IEEE Conference on Decision and Control 2022

Chance-Constrained Stochastic Optimal Control via Path Integral and Finite Difference Methods

Apurva Patil Alfredo Duarte Aislinn Smith
Prof. Fabrizio Bisetti Prof. Takashi Tanaka

Outline

Outline

Background

Chance-constrained stochastic optimal control problem

The drunken spider problem[1]

- A drunken spider wants to take the shortest path to home.
- Probability of falling into the water should be small $\rightarrow$ chance constraint

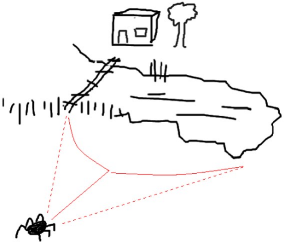


Image credit [1]

[1] Kappen "Path integrals and symmetry breaking for optimal control theory", Journal of statistical mechanics: theory and experiment, 2005, no. 11

System Model

- State equation: Control-affine n-dimensional Ito process:

$$d\boldsymbol{x}_t = f(\boldsymbol{x}_t)dt + G(\boldsymbol{x}_t)\boldsymbol{u}_t dt + \Sigma(\boldsymbol{x}_t)d\boldsymbol{w}_t, \;\; t \in [t_0, T]$$
$$\boldsymbol{x}_{t_0} = x_0$$

  where $\boldsymbol{w}_t$ is an n-dimensional Brownian motion.
- Safe region: $\mathcal{X}_s$, boundary: $\partial\mathcal{X}_s$
- Probability of failure:

$$P_{\text{fail}} = P_{x_0,t_0}\left(\bigvee_{t\in(t_0,T]} \boldsymbol{x}_t \notin \mathcal{X}_s\right).$$

System Model

- Exit time (final time):

$$\boldsymbol{t}_f = \begin{cases} T & \text{if } \boldsymbol{x}_t \in \mathcal{X}_s, \forall t \in (t_0, T) \\ \inf\{t \in (t_0, T) : \boldsymbol{x}_t \notin \mathcal{X}_s\} & \text{otherwise} \end{cases}$$

- Cost function: We assume cost function is quadratic in $\boldsymbol{u}_t$.

$$C(x_0, t_0, u(\cdot)) :=$$

$$\mathbb{E}\left[\underbrace{\int_{t_0}^{\boldsymbol{t}_f}\left(V(\boldsymbol{x}_t, t) + \frac{1}{2}\boldsymbol{u}_t^\top R(\boldsymbol{x}_t, t)\boldsymbol{u}_t\right)dt}_{\substack{\text{Running cost} \\ \text{(e.g., travel distance)}}} + \underbrace{\psi(\boldsymbol{x}_{\boldsymbol{t}_f}) \cdot \mathbb{1}(\boldsymbol{x}_{\boldsymbol{t}_f} \in \mathcal{X}_s)}_{\substack{\text{Terminal cost} \\ \text{(e.g., distance from home)}}}\right]$$

Risk-constrained Stochastic Optimal Control (SOC) Problem

$$\min_{\boldsymbol{u}} \mathbb{E}_{x_0,t_0}\left[\int_{t_0}^{\boldsymbol{t}_f}\left(V(\boldsymbol{x}_t,t)+\frac{1}{2}\boldsymbol{u}_t^\top R(\boldsymbol{x}_t,t)\boldsymbol{u}_t\right)dt+\psi(\boldsymbol{x}_{\boldsymbol{t}_f})\cdot\mathbb{1}(\boldsymbol{x}_{\boldsymbol{t}_f}\in\mathcal{X}_s)\right]$$

$$\text{s.t.}\quad d\boldsymbol{x}_t = f(\boldsymbol{x}_t)dt + G(\boldsymbol{x}_t)\boldsymbol{u}_t dt + \Sigma(\boldsymbol{x}_t)d\boldsymbol{w}_t, \quad \boldsymbol{x}_{t_0} = x_0,$$

$$P_{x_0,t_0}\left(\bigvee_{t\in(t_0,T]} \boldsymbol{x}_t \notin \mathcal{X}_s\right) < \Delta \quad \text{(Chance constraint)}$$

- This is a variable end-time problem – there is no cost after the system fails.
- We consider end-to-end risk (not pointwise risk).

