IFAC 2023

Upper and Lower Bounds for End-to-End Risks in Stochastic Robot Navigation

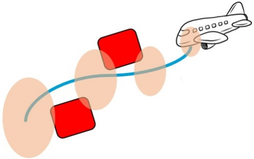
Apurva Patil   Takashi Tanaka

Outline

Outline

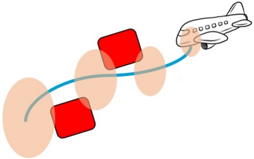
Background

- End-to-end risks in stochastic robot navigation
  - Characterize safety of the planned trajectories.
  - Plan risk optimal trajectories.



Background

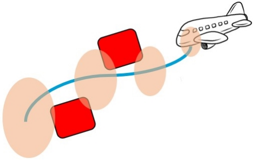
- End-to-end risks in stochastic robot navigation
  - Characterize safety of the planned trajectories.
  - Plan risk optimal trajectories.



- Discrete-time end-to-end risk:

$$P(E) = P(\bigvee_{t=0}^{T} E_t), \quad \text{where } E_t := \boldsymbol{x}_t^{\mathrm{sys}} \in \mathcal{X}^{\mathrm{obs}}.$$

Background



- End-to-end risks in stochastic robot navigation
  - Characterize safety of the planned trajectories.
  - Plan risk optimal trajectories.
- Discrete-time end-to-end risk:

$$P(E) = P(\bigvee_{t=0}^{T} E_t), \quad \text{where } E_t := \boldsymbol{x}_t^{\mathrm{sys}} \in \mathcal{X}^{\mathrm{obs}}.$$

- Exact evaluation of the end-to-end risk $P(E)$ is challenging
  - $\{E_t\}_{t=0}^{T}$ are statistically dependent events.
  - We derive upper and lower bounds using inequalities of **Hunter, Kounias, Fréchet, and Dawson** .

Outline

Existing Approaches

- Monte Carlo methods [1]
  - Computationally expensive and might underestimate the failure probability.

[1] (Janson, Schmerling, and Pavone 2018), (Blackmore, Ono, Bektassov, et al. 2010)

[2] (Strawser and B. Williams 2018)

[3] (Blackmore, Ono, and B. C. Williams 2011), (Ono et al. 2015)

Existing Approaches

- Monte Carlo methods [1]
  - Computationally expensive and might underestimate the failure probability.
- Approximation of law of total probability [2]

$$P\left(\bigvee_{t=0}^{T} E_t\right) \approx 1 - \prod_{t=0}^{T} P\left(E_t^c | E_{t-1}^c\right)$$

  - Can result in overly conservative estimates or can underestimate the failure probability.

[1] (Janson, Schmerling, and Pavone 2018), (Blackmore, Ono, Bektassov, et al. 2010)

[2] (Strawser and B. Williams 2018)

[3] (Blackmore, Ono, and B. C. Williams 2011), (Ono et al. 2015)

Existing Approaches

- Monte Carlo methods [1]
  - Computationally expensive and might underestimate the failure probability.
- Approximation of law of total probability [2]

$$P\left(\bigvee_{t=0}^{T} E_t\right) \approx 1 - \prod_{t=0}^{T} P\left(E_t^c \middle| E_{t-1}^c\right)$$

  - Can result in overly conservative estimates or can underestimate the failure probability.
- Boole's inequality [3]: $P\left(\bigvee_{t=0}^{T} E_t\right) \leq \sum_{t=0}^{T} P\left(E_t\right)$
  - Commonly used in risk-aware control problems due to its time-additivity.
  - Ignores the dependency among events $\{E_t\}_{t=0}^{T}$, can result in overly conservative estimates.

