0; \mu_z(\mu_{\Theta_P}, \Theta_E), \sigma^2(\mu_{\Theta_P}, \Sigma_{\Theta_P} \Theta_E)), \tag{35}$$

where $F(x; \mu, \sigma^2)$ is the single variable Gaussian cumulative distribution function (CDF). The CDF can be written in terms of the standard error function as

$$F(x; \mu, \sigma^2) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\,\sigma} \exp\!\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) \mathrm{d}t = \frac{1}{2}\left[1 + \operatorname{erf}\!\left(\tfrac{x-\mu}{\sigma\sqrt{2}}\right)\right]. \tag{36}$$

It is important to note that both the mean $\mu_z(\mu_{\Theta_P}, \Theta_E)$ and the variance $\sigma_z^2(\mu_{\Theta_P}, \Theta_E)$ depend on the evader parameters (position, heading, and speed). If the LCSPEZ is used as a constraint within a path planning algorithm, a distinct mean and variance must be evaluated at points sampled along the trajectory, based on the position and heading defined by the path. The corresponding LCSPE values can then be computed at each point using the Gaussian CDF. In our formulation, we treat the pursuer's parameter distribution as fixed throughout the trajectory. However, if the distribution were time-varying—such as if the covariance evolved over time or the initial position of the pursuer were dynamic—then the relevant parameters would also need to be evaluated as functions of time and sampled along the path.

The LCSPEZ has a smaller memory and complexity cost than MCCSPEZ, at the cost of accuracy. Specifically, the LCSPEZ struggles when the CSPEZ function (Equation (25)) is highly non-linear, contains discontinuities, or when the Jacobian is zero. LCSPEZ also struggles when there are large levels of uncertainty.

C. Quadratic Probabilistic Engagement Zone (QCSPEZ)

Instead of creating a linear model of the CSBEZ function, QCSPEZ creates a quadratic model. This helps in situations where the CSBEZ is more nonlinear or where the Jacobian is zero. Instead of creating a first-order Taylor series, we create a second-order approximation

$$z(\Theta_P + \boldsymbol{\delta}_P, \Theta_E) \approx z(\Theta_P, \Theta_E) + \boldsymbol{\delta}_P^T \frac{\partial z}{\partial \Theta_P}\left(\Theta_P, \Theta_E\right) + \tfrac{1}{2}\, \boldsymbol{\delta}_P^T \frac{\partial^2 z}{\partial \Theta_P^2}\left(\Theta_P, \Theta_E\right) \boldsymbol{\delta}_P, \tag{37}$$

where $\partial^2 z / \partial \Theta_P^2$ is the Hessian of the CSPEZ equation with respect to the pursuer parameters. Unlike the LCSPEZ, there is no analytic solution of a quadratic function of a Gaussian random variable, so we must approximate. We do this by first finding the mean value of the quadratic approximation

$$\mu_z(\mu_{\Theta_P}, \Sigma_{\Theta_P}, \Theta_E) = z(\mu_{\Theta_P}, \Theta_E) + \tfrac{1}{2} \operatorname{tr}\left(\frac{\partial^2 z}{\partial \Theta_P^2}(\mu_{\Theta_P}, \Theta_E)\, \Sigma_{\Theta_P}\right) \tag{38}$$

and variance

$$\sigma_z^2(\mu_{\Theta_P}, \Sigma_{\Theta_P}, \Theta_E) = \frac{\partial z}{\partial \Theta_P}(\mu_{\Theta_P}, \Theta_E)\Sigma_{\Theta_P} \frac{\partial z}{\partial \Theta_P}(\Theta_P, \Theta_E)^\top + 2\, \operatorname{tr}\left(\frac{\partial^2 z}{\partial \Theta_P^2}(\mu_{\Theta_P}, \Theta_E)\Sigma_{\Theta_P} \frac{\partial^2 z}{\partial \Theta_P^2}(\mu_{\Theta_P}, \Theta_E)\Sigma_{\Theta_P}\right). \tag{39}$$

Using this mean and variance we then approximate the QCSPEZ probability as

$$P_{QCSPEZ}(\mu_{\Theta_P}, \Sigma_{\Theta_P}, \Theta_E) = P(z(\Theta_P, \Theta_E) \leq 0) \approx F(0; \mu_z(\mu_{\Theta_P}, \Theta_E), \sigma^2(\mu_{\Theta_P}, \Sigma_{\Theta_P}\Theta_E)), \tag{40}$$

where F is the single variable Gaussian CDF, defined in Equation (36). Just like the LCSPEZ, there is a unique mean and variance for a given set of evader parameters (position, heading, speed). If QCSPEZ is used as a constraint for a path planning algorithm, the mean and variance must be sampled at points along the trajectory. The QCSPEZ probability can then be found using the Gaussian CDF.