Outline

Related Work (Non-Exhaustive)

Discrete-time approaches:

- Iterative risk allocation scheme with Boole's bound [Ono et al. 2008]: Boole's bound is used to approximate the joint chance constraint and the user-specified risk "budget" is allocated optimally between timesteps.
- Lagrangian relaxation with Boole's bound [Ono et al. 2015]: Joint chance constraint is approximated using Boole's inequality, and Lagrangian relaxation is used to obtain an unconstrained optimal control problem which is solved using dynamic programming.
- Sampling-based approaches [Blackmore et al. 2010]
- Reinforcement learning [Huang et al. 2021]

Related Work (Non-Exhaustive)

Continuous-time approaches:

- Generalized polynomial chaos [Nakka et al. 2019]: A stochastic optimal control problem is converted to a deterministic optimal control problem using generalized polynomial chaos expansion and then solved using sequential convex programming.
- Stochastic Control Barrier Functions [Santoyo et al. 2019]: Stochastic control barrier functions are used to derive sufficient conditions on the control input that bound the probability of failure.
- Reflection principle [Ariu et al. 2017]: Reflection principle of Brownian motion along with Boole's inequality is used to bound the failure probability in continuous-time.

Outline

Risk-Constrained SOC $\rightarrow$ Risk-Minimizing SOC

- Since a hard chance-constraint is difficult to deal with, we introduce a new objective function with a soft chance-constraint:

$$\widehat{C}(x_0, t_0, u(\cdot)) = C(x_0, t_0, u(\cdot)) + \eta P_{x_0, t_0}\left(\bigvee_{t \in (t_0, T]} \boldsymbol{x}_t \notin \mathcal{X}_s\right)$$

  where $\eta > 0$ is the Lagrange multiplier.

- Notice that P_{fail} can be expressed in terms of the exit time as

$$P_{x_0, t_0}\left(\bigvee_{t \in (t_0, T]} \boldsymbol{x}_t \notin \mathcal{X}_s\right) = \mathbb{E}_{x_0, t_0}\left[\mathbb{1}(\boldsymbol{x}_{\boldsymbol{t}_f} \in \partial \mathcal{X}_s)\right]$$

Risk-Constrained SOC $\rightarrow$ Risk-Minimizing SOC

- Introduce a new terminal cost function $\phi : \overline{\mathcal{X}}_s \rightarrow \mathbb{R}$ as

$$\phi(x) := \underbrace{\psi(x) \cdot \mathbb{1}(x \in \mathcal{X}_s)}_{\text{Terminal cost when } \boldsymbol{t}_f = T} + \underbrace{\eta \cdot \mathbb{1}(x \in \partial \mathcal{X}_s)}_{\text{Terminal cost when } \boldsymbol{t}_f < T}$$

- P_{fail} is now absorbed in the new terminal cost function ϕ as

$$\widehat{C}(x_0, t_0, u(\cdot)) = \mathbb{E}_{x_0, t_0}\left[\int_{t_0}^{\boldsymbol{t}_f} \left(V(\boldsymbol{x}_t, t) + \frac{1}{2}\boldsymbol{u}_t^\top R(\boldsymbol{x}_t, t)\boldsymbol{u}_t\right) dt + \phi(\boldsymbol{x}_{\boldsymbol{t}_f})\right].$$

Risk-Minimizing SOC Problem

$$\min_{\boldsymbol{u}} \quad \mathbb{E}_{x_0,t_0}\left[\int_{t_0}^{\boldsymbol{t}_f}\left(V(\boldsymbol{x}_t, t) + \frac{1}{2}\boldsymbol{u}_t^\top R(\boldsymbol{x}_t, t)\boldsymbol{u}_t\right)dt + \phi(\boldsymbol{x}_{\boldsymbol{t}_f})\right]$$

$$\text{s.t.} \quad d\boldsymbol{x}_t = f(\boldsymbol{x}_t)dt + G(\boldsymbol{x}_t)\boldsymbol{u}_t dt + \Sigma(\boldsymbol{x}_t)d\boldsymbol{w}_t, \quad \boldsymbol{x}_{t_0} = x_0$$

- This problem has a time-additive Bellman structure suitable for dynamic programming.
- Next we show that solving this problem (like many other optimal control problem to which dynamic programming is applicable) is equivalent to solving a Hamilton Jacobi Bellman (HJB) partial differential equation (PDE).
- The regularizer $\eta > 0$ for the chance-constraint will appear as a Dirichlet boundary condition for the HJB PDE.