[1] (Janson, Schmerling, and Pavone 2018), (Blackmore, Ono, Bektassov, et al. 2010)

[2] (Strawser and B. Williams 2018)

[3] (Blackmore, Ono, and B. C. Williams 2011), (Ono et al. 2015)

Outline

Problem Formulation

- Planned trajectory
  - A finite sequences of positions $\{x_t^{\text{plan}} \in \mathcal{X}_{\text{free}}\}_{t=0}^{T}$ and control inputs $\{u_t^{\text{plan}} \in \mathbb{R}^m\}_{t=0}^{T-1}$ satisfying
$$x_{t+1}^{\text{plan}} = A_t x_t^{\text{plan}} + B_t u_t^{\text{plan}}.$$

Problem Formulation

- Planned trajectory
  - A finite sequences of positions $\{x_t^{\text{plan}} \in \mathcal{X}_{\text{free}}\}_{t=0}^{T}$ and control inputs $\{u_t^{\text{plan}} \in \mathbb{R}^m\}_{t=0}^{T-1}$ satisfying
  $$x_{t+1}^{\text{plan}} = A_t x_t^{\text{plan}} + B_t u_t^{\text{plan}}.$$
- Executed trajectory
  $$\boldsymbol{x}_{t+1}^{\text{sys}} = A_t \boldsymbol{x}_t^{\text{sys}} + B_t \boldsymbol{u}_t^{\text{sys}} + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim \mathcal{N}(0, W_t)$$

Problem Formulation

- Planned trajectory
  - A finite sequences of positions $\{x_t^{\mathrm{plan}} \in \mathcal{X}_{\mathrm{free}}\}_{t=0}^{T}$ and control inputs $\{u_t^{\mathrm{plan}} \in \mathbb{R}^m\}_{t=0}^{T-1}$ satisfying
$$x_{t+1}^{\mathrm{plan}} = A_t x_t^{\mathrm{plan}} + B_t u_t^{\mathrm{plan}}.$$
- Executed trajectory
$$\boldsymbol{x}_{t+1}^{\mathrm{sys}} = A_t \boldsymbol{x}_t^{\mathrm{sys}} + B_t \boldsymbol{u}_t^{\mathrm{sys}} + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim \mathcal{N}(0, W_t)$$
  - Defining $\boldsymbol{x}_t := \boldsymbol{x}_t^{\mathrm{sys}} - x_t^{\mathrm{plan}}$ and $\boldsymbol{u}_t := \boldsymbol{u}_t^{\mathrm{sys}} - u_t^{\mathrm{plan}}$, we get
$$\boldsymbol{x}_{t+1} = A_t \boldsymbol{x}_t + B_t \boldsymbol{u}_t + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim \mathcal{N}(0, W_t),$$

Problem Formulation

- Planned trajectory
  - A finite sequences of positions $\{x_t^{\text{plan}} \in \mathcal{X}_{\text{free}}\}_{t=0}^{T}$ and control inputs $\{u_t^{\text{plan}} \in \mathbb{R}^m\}_{t=0}^{T-1}$ satisfying
$$x_{t+1}^{\text{plan}} = A_t x_t^{\text{plan}} + B_t u_t^{\text{plan}}.$$
- Executed trajectory
$$\boldsymbol{x}_{t+1}^{\text{sys}} = A_t \boldsymbol{x}_t^{\text{sys}} + B_t \boldsymbol{u}_t^{\text{sys}} + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim \mathcal{N}(0, W_t)$$
  - Defining $\boldsymbol{x}_t := \boldsymbol{x}_t^{\text{sys}} - x_t^{\text{plan}}$ and $\boldsymbol{u}_t := \boldsymbol{u}_t^{\text{sys}} - u_t^{\text{plan}}$, we get
$$\boldsymbol{x}_{t+1} = A_t \boldsymbol{x}_t + B_t \boldsymbol{u}_t + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim \mathcal{N}(0, W_t),$$
- Sensor model
$$\boldsymbol{y}_t = C_t \boldsymbol{x}_t + \boldsymbol{v}_t, \qquad \boldsymbol{v}_t \sim \mathcal{N}(0, V_t)$$

Problem Formulation

- Planned trajectory
  - A finite sequences of positions $\{x_t^{\text{plan}} \in \mathcal{X}_{\text{free}}\}_{t=0}^{T}$ and control inputs $\{u_t^{\text{plan}} \in \mathbb{R}^m\}_{t=0}^{T-1}$ satisfying

$$x_{t+1}^{\text{plan}} = A_t x_t^{\text{plan}} + B_t u_t^{\text{plan}}.$$

- Executed trajectory

$$\boldsymbol{x}_{t+1}^{\text{sys}} = A_t \boldsymbol{x}_t^{\text{sys}} + B_t \boldsymbol{u}_t^{\text{sys}} + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim \mathcal{N}(0, W_t)$$