Similar to the LCSPEZ, the QCSPEZ struggles with high nonlinearities and discontinuities in the CSBEZ function. However, it provides a better approximation than LCSPEZ when the gradient is zero because it incorporates information from the Hessian.

D. Neural Network Probabilistic Engagement Zone (NNCSPEZ)

Instead of first approximating the CSBEZ equation, NNCSPEZ learns a direct mapping from the pursuer's parameter mean and covariance, along with the evader's parameters, to the probability that the evader lies within the CSBEZ. This is achieved using a multilayer perceptron (MLP) regressor that predicts the probability from the combined input.

To reduce the input dimensionality and exploit problem symmetries, the MLP input is expressed in a relative frame: the pursuer is placed at the origin with zero heading, and the evader's position and heading are defined relative to this frame. The combined input vector for the MLP is constructed as:

$$x_{\text{NN}}(\mu_{\Theta_P}, \Sigma_{\Theta_P}, \Theta_E) = \begin{bmatrix} \mu_a \\ \mu_R \\ \mu_{v_P} \\ \sigma^2_{x_P} \\ \sigma^2_{y_P} \\ \sigma_{x_P y_P} \\ \sigma^2_{\psi_P} \\ \sigma^2_a \\ \sigma^2_R \\ \sigma^2_{v_P} \\ x_E - \mu_{x_P} \\ y_E - \mu_{y_P} \\ \psi_E - \mu_{\psi_P} \\ v_E \end{bmatrix} \in \mathbb{R}^{14}. \tag{41}$$

The neural network then learns a function $f_\phi \colon \mathbb{R}^{14} \to [0, 1]$, where $f_\phi(x_{\text{NN}})$ approximates the probability that the evader lies within the CSBEZ, given the specified uncertainty and relative configuration.

We implement the probabilistic engagement regressor as a fully connected MLP using the Flax framework [15]. The network receives as input the 14-dimensional feature vector $x_{\text{NN}} \in \mathbb{R}^{14}$ and propagates it through four hidden layers with widths of 512, 256, 256, and 128 neurons, respectively. Each hidden layer applies layer normalization followed by a sigmoid linear unit (SiLU) activation. Formally, the hidden activations are computed as:

$$\begin{aligned} h^{(0)} &= x_{\text{NN}}, \\ h^{(i)} &= \text{SiLU}\big(\text{LayerNorm}(W^{(i)} h^{(i-1)} + b^{(i)})\big), \quad i = 1, \ldots, 4, \end{aligned} \tag{42}$$

where the weight matrices are

$$\begin{aligned} W^{(1)} \in \mathbb{R}^{512\times14}, \quad W^{(2)} \in \mathbb{R}^{256\times512}, \\ W^{(3)} \in \mathbb{R}^{256\times256}, \quad W^{(4)} \in \mathbb{R}^{128\times256}, \end{aligned} \tag{43}$$

with corresponding bias vectors $b^{(i)}$. A final output layer defines the output of the neural network:

$$f_\phi(x_{\text{NN}}) = \sigma\big(w\, h^{(4)} + b\big), \tag{44}$$

where $w \in \mathbb{R}^{1\times 128}$ and $b \in \mathbb{R}$ are the weights and bias of the output layer, and $\sigma(z) = 1/(1+e^{-z})$ is the sigmoid function, ensuring that $f_\phi(x_{\text{NN}}) \in [0,1]$. The complete set of parameters ϕ includes all weight matrices $W^{(i)}$, bias vectors $b^{(i)}$, the output weights w, and output bias b.

To train the network, we generate N_T samples $\{x_{\text{NN}}^{(i)}\}_{i=1}^{N_T}$ using Latin hypercube sampling [16], drawing from uniform distributions over the admissible range of each feature. For each sample, we compute the corresponding Monte Carlo CSPEZ probability $P_{MCCSPEZ}^{(i)}$, which serves as the ground truth. We then optimize the network parameters ϕ using the Adam optimizer [17] to minimize the root mean square error (RMSE) loss:

$$\mathcal{L}(\phi) = \sqrt{\frac{1}{N_T}\sum_{i=1}^{N_T}\big(f_\phi(x_{\text{NN}}^{(i)}) - P_{MCCSPEZ}^{(i)}\big)^2}\,. \tag{45}$$

Using the trained network, the NNCSPEZ is defined as

$$P_{NNCSPEZ}(\mu_{\Theta_P}, \Sigma_{\Theta_P}, \Theta_E) = f_\phi\big(x_{\text{NN}}(\mu_{\Theta_P}, \Sigma_{\Theta_P}, \Theta_E)\big). \tag{46}$$

If trained effectively, the NNCSPEZ will achieve improved accuracy over LCSPEZ and QCSPEZ approximations, while offering significantly reduced memory requirements compared to MCCSPEZ. An additional advantage is that the Jacobian of the network output with respect to its inputs is readily available, enabling its integration into gradient-based path planning algorithms.