Outline

Value Function

- For each state-time pair (x, t), define the value function for the risk-minimizing SOC as

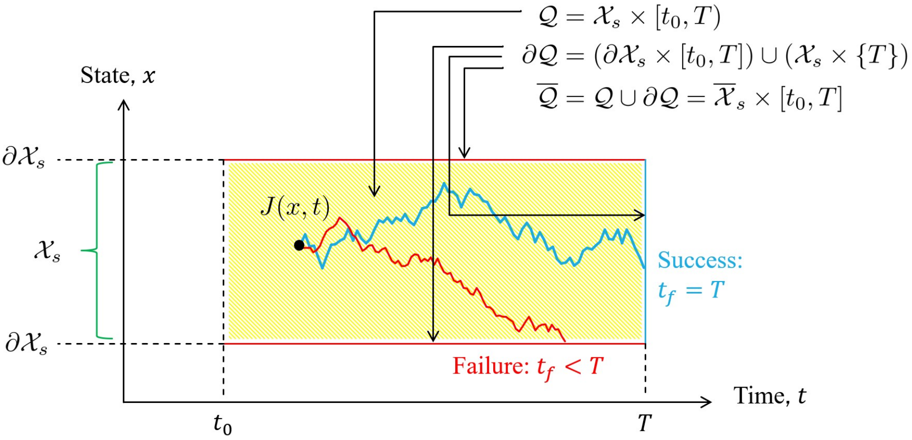
$$J(x,t) = \min_{u(\cdot)} \widehat{C}(x,t,u(\cdot))$$

where

$$\widehat{C}(x,t,u(\cdot)) = \mathbb{E}_{x,t}\left[\int_t^{\boldsymbol{t}_f} \left(V(\boldsymbol{x}_s, s) + \frac{1}{2}\boldsymbol{u}_s^\top R(\boldsymbol{x}_s, s)\boldsymbol{u}_s \right) ds + \phi(\boldsymbol{x}_{\boldsymbol{t}_f})\right].$$

- We are interested in finding a PDE that $J(x, t)$ will satisfy in the domain $(x, t) \in \mathcal{Q}$, where $\mathcal{Q}$ is ...

Domain of the HJB PDE



Verification Theorem

Theorem:

Suppose there exists a function $J : \overline{\mathcal{Q}} \to \mathbb{R}$ such that (i) $J(x,t)$ is continuously differentiable in t and twice continuously differentiable in x in $\mathcal{Q}$, and (ii) $J(x,t)$ solves the HJB PDE:

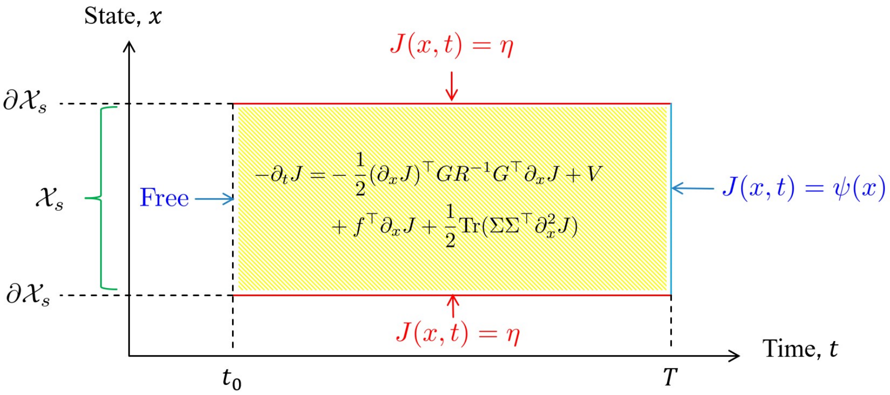
$$-\partial_t J = -\frac{1}{2}(\partial_x J)^\top G R^{-1} G^\top \partial_x J + V + f^\top \partial_x J + \frac{1}{2}\mathrm{Tr}(\Sigma\Sigma^\top \partial_x^2 J), \quad \forall (x,t) \in \mathcal{Q}$$

$$\lim_{(x,t)\to(y,t)} J(x,t) = \phi(y), \quad \forall (y,t) \in \partial\mathcal{Q} \quad \text{(Dirichlet BC)}$$

Then,

1. $J(x,t)$ is the value function for the risk-minimizing SOC, i.e., $J(x,t) = \min_{u(\cdot)} \widehat{C}(x,t,u(\cdot))$;
2. The optimal control is given by $u^*(x,t) = -R^{-1}(x,t) G^\top(x,t) \partial_x J(x,t)$.