  - Defining $\boldsymbol{x}_t := \boldsymbol{x}_t^{\text{sys}} - x_t^{\text{plan}}$ and $\boldsymbol{u}_t := \boldsymbol{u}_t^{\text{sys}} - u_t^{\text{plan}}$, we get

$$\boldsymbol{x}_{t+1} = A_t \boldsymbol{x}_t + B_t \boldsymbol{u}_t + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim \mathcal{N}(0, W_t),$$

- Sensor model

$$\boldsymbol{y}_t = C_t \boldsymbol{x}_t + \boldsymbol{v}_t, \qquad \boldsymbol{v}_t \sim \mathcal{N}(0, V_t)$$

- Discrete-time end-to-end risk:

$$P(E) = P\left(\bigvee_{t=0}^{T} E_t\right), \quad \text{where } E_t := \boldsymbol{x}_t^{\text{sys}} \in \mathcal{X}^{\text{obs}}.$$

Outline

Probability Inequalities

Define:

$$p_t := P\left(E_t\right),\ p_{s,t} := P\left(E_s \bigwedge E_t\right),\ S_1 := \sum_{0\leq t\leq T} p_t,\ S_2 := \sum_{0\leq s<t\leq T} p_{s,t}.$$

Probability Inequalities

Define:

$$p_t := P\left(E_t\right), \ p_{s,t} := P\left(E_s \bigwedge E_t\right), \ S_1 := \sum_{0 \le t \le T} p_t, \ S_2 := \sum_{0 \le s < t \le T} p_{s,t}.$$

- Kwerel's upper bound: $P\left(E\right) \le S_1 - \frac{2}{T+1} S_2$.

Probability Inequalities

Define:

$$p_t := P\left(E_t\right),\ p_{s,t} := P\left(E_s \bigwedge E_t\right),\ S_1 := \sum_{0 \le t \le T} p_t,\ S_2 := \sum_{0 \le s < t \le T} p_{s,t}.$$

- Kwerel's upper bound: $P\left(E\right) \le S_1 - \frac{2}{T+1} S_2$.
- Kounias' upper bound: $P\left(E\right) \le S_1 - \max_{0 \le s \le T} \sum_{0 \le t \le T,\, t \ne s} p_{s,t}$.

Probability Inequalities

Define:

$$p_t := P\left(E_t\right),\ p_{s,t} := P\left(E_s \bigwedge E_t\right),\ S_1 := \sum_{0\leq t\leq T} p_t,\ S_2 := \sum_{0\leq s<t\leq T} p_{s,t}.$$

- Kwerel's upper bound: $P\left(E\right) \leq S_1 - \frac{2}{T+1} S_2$.
- Kounias' upper bound: $P\left(E\right) \leq S_1 - \max_{0\leq s\leq T} \sum_{0\leq t\leq T,\, t\neq s} p_{s,t}$.
- Hunter's upper bound: $P\left(E\right) \leq S_1 - \max_{\tau} \sum_{(s,t):e_{s,t}\in\tau} p_{s,t}$

where τ is a spanning tree of the graph whose vertices are $\{E_t\}_{t=0}^{T}$, with E_s and E_t joined by an edge $e_{s,t}$ if $E_s \bigwedge E_t \neq \emptyset$.

Probability Inequalities

Define:

$$p_t := P(E_t),\ p_{s,t} := P(E_s \bigwedge E_t),\ S_1 := \sum_{0 \le t \le T} p_t,\ S_2 := \sum_{0 \le s < t \le T} p_{s,t}.$$

- Kwerel's upper bound: $P(E) \le S_1 - \frac{2}{T+1} S_2$.
- Kounias' upper bound: $P(E) \le S_1 - \max_{0 \le s \le T} \sum_{0 \le t \le T,\, t \neq s} p_{s,t}$.
- Hunter's upper bound: $P(E) \le S_1 - \max_{\tau} \sum_{(s,t): e_{s,t} \in \tau} p_{s,t}$

where τ is a spanning tree of the graph whose vertices are $\{E_t\}_{t=0}^{T}$, with E_s and E_t joined by an edge $e_{s,t}$ if $E_s \bigwedge E_t \neq \emptyset$.