IV. Path Planning Using CSPEZ

To illustrate the utility of the CSPEZ formulations, we now present a path optimization framework that incorporates CSPEZ-based constraints. In contested environments, agents require safe trajectories that avoid potential engagements while accounting for uncertainty in adversary information. Prior work in [4] and [3] used cardioid-shaped EZs to compute minimum-time paths that avoid threats. The BEZ formulation in [6] extended this concept to model an infinite-turn-rate pursuer for conflict-free trajectory planning. In our earlier work [8], we introduced a linearized probabilistic engagement zone (PEZ) to handle parameter uncertainty in the planning process. Here, we build on that foundation by integrating the CSPEZ approximations—LCSPEZ, QCSPEZ, and NNCSPEZ—directly as constraints within the trajectory optimization routine.

We use B-splines to parameterize our trajectories because they are easily differentiable and have local support (sparse Jacobians) to aid in trajectory optimization. B-splines are parammeterized by a set of control points $\overline{C} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{N_c})$, where N_c is the number of control points, and knot points $\boldsymbol{t}_k = (t_0 - k\Delta_t, \ldots, t_0 - \Delta_t, t_0, t_0 + \Delta_t, \ldots, t_f, t_f + \Delta_t, \ldots, t_f + k\Delta_t)$, where t_0 is the starting time of the trajectory, t_f is the final time of the trajectory, k is the order of the B-spline, $\Delta_t = (t_f - t_0)/N_k$ is the time spacing of the knot points, and N_k is the number of internal knot points. The B-spline path is then defined as a weighted sum of basis functions where the basis functions are defined using the knot points and the weights are the control points

$$\boldsymbol{p}(t) = \sum_{i=1}^{N_c} B_{i,k}(t)\boldsymbol{c}_i. \tag{47}$$

The basis functions are defined using the Cox-De-Boor recursive formula [18].

The goal of the path planning algorithm is to find the minimum time trajectory for the evader staring at an initial position E_0 and ending at a goal location E_f. The CSPEZ approximations are used as constraints to ensure the probability of entering the true CSBEZ is held below a threshold ϵ. We also employ kinematic feasibility constraints that are defined using differential flatness, assuming the evader follows a unicycle kinematic model. The path optimization

Table 1 Error Metrics For Each Approximation

Approximation	RMSE	Average Absolute Error	Max Error
LCSPEZ	0.01343	0.0655	0.9160
QCSPEZ	0.00792	0.0573	0.8522
NNCSPEZ	**2.645e-6**	**0.0009**	**0.0448**

problem is

$$\overline{C}_{opt}, t_{f_{opt}} = \underset{\overline{C}, t_f}{\operatorname{argmin}}\, t_f \tag{48a}$$

$$\text{subject to } \boldsymbol{p}(0) = E_0 \tag{48b}$$

$$\boldsymbol{p}(t_f) = E_f \tag{48c}$$

$$\boldsymbol{p}(\boldsymbol{t}_s) \in \mathcal{D} \tag{48d}$$

$$P_{CSPEZ}(\mu_{\Theta_P}, \Sigma_{\Theta_P}, \Theta_E(\boldsymbol{t}_s) \leq \epsilon \tag{48e}$$

$$v(\boldsymbol{t}_s) = v_E \tag{48f}$$

$$u_{lb} \leq u_E(\boldsymbol{t}_s) \leq u_{ub} \tag{48g}$$

$$-\kappa_{ub} \leq \kappa_E(\boldsymbol{t}_s) \leq \kappa_{ub}, \tag{48h}$$

where u_{lb} and u_{ub} are the lower and upper turn-rate constraints, κ_{ub} is the path curvature constraint and $u_E(t)$ and $\kappa_E(t)$ are the turn rate and curvature of the trajectory. The constraints must be discretely sampled, $\boldsymbol{t}_s = \{0, \Delta_s, 2\Delta_s, \ldots, t_f\}$, $\Delta_s = t_f/N_s$, where N_s is the number of discrete constraint samples. The first two constraints ensure that the trajectory starts and ends at the desired point. The third constraint (48d) ensures that the evader remains within the desired operating region. The next constraint (48e), is the CSPEZ constraint. We alternatively use LCSPEZ, QCSPEZ, and NNCSPEZ to approximate this constraint. A comparison of these three constraints is shown in Section V. The next three constraints ensure the trajectory is kinematically feasible for the evader. Assuming the evader follows a kinematic unicycle model, the velocity is $v(t) = ||\dot{\boldsymbol{p}}(t)||_2$ and is contained to follow a set velocity of v_E. The turn rate is $u(t) = (\dot{\boldsymbol{p}}(t) \times \ddot{\boldsymbol{p}}(t))/||\dot{\boldsymbol{p}}(t)||_2^2$ is bounded to fall between u_{lb} and u_{ub}. And finally, the curvature of the evader's trajectory is also bounded and can be found as $\kappa(t) = u(t)/v(t)$.