HJB PDE: Boundary Conditions



Verification Theorem: Sketch of Proof

By **Dynkin's formula**, we have

$\mathbb{E}_{x,t}\left[J(\boldsymbol{x}_{\boldsymbol{t}_f}, \boldsymbol{t}_f)\right] = J(x,t) + \mathbb{E}_{x,t}\left[\int_t^{\boldsymbol{t}_f} dJ(\boldsymbol{x}_s, s)\right].$

- $J(\boldsymbol{x}_{\boldsymbol{t}_f}, \boldsymbol{t}_f) = \phi(\boldsymbol{x}_{\boldsymbol{t}_f})$ by boundary condition.
- Using **Ito's formula**, dJ can be computed as

$$dJ(\boldsymbol{x}_s, s) = (\partial_t J)ds + (f + Gu)^\top(\partial_x J)ds + (\Sigma d\boldsymbol{w}_s)^\top \partial_x J + \frac{1}{2}\mathrm{Tr}(\Sigma\Sigma^\top \partial_x^2 J)ds$$

Therefore

$$J(x,t) = \mathbb{E}_{x,t}[\phi(\boldsymbol{x}_{\boldsymbol{t}_f})] - \mathbb{E}_{x,t}\left[\int_f^{\boldsymbol{t}_f}\left(\partial_t J + (f + Gu)^\top(\partial_x J) + \frac{1}{2}\mathrm{Tr}(\Sigma\Sigma^\top \partial_x^2 J)\right)ds\right]$$

Verification Theorem: Sketch of Proof

Notice that the RHS of the HJB PDE can be written as

$$\begin{aligned} -\partial_t J &= -\tfrac{1}{2}(\partial_x J)^\top G R^{-1} G^\top \partial_x J + V + f^\top \partial_x J + \tfrac{1}{2}\mathrm{Tr}(\Sigma\Sigma^\top \partial_x^2 J) \\ &= \min_u \left[\tfrac{1}{2} u^\top R u + V + (f + Gu)^\top \partial_x J + \tfrac{1}{2}\mathrm{Tr}(\Sigma\Sigma^\top \partial_x^2 J) \right] \end{aligned}$$

Therefore, $-\partial_t J \leq \frac{1}{2} u^\top R u + V + (f + Gu)^\top \partial_x J + \frac{1}{2}\mathrm{Tr}(\Sigma\Sigma^\top \partial_x^2 J)$ holds for any u (equality holds iff $u = -R^{-1} G^\top \partial_x J$).
Substituting this result to the last equation in the previous slide, we obtain

$$J(x,t) \leq \mathbb{E}_{x,t}[\phi(\boldsymbol{x}_{\boldsymbol{t}_f})] + \mathbb{E}_{x,t}\left[\int_t^{\boldsymbol{t}_f} \left(\frac{1}{2} u^\top R u + V\right)\right] = \widehat{C}(x,t,u(\cdot)).$$

Thus, we have shown that $J(x,t) = \min_u \widehat{C}(x,t,u(\cdot))$.

Outline

Risk Estimation

- Risk estimation of a given policy is a special case of our risk-minimizing SOC problem.

Risk Estimation

- Risk estimation of a given policy is a special case of our risk-minimizing SOC problem.
- Suppose, $\psi(x) \equiv 0,\ R(x,t) \equiv 0,\ V(x,t) \equiv 0,\ \eta = 1$, then

$$\phi(x) = \mathbb{1}_{x \in \partial \mathcal{X}_s},\ \widehat{C}\,(x_0, t_0, u(\cdot)) = \mathbb{E}_{x_0, t_0}\big[\mathbb{1}_{\boldsymbol{x}(\boldsymbol{t}_f) \in \partial \mathcal{X}_s}\big] = P_{\text{fail}}.$$

Risk Estimation

- Risk estimation of a given policy is a special case of our risk-minimizing SOC problem.
- Suppose, $\psi(x) \equiv 0,\ R(x,t) \equiv 0,\ V(x,t) \equiv 0,\ \eta = 1$, then

$$\phi(x) = \mathbb{1}_{x \in \partial \mathcal{X}_s},\ \widehat{C}\,(x_0, t_0, u(\cdot)) = \mathbb{E}_{x_0,t_0}\left[\mathbb{1}_{\boldsymbol{x}(\boldsymbol{t}_f) \in \partial \mathcal{X}_s}\right] = P_{\text{fail}}.$$