- Suboptimal Hunter's upper bound:

$$P(E) \le S_1 - \sum_{1 \le t \le T} p_{t-1,t}.$$

Probability Inequalities

- Fréchet's lower bound: $P(E) \geq \max_{0 \leq t \leq T} p_t$.

Probability Inequalities

- Fréchet's lower bound: $P(E) \geq \max_{0 \leq t \leq T} p_t$.
- Bonferroni's second-order lower bound: $P(E) \geq S_1 - S_2$.

Probability Inequalities

- Fréchet's lower bound: $P(E) \geq \max_{0 \leq t \leq T} p_t$.
- Bonferroni's second-order lower bound: $P(E) \geq S_1 - S_2$.
- Dawson and Sankoff's lower bound: if $S_1 > 0$,

$$P(E) \geq \frac{2}{k+1} S_1 - \frac{2}{k(k+1)} S_2,$$

where $k - 1$ is the integer part of $2S_2/S_1$.

Outline

Trajectory Tracking Controller

- Optimal controller for stochastic LQR [4]:

$$\boldsymbol{u}_t = F_t \hat{\boldsymbol{x}}_{t|t}$$

F_t are the LQR gains and $\hat{\boldsymbol{x}}_{t|t}$ are the state estimates obtained by the Kalman filter.

[4] (Stengel 1994)

Trajectory Tracking Controller

- Optimal controller for stochastic LQR [4]:

$$\boldsymbol{u}_t = F_t \hat{\boldsymbol{x}}_{t|t}$$

  F_t are the LQR gains and $\hat{\boldsymbol{x}}_{t|t}$ are the state estimates obtained by the Kalman filter.

- The *a priori* state estimates $\hat{\boldsymbol{x}}_{t|t-1}$ and the *a posteriori* state estimates $\hat{\boldsymbol{x}}_{t|t}$ evolve according to

$$\begin{aligned} \hat{\boldsymbol{x}}_{t|t-1} &= A_{t-1}\hat{\boldsymbol{x}}_{t-1|t-1} + B_{t-1}\boldsymbol{u}_{t-1}, \quad \hat{x}_{0|0} = 0 \\ \hat{\boldsymbol{x}}_{t|t} &= \hat{\boldsymbol{x}}_{t|t-1} + G_t\left(\boldsymbol{y}_t - C_t\hat{\boldsymbol{x}}_{t|t-1}\right), \end{aligned}$$

  where G_t are the Kalman gains.

[4] (Stengel 1994)

Closed-Loop Trajectory Distribution

The state deviation $\boldsymbol{x}_t$ and its *a priori* estimate $\hat{\boldsymbol{x}}_{t|t-1}$ jointly evolve as:

$$\overline{\boldsymbol{x}}_{t+1} = \overline{A}_t \overline{\boldsymbol{x}}_t + \overline{\boldsymbol{w}}_t, \qquad \overline{\boldsymbol{w}}_t \sim \mathcal{N}\left(0, \overline{W}_t\right)$$

where $\quad \overline{\boldsymbol{x}}_t := \begin{bmatrix} \boldsymbol{x}_t \\ \hat{\boldsymbol{x}}_{t|t-1} \end{bmatrix}, \quad \overline{\boldsymbol{w}}_t := \begin{bmatrix} B_t F_t G_t \boldsymbol{v}_t + \boldsymbol{w}_t \\ (A_t + B_t F_t) G_t \boldsymbol{v}_t \end{bmatrix},$

$$\overline{A}_t := \begin{bmatrix} A_t + B_t F_t G_t C_t & B_t F_t (I - G_t C_t) \\ (A_t + B_t F_t) G_t C_t & (A_t + B_t F_t)(I - G_t C_t) \end{bmatrix},$$

$$\overline{W}_t := \begin{bmatrix} B_t F_t G_t V_t G_t^\top F_t^\top B_t^\top + W_t & B_t F_t G_t V_t G_t^\top (A_t + B_t F_t)^T \\ (A_t + B_t F_t) G_t V_t^\top G_t^\top F_t^\top B_t^\top & (A_t + B_t F_t) G_t V_t G_t^\top (A_t + B_t F_t)^T \end{bmatrix}.$$