We use a gradient-based interior point optimization algorithm called IPOPT [19] to perform the optimization problem in Equation (48). Because we use a gradient-based optimization algorithm, the Jacobians of the constraints are needed. We use Jax [14] to perform automatic differentiation to find the Jacobians of the constraints. One reason the MCCSPEZ approximation is not used in the path planning algorithm is because it has a zero value for its Jacobian due to the flat indicator function, despite it being the most accurate approximation.

V. Results

In this section, we provide results showing the utility of our CSPEZ formulation. First, we evaluate the accuracy of the LCSPEZ, QCSPEZ, and NNCSPEZ approximations compared to the MCCSPEZ baseline. Then we show results for the path planning algorithm when each approximation is used as a constraint.

A. CSPEZ Comparison

To evaluate the accuracy of each approximation, we generate a test set consisting of $N_T = 500{,}000$ samples. Each sample defines a complete CSPEZ configuration consisting of evader parameters, a pursuer mean, and a pursuer covariance. These samples are drawn independently from uniform distributions over the admissible ranges of each parameter. We denote the resulting sets as

$$\begin{aligned} \{\Theta_E^i\}_{i=1}^{N_T} &= \overline{\Theta}_E, \\ \{\mu_{\Theta_P}^i\}_{i=1}^{N_T} &= \overline{\mu}_{\Theta_P}, \\ \{\Sigma_{\Theta_P}^i\}_{i=1}^{N_T} &= \overline{\Sigma}_{\Theta_P}, \end{aligned} \tag{49}$$

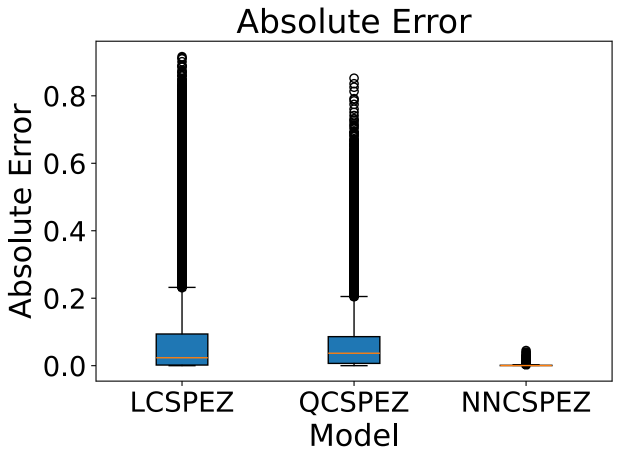

Fig. 2 A box plot of the absolute error compared to the MCCSPEZ baseline is shown for each model. NNCSPEZ clearly outperforms the other methods.

where each tuple $(\mu^i_{\Theta_P}, \Sigma^i_{\Theta_P}, \Theta^i_E)$ represents a unique configuration over which the CSPEZ probability approximations are evaluated. These points are generated separately from the points used to train the NNCSPEZ model.

To quantify the accuracy of each approximation method, we evaluate the root mean squared error (RMSE), average absolute error (AAE), and maximum absolute error (MaxAE) relative to the MCCSPEZ baseline. These metrics are computed over a test set of $N_T = 500{,}000$ configurations. For each sample i, let $p^i_{\text{approx}} = P_{APPROX}(\mu^i_{\Theta_P}, \Sigma^i_{\Theta_P}, \Theta^i_E)$ denote the predicted CSPEZ probability from a given approximation method—specifically, LCSPEZ (Equation (35)), QCSPEZ (Equation (40)), or NNCSPEZ (Equation (46)). Let $p^i_{\text{MCCSPEZ}} = P_{MCCSPEZ}(\mu^i_{\Theta_P}, \Sigma^i_{\Theta_P}, \Theta^i_E)$ denote the corresponding probability from the MCCSPEZ baseline. The error metrics are then defined as

$$\text{RMSE} = \sqrt{\frac{1}{N_T}\sum_{i=1}^{N_T}\left(p^i_{\text{approx}} - p^i_{\text{MCCSPEZ}}\right)^2}, \tag{50}$$

for the root mean square error,

$$\text{AAE} = \frac{1}{N_T}\sum_{i=1}^{N_T}\left|p^i_{\text{approx}} - p^i_{\text{MCCSPEZ}}\right|, \tag{51}$$

for the absolute error, and

$$\text{MaxAE} = \max_i \left|p^i_{\text{approx}} - p^i_{\text{MCCSPEZ}}\right|, \tag{52}$$

for the maximum error.

RMSE penalizes larger errors more heavily and provides an overall measure of approximation accuracy. AAE represents the average magnitude of the approximation error, while MaxAE captures the worst-case deviation from the baseline. The results of these metrics for each method are shown in Table 1. From this table and figure we can see that NNCSPEZ is the best approximation. However, in many situations the LCSPEZ and QCSPEZ perform adequately.

In addition to the summary statistics, Figure 2 shows the full distribution of absolute errors across all test points, providing insight into the spread and variability of errors for each approximation method. As can be seen, NNCSPEZ performs the best overall, however, both LCSPEZ and QCSPEZ perform adequately in a large range of configurations.

To illustrate where the LCSPEZ and QCSPEZ perform well, we show an example scenario where the pursuer's

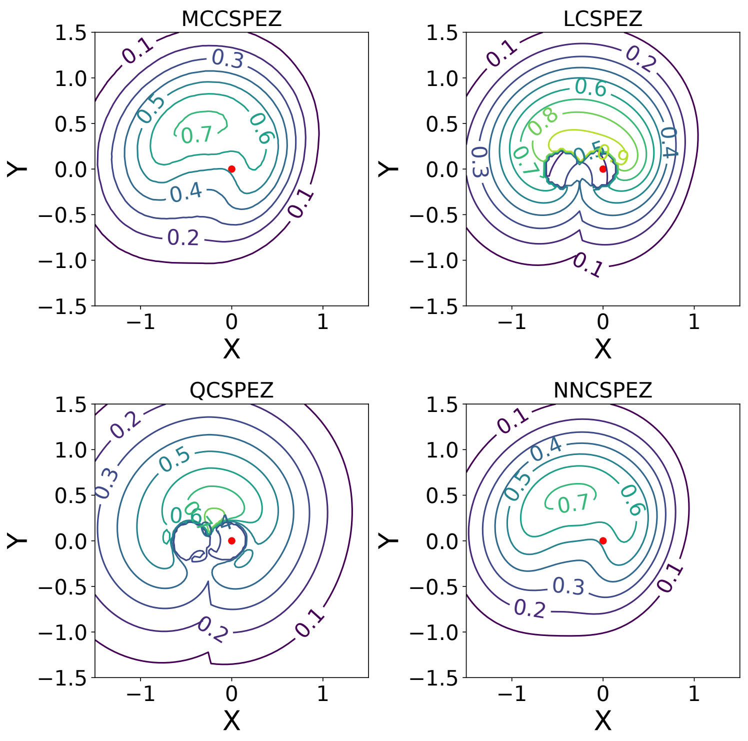

Fig. 3 The level sets of approximated CSPEZ for an example scenario. The pursuer's heading mean value is $\pi/2$ and the pursuer's mean position is shown in red. We plot the $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$ level sets of probability. The top left figure is the MCCSPEZ baseline. Top right shows LCSPEZ. Bottom left is QCSPEZ and bottom left is the NNCSPEZ approximation. Note that visual observation shows that NNCSPEZ provides the closest approximation to the MCCSPEZ baseline.

parameter mean value is $\mu_{\Theta_P} = [0.0, 0.0, \pi/2, 0.2, 1.0, 2.0]^\top$ and covariance is

$$\Sigma_{\Theta_P} = \begin{bmatrix} \begin{bmatrix} 0.025 & 0.04 \\ 0.04 & 0.1 \end{bmatrix} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} \\ \mathbf{0}_{1\times2} & 0.2 & 0 & 0 & 0 \\ \mathbf{0}_{1\times2} & 0 & 0.005 & 0 & 0 \\ \mathbf{0}_{1\times2} & 0 & 0 & 0.1 & 0 \\ \mathbf{0}_{1\times2} & 0 & 0 & 0 & 0.3 \end{bmatrix}. \quad (53)$$

We fix the evader heading at 0.0 and evaluate each model on a uniform grid of evader positions in the x-y plane. At each point, the CSPEZ probability is computed using the given approximation method, while holding the pursuer parameters constant. The resulting probability values are visualized using level sets, where each contour corresponds to a fixed CSPEZ probability threshold. These level sets are shown in Figure 3.

To assess the accuracy of each approximation, we compute the absolute error between the predicted probability and the MCCSPEZ baseline at each grid point. These spatial error distributions are shown in Figure 4. In this scenario, the mean value of the pursuer's heading points in the positive y direction. The largest errors for the LCSPEZ and QCSPEZ approximations occur in regions near the shifted turn radii of the purser. This corresponds to the cavity in the deterministic CBEZ from Figure 1. The high error in these regions is due to discontinuities in the CSBEZ surface, which reduce the accuracy of the linear and quadratic approximations. However, the overall shapes of the probability contours remains similar accross all methods.