Closed-Loop Trajectory Distribution

Stacking $\overline{\boldsymbol{x}}_t$ for all time steps, we get

$$\overline{\boldsymbol{x}}^{\mathrm{traj}} = M\overline{x}_0 + N\overline{\boldsymbol{w}}^{\mathrm{traj}}, \qquad \overline{\boldsymbol{w}}^{\mathrm{traj}} \sim \mathcal{N}\left(0, \operatorname*{diag}_{0\le t\le T-1} \overline{W}_t\right),$$

where $\overline{\boldsymbol{x}}^{\mathrm{traj}} := \begin{bmatrix} \overline{\boldsymbol{x}}_0 \\ \overline{\boldsymbol{x}}_1 \\ \overline{\boldsymbol{x}}_2 \\ \vdots \\ \overline{\boldsymbol{x}}_T \end{bmatrix}, \overline{\boldsymbol{w}}^{\mathrm{traj}} := \begin{bmatrix} \overline{\boldsymbol{w}}_0 \\ \overline{\boldsymbol{w}}_1 \\ \overline{\boldsymbol{w}}_2 \\ \vdots \\ \overline{\boldsymbol{w}}_{T-1} \end{bmatrix}, M := \begin{bmatrix} I \\ \overline{A}_0 \\ \overline{A}_1\overline{A}_0 \\ \vdots \\ \overline{A}_{T-1}\ldots\overline{A}_0 \end{bmatrix},$

$$N := \begin{bmatrix} 0 & 0 & \ldots & 0 \\ I & 0 & \ldots & 0 \\ \overline{A}_1 & I & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \overline{A}_{T-1}\ldots\overline{A}_1 & \overline{A}_{T-1}\ldots\overline{A}_2 & \ldots & I \end{bmatrix}.$$

Closed-Loop Trajectory Distribution

- Assuming $\overline{x}_0 = 0$, the distribution of $\overline{\boldsymbol{x}}^{\mathrm{traj}}$ can be written as

$$\overline{\boldsymbol{x}}^{\mathrm{traj}} \sim \mathcal{N}\left(0, \overline{X}^{\mathrm{traj}}\right), \quad \overline{X}^{\mathrm{traj}} = N \left(\operatorname*{diag}_{0 \le t \le T-1} \overline{W}_t \right) N^\top .$$

Closed-Loop Trajectory Distribution

- Assuming $\overline{x}_0 = 0$, the distribution of $\overline{\boldsymbol{x}}^{\mathrm{traj}}$ can be written as

$$\overline{\boldsymbol{x}}^{\mathrm{traj}} \sim \mathcal{N}\left(0, \overline{X}^{\mathrm{traj}}\right), \quad \overline{X}^{\mathrm{traj}} = N\left(\operatorname*{diag}_{0 \le t \le T-1} \overline{W}_t\right) N^{\top}.$$

- Defining $\boldsymbol{x}^{\mathrm{traj}} := \begin{bmatrix} \boldsymbol{x}_0 & \boldsymbol{x}_1 & \dots & \boldsymbol{x}_T \end{bmatrix}^{\top}$, $\boldsymbol{x}^{\mathrm{traj}} \sim \mathcal{N}\left(0, X^{\mathrm{traj}}\right)$ where X^{traj} is obtained by marginalizing $\overline{X}^{\mathrm{traj}}$.

Closed-Loop Trajectory Distribution

- Assuming $\overline{x}_0 = 0$, the distribution of $\overline{\boldsymbol{x}}^{\mathrm{traj}}$ can be written as

$$\overline{\boldsymbol{x}}^{\mathrm{traj}} \sim \mathcal{N}\left(0, \overline{X}^{\mathrm{traj}}\right), \quad \overline{X}^{\mathrm{traj}} = N\left(\operatorname*{diag}_{0 \leq t \leq T-1} \overline{W}_t\right) N^\top.$$

- Defining $\boldsymbol{x}^{\mathrm{traj}} := \begin{bmatrix} \boldsymbol{x}_0 & \boldsymbol{x}_1 & \dots & \boldsymbol{x}_T \end{bmatrix}^\top$, $\boldsymbol{x}^{\mathrm{traj}} \sim \mathcal{N}\left(0, X^{\mathrm{traj}}\right)$ where X^{traj} is obtained by marginalizing $\overline{X}^{\mathrm{traj}}$.
- End-to-end probability of failure:

$$P(E) = 1 - \int_{\mathcal{X}^{\mathrm{free}}} \mathcal{N}\left(0, X^{\mathrm{traj}}\right) dx^{\mathrm{traj}}.$$