We also show how well each approximation performs under varying levels of uncertainty. To do this, we compute the trace of each pursuer covariance matrix to represent the overall level of uncertainty for that configuration. The 500,000 test samples are sorted by trace and grouped into uniformly spaced bins. Within each bin, we compute the

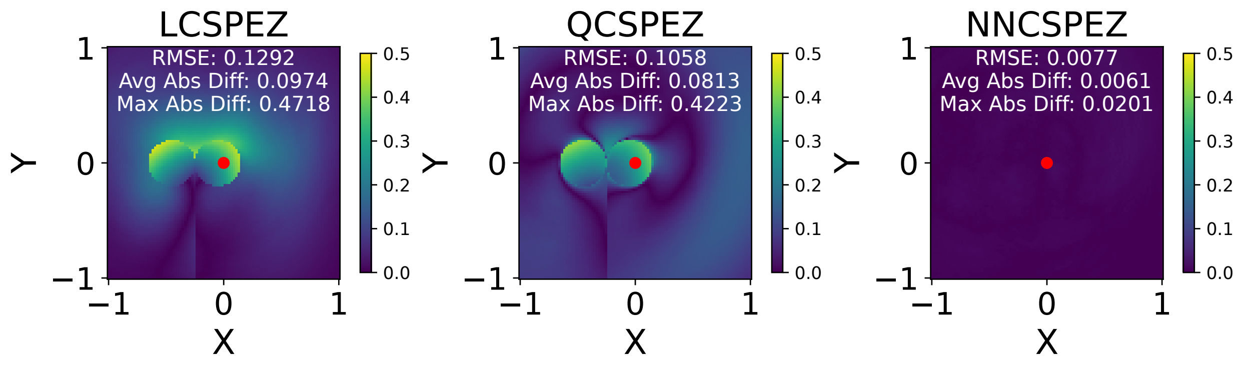

Fig. 4 The absolute error between the three approximations and the MCCSSPEZ baseline for the example scenario. The pursuer's heading mean value is $\pi/2$ and the pursuer's mean position is shown in red. For reference, the RMSE, average, and maximum absolute error values for this example scenario are also included in each plot.

Table 2 Path Planning Comparison

	LCSPEZ			QCSPEZ			NNCSPEZ		
CSPEZ	t_f	MCCSPEZ	Opt Time	t_f	MCCSPEZ	Opt Time	t_f	MCCSPEZ	Opt Time
0.01	11.76	0.0013	0.5006	12.21	0.0000	0.5943	11.62	0.0087	0.4987
0.05	11.58	0.0213	0.4057	11.81	0.0010	0.5351	11.49	0.0563	0.4277
0.25	11.38	0.1761	0.3453	11.42	0.1117	0.3380	11.34	0.2562	0.4326
0.5	11.29	0.3973	0.2762	11.24	0.5336	0.4150	11.25	0.4986	0.4063

average absolute error between each approximation and the MCCSPEZ baseline. This analysis is shown in Figure 5. As the level of uncertainty increases, the accuracy of both the LCSPEZ and QCSPEZ deteriorates. This is expected, since both the linear and quadratic approximations are local to the mean value, and higher uncertainty leads to larger deviations from the region where the approximation is valid. In contrast, the NNCSPEZ approximation does not exhibit a strong dependence on uncertainty, as it is trained directly on CSPEZ probability values. However, the NNCSPEZ can suffer in configurations that are far from its training distribution.

B. Path Planning

To evaluate the effect of each approximation on trajectory optimization, we use the CSPEZ probability as a constraint in the path planning algorithm described in Section IV. At each discretized point along the B-spline trajectory, the CSPEZ probability is evaluated using the chosen approximation method. A constraint is enforced such that the probability of the evader being inside the engagement zone does not exceed a user-specified threshold.

We use the same mean and covariance for the pursuer's parameters as in Figure 3 to test the CSPEZ approximations as constraints for the path planning algorithm outlined in Section IV. The evader's velocity is 1.0, the turn-rate limit is ± 1.0(rad/s) and the curvature limit is 0.2(rad/m). The evader's starting location is $(-4.0, -4.0)$ and goal $(4.0, 4.0)$. The B-spline trajectory has 8 control points and has a degree of 3. CSPEZ probability limits of $[0.01, 0.05, 0.25, 0.5]$ are used.

Figure 6 shows paths planned using LCSPEZ, QCSPEZ, and NNCSPEZ as approximations for the CSPEZ constraint. For each resulting trajectory, we report three quantities: the final time t_f of the trajectory (proportional to path length), the maximum value of the baseline MCCSPEZ probability along the path (i.e., the least safe point on the trajectory), and the total optimization time required to compute the solution. These values are shown in Table 2. From the table we see that NNCSPEZ consistently provided the closest approximation to the MCCSPEZ baseline when used as a constraint. Both LCSPEZ and QCSPEZ result in more conservative paths, reflected in lower maximum MCCSPEZ values along the trajectory. For higher CSPEZ constraints, LCSPEZ also tends to result in shorter optimization times compared to NNCSPEZ.

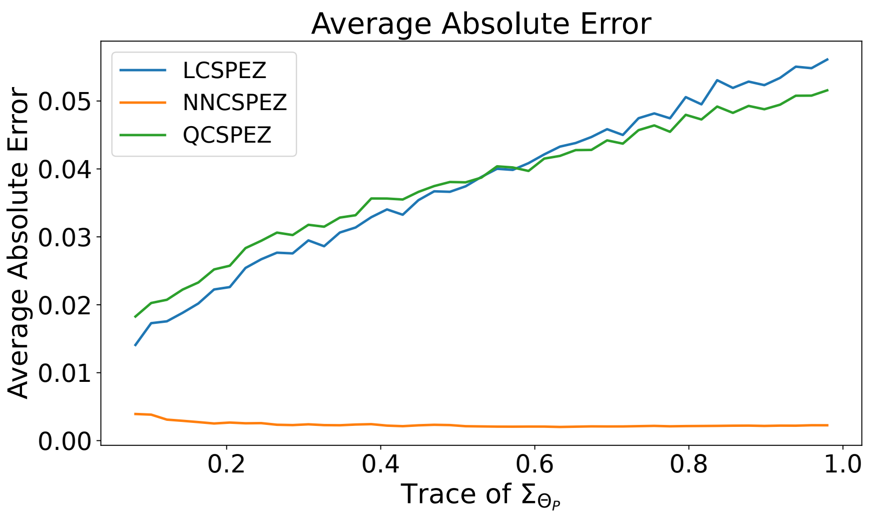

Fig. 5 The absolute error between the MCCSPEZ baseline and other approximation methods versus the trace of the pursuer parameter covariance. This shows that as the amount of uncertainty increases both the LCSPEZ and QCSPEZ approximations deteriorate as the NNCSPEZ stays accurate.

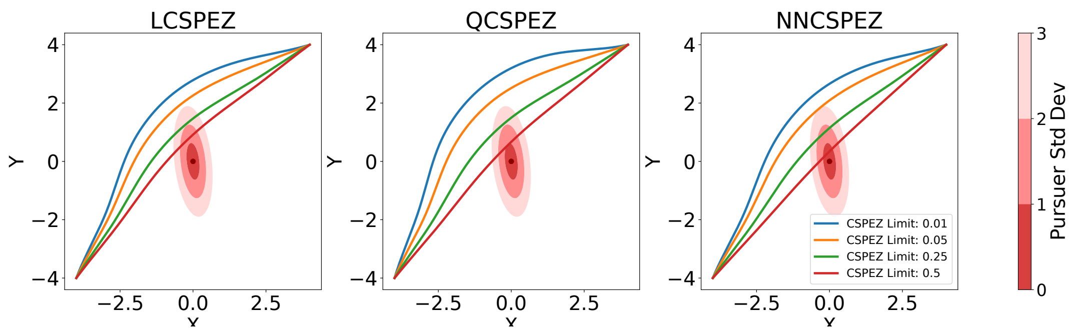

Fig. 6 The left plot shows the paths planned using the LCSPEZ approximation as the CSPEZ constraint. Four different CSPEZ constraint values are shown $[0.01, 0.05, 0.25, 0.5]$. Mahalonobis distance values of one two and three are shown in red. The middle plot shows the paths planned using the QCSPEZ as the approximation and the right figure shows NNCSPEZ as the approximation.

So far, all results have compared different CSPEZ approximations as constraints for the path planning algorithm. To evaluate the benefit of using probabilistic constraints over a deterministic one, we compare against the CSBEZ formulation. The CSBEZ, being deterministic, ignores uncertainty in the pursuer's parameters, assuming the mean values are exact. In contrast, the CSPEZ approximations incorporate uncertainty in the pursuer's parameters, allowing for a tunable safety threshold. Notably, a CSPEZ constraint of 0.5 using the LCSPEZ approximation corresponds to enforcing that the evader lies outside the deterministic CSBEZ defined by the pursuer's mean parameters (i.e., $z(\mu_{\Theta_P}, \Theta_E) \leq 0$, Equation (25)). From Table 2, we observe that using the CSPEZ approximations instead of the deterministic CSBEZ results in significantly safer paths, with the added flexibility of tuning the constraint to balance risk and performance. This demonstrates that incorporating probabilistic constraints can significantly increase path safety by explicitly accounting for uncertainty, while incurring only a modest increase in path length.