Closed-Loop Trajectory Distribution

- Assuming $\overline{x}_0 = 0$, the distribution of $\overline{\boldsymbol{x}}^{\mathrm{traj}}$ can be written as

$$\overline{\boldsymbol{x}}^{\mathrm{traj}} \sim \mathcal{N}\left(0, \overline{X}^{\mathrm{traj}}\right), \quad \overline{X}^{\mathrm{traj}} = N\left(\operatorname*{diag}_{0 \le t \le T-1} \overline{W}_t\right) N^\top.$$

- Defining $\boldsymbol{x}^{\mathrm{traj}} := \begin{bmatrix} \boldsymbol{x}_0 & \boldsymbol{x}_1 & \dots & \boldsymbol{x}_T \end{bmatrix}^\top$, $\boldsymbol{x}^{\mathrm{traj}} \sim \mathcal{N}\left(0, X^{\mathrm{traj}}\right)$ where X^{traj} is obtained by marginalizing $\overline{X}^{\mathrm{traj}}$.
- End-to-end probability of failure:

$$P(E) = 1 - \int_{\mathcal{X}^{\mathrm{free}}} \mathcal{N}\left(0, X^{\mathrm{traj}}\right) dx^{\mathrm{traj}}.$$

- Evaluating the above integral on a high dimensional non-convex region is computationally expensive.

Outline

Computation of the Bounds

- The main task in computing the probability bounds is to compute the univariate probabilities p_t, and bivariate joint probabilities $p_{s,t}$.

Computation of the Bounds

- The main task in computing the probability bounds is to compute the univariate probabilities p_t, and bivariate joint probabilities $p_{s,t}$.

$$\boldsymbol{x}_t^{\mathrm{sys}} \sim \mathcal{N}\left(x_t^{\mathrm{plan}}, X_t\right)$$
$$\begin{bmatrix}\boldsymbol{x}_s^{\mathrm{sys}} & \boldsymbol{x}_t^{\mathrm{sys}}\end{bmatrix}^\top \sim \mathcal{N}\left(\begin{bmatrix}x_s^{\mathrm{plan}} x_t^{\mathrm{plan}}\end{bmatrix}^\top, X_{st}\right)$$

where X_t and X_{st} are obtained by marginalizing X^{traj}.

Computation of the Bounds

- The main task in computing the probability bounds is to compute the univariate probabilities p_t, and bivariate joint probabilities $p_{s,t}$.

$$\boldsymbol{x}_t^{\text{sys}} \sim \mathcal{N}\left(x_t^{\text{plan}}, X_t\right)$$
$$\begin{bmatrix}\boldsymbol{x}_s^{\text{sys}} & \boldsymbol{x}_t^{\text{sys}}\end{bmatrix}^\top \sim \mathcal{N}\left(\begin{bmatrix}x_s^{\text{plan}} x_t^{\text{plan}}\end{bmatrix}^\top, X_{st}\right)$$

where X_t and X_{st} are obtained by marginalizing X^{traj}.

$$p_t = \int_{\mathcal{X}^{\text{obs}}} \mathcal{N}\left(x_t^{\text{plan}}, X_t\right) dx_t^{\text{sys}}$$
$$p_{s,t} = \int_{\mathcal{X}^{\text{obs}}} \int_{\mathcal{X}^{\text{obs}}} \mathcal{N}\left(\begin{bmatrix}x_s^{\text{plan}} x_t^{\text{plan}}\end{bmatrix}^\top, X_{st}\right) dx_s^{\text{sys}} dx_t^{\text{sys}}$$

Computation of the Bounds

- The main task in computing the probability bounds is to compute the univariate probabilities p_t, and bivariate joint probabilities $p_{s,t}$.

$$\boldsymbol{x}_t^{\text{sys}} \sim \mathcal{N}\left(x_t^{\text{plan}}, X_t\right)$$
$$\begin{bmatrix}\boldsymbol{x}_s^{\text{sys}} & \boldsymbol{x}_t^{\text{sys}}\end{bmatrix}^\top \sim \mathcal{N}\left(\begin{bmatrix}x_s^{\text{plan}} x_t^{\text{plan}}\end{bmatrix}^\top, X_{st}\right)$$