VI. Conclusion

In this work, we introduced a framework for modeling probabilistic engagement zones to support path planning in adversarial environments with uncertainty. Our first contribution was an analytic solution to the Curve-Straight Basic Engagement Zone (CSBEZ), extending previous geometric models. We then proposed the Curve-Straight Probabilistic

Engagement Zone (CSPEZ), which estimates the probability of an evader being inside the true CSBEZ when the pursuer's parameters are uncertain.

To approximate the CSPEZ probability, we presented four methods: a Monte Carlo baseline (MCCSPEZ), a linearized model (LCSPEZ), a quadratic model (QCSPEZ), and a neural network model (NNCSPEZ). MCCSPEZ provides accurate results but is too slow and non-smooth for use in optimization. LCSPEZ and QCSPEZ offer analytic, differentiable approximations that require no pretraining. NNCSPEZ achieves the highest accuracy but depends on representative training data near the input configuration.

We showed how each method can be used as a constraint in trajectory optimization, enabling planners to generate paths that explicitly account for risk. This supports decision-making in uncertain, contested regions where deterministic methods may be unsafe. Our results showed that NNCSPEZ enabled the shortest, safest paths, while LCSPEZ and QCSPEZ remain useful when training data is unavailable.

This study focused on two-dimensional scenarios to enable tractable analysis, training, and visualization. However, the underlying framework can extend to 3D vehicle dynamics and more realistic engagement models. Future work will address uncertainty in the evader's state, incorporate multiple threats, and develop trajectory initialization strategies to improve convergence.

References

[1] Jun, M., and D'Andrea, R., "Path planning for unmanned aerial vehicles in uncertain and adversarial environments," *Cooperative control: models, applications and algorithms*, 2003, pp. 95–110.

[2] Peng, X., and Xu, D., "Intelligent online path planning for UAVs in adversarial environments," *International Journal of Advanced Robotic Systems*, Vol. 9, No. 1, 2012, p. 3.

[3] Weintraub, I. E., Von Moll, A., Carrizales, C. A., Hanlon, N., and Fuchs, Z. E., "An Optimal Engagement Zone Avoidance Scenario in 2-D," *AIAA SCITECH 2022 Forum*, 2022, p. 1587.

[4] Dillon, P. M., Zollars, M. D., Weintraub, I. E., and Von Moll, A., "Optimal Trajectories for Aircraft Avoidance of Multiple Weapon Engagement Zones," *Journal of Aerospace Information Systems*, Vol. 20, No. 8, 2023, pp. 520–525.

[5] Wolek, A., Weintraub, I. E., Von Moll, A., Casbeer, D., and Manyam, S. G., "Sampling-Based Risk-Aware Path Planning Around Dynamic Engagement Zones," *arXiv preprint arXiv:2403.05480*, 2024.

[6] Von Moll, A., and Weintraub, I., "Basic engagement zones," *Journal of Aerospace Information Systems*, 2024, pp. 1–7.

[7] Isaacs, R., "Differential Games, a Mathematical Theory with Applications to Optimization," , 1965.

[8] Stagg, G., and Peterson, C. K., "Probabilistic Weapon Engagement Zones," , 2025. Accepted for presentation at the *2025 American Control Conference (ACC)*.

[9] Dubins, L. E., "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, Vol. 79, No. 3, 1957, pp. 497–516.

[10] Chapman, T., Weintraub, I. E., Von Moll, A., and Garcia, E., "Engagement Zones for a Turn Constrained Pursuer," *arXiv preprint arXiv:2502.00364*, 2025.

[11] Friedman, A., "Stochastic differential games," *Journal of differential equations*, Vol. 11, No. 1, 1972, pp. 79–108.

[12] Patil, A., Zhou, Y., Fridovich-Keil, D., and Tanaka, T., "Risk-minimizing two-player zero-sum stochastic differential game via path integral control," *2023 62nd IEEE Conference on Decision and Control (CDC)*, IEEE, 2023, pp. 3095–3101.

[13] Li, D., and Cruz, J. B., "A two-player stochastic pursuit-evasion differential game," *2007 46th IEEE Conference on Decision and Control*, IEEE, 2007, pp. 4057–4062.

[14] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q., "JAX: composable transformations of Python+NumPy programs," , 2018. URL `http://github.com/google/jax`.

[15] Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M., "Flax: A neural network library and ecosystem for JAX," , 2024. URL `http://github.com/google/flax`.

[16] McKay, M. D., Beckman, R. J., and Conover, W. J., “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, Vol. 42, No. 1, 2000, pp. 55–61.

[17] Kingma, D. P., and Ba, J., “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[18] Cox, M. G., “The numerical evaluation of B-splines,” *IMA Journal of Applied mathematics*, Vol. 10, No. 2, 1972, pp. 134–149.

[19] Wächter, A., and Biegler, L. T., “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical programming*, Vol. 106, No. 1, 2006, pp. 25–57.