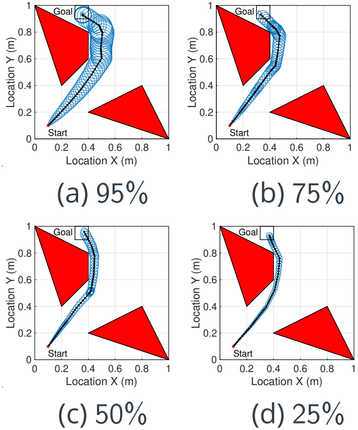
  where X_t and X_{st} are obtained by marginalizing X^{traj}.

$$p_t = \int_{\mathcal{X}^{\text{obs}}} \mathcal{N}\left(x_t^{\text{plan}}, X_t\right) dx_t^{\text{sys}}$$
$$p_{s,t} = \int_{\mathcal{X}^{\text{obs}}} \int_{\mathcal{X}^{\text{obs}}} \mathcal{N}\left(\begin{bmatrix}x_s^{\text{plan}} x_t^{\text{plan}}\end{bmatrix}^\top, X_{st}\right) dx_s^{\text{sys}} dx_t^{\text{sys}}$$

- In this work, we assume that the obstacles are convex polytopes and develop a formulation to compute p_t and $p_{s,t}$, numerically. Please refer to the paper for the details.

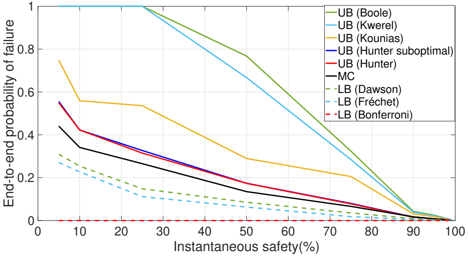
Outline

Simulation Results



Trajectories planned with the instantaneous safety

Pedram et al. 2021.



Simulation Results

Comparison of different risk estimates over 100 trajectories

Estimates	Mean Abs. Error	Avg. Time [s]
Monte Carlo	0	46.83
Upper bounds		
Boole	40.59	0.01
Kwerel	38.15	2.43
Kounias	13.34	2.42
Hunter	8.63	2.41
Hunter suboptimal	10.25	0.18
Lower bounds		
Bonferroni	54.88	2.44
Fréchet	40.08	0.01
Dawson	16.74	2.44

Outline

Higher-Order Probability Bounds

- A third-order upper bound computed using the Cherry Trees approach [5]
- Bounds higher than order 3 can be computed using the linear programming algorithms [6]
- Bonferroni's k^{th}-order bound:
  $P(E) \leq S_1 - S_2 + S_3 - \ldots - S_{k-1} + S_k$ if $k < T$ is odd
  $P(E) \geq S_1 - S_2 + S_3 - \ldots + S_{k-1} - S_k$ if $k < T$ is even
  where $S_r = \sum_{0 \leq j_1 < \ldots < j_r \leq T} P\left(E_{j_1} \bigwedge \ldots \bigwedge E_{j_r}\right)$.

[5] (Bukszár and Prekopa 2001)

[6] (Prékopa 1988)

Outline

Summary

- We presented an analytical method to compute upper and lower bounds for the discrete-time collision probability of motion plans.
- The bounds are computed using the closed-loop system trajectory distribution of the system without making any independence assumptions on the events of collision at different time steps.
- Numerical validation demonstrates that our bounds are less conservative than the Boole's bound commonly used in the literature and are cheaper in computation than the Monte Carlo method.

Check out the paper for more details and results.

Summary

- We presented an analytical method to compute upper and lower bounds for the discrete-time collision probability of motion plans.
- The bounds are computed using the closed-loop system trajectory distribution of the system without making any independence assumptions on the events of collision at different time steps.
- Numerical validation demonstrates that our bounds are less conservative than the Boole's bound commonly used in the literature and are cheaper in computation than the Monte Carlo method.

Check out the paper for more details and results.

- **Future work**: Incorporation of the presented bounds in the planning phase to generate risk-optimal trajectories.