Corollary:

Suppose there exists a continuous function $J : \overline{\mathcal{Q}} \to \mathbb{R}$ that solves the following PDE:

$$\begin{cases} -\partial_t J = (f + gu)^T \partial_x J + \frac{1}{2} \mathrm{Tr}\left(\sigma \sigma^T \partial_x^2 J\right), & (x,t) \in \mathcal{Q}, \\ J(x,t) = \phi(x), & (x,t) \in \partial \mathcal{Q}. \end{cases}$$

Then, $P_{fail} = J(x_0, t_0)$.

Outline

Finite Difference Method (FDM)

One of the most popular approaches to solve PDEs.
Computational domain $\overline{\mathcal{Q}}$ is discretized into a finite number of grid points and the solution to the PDE is sought at these locations.
Example: Drunken spider (velocity input model)

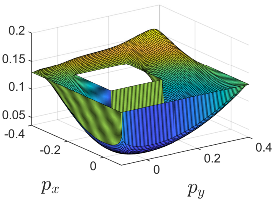
- State $(\boldsymbol{p}_x, \boldsymbol{p}_y)$: Position of the spider

$$d\boldsymbol{p}_x = -k_x \boldsymbol{p}_x dt + u_x dt + \sigma d\boldsymbol{w}_x$$
$$d\boldsymbol{p}_y = -k_y \boldsymbol{p}_y dt + u_y dt + \sigma d\boldsymbol{w}_y$$

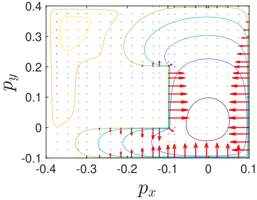
- Running cost: $V(x_t) = \boldsymbol{p}_{x,t}^2 + \boldsymbol{p}_{y,t}^2$
- Terminal cost: $\psi(x_T) = \boldsymbol{p}_{x,T}^2 + \boldsymbol{p}_{y,T}^2$

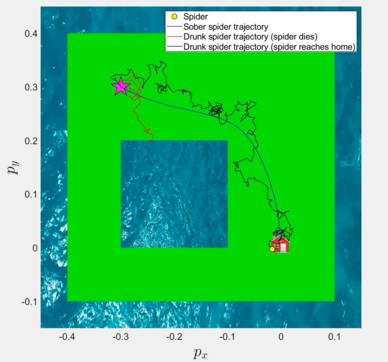
Finite Difference Method: Example

$J(x, t_0)$



$u^*(x, t_0)$





Limitations of FDM

- Curse of dimensionality – Gridding is prohibitive for problems with higher dimensions.
- HJB equation for our SOC must be solved backward-in-time, which is inconvenient for real-time implementations.
- FDM computes the global solution $J(x, t)$ over the entire domain $\overline{\mathcal{Q}}$ even if the majority of the state-time pairs (x, t) will never be visited by the actual system.

Limitations of FDM

- Curse of dimensionality – Gridding is prohibitive for problems with higher dimensions.
- HJB equation for our SOC must be solved backward-in-time, which is inconvenient for real-time implementations.
- FDM computes the global solution $J(x,t)$ over the entire domain $\overline{\mathcal{Q}}$ even if the majority of the state-time pairs (x,t) will never be visited by the actual system.

We want an algorithm to compute $J(x,t)$ and $u^*(x,t) = -R^{-1}G^\top \partial_x J(x,t)$ on-the-fly for the current state-time pair (x,t).

Path Integral Method

- Computes the solution $J(x, t)$ of the HJB equation at an arbitrary (x, t) using forward-in-time Monte-Carlo simulations of system trajectories.
- $J(x, t)$ is computed by the empirical mean of the path cost ("path integral") of the simulated sample paths.
- Optimal control $u^*(x, t)$ can also be computed by Monte-Carlo simulation without solving HJB equation backward in time.
- Massively parallelizable on GPUs.
- Path integral method is considered less susceptible to curse of dimensionality

Path Integral Control: Theorem

Theorem:

Suppose there exists a constant $\lambda > 0$ such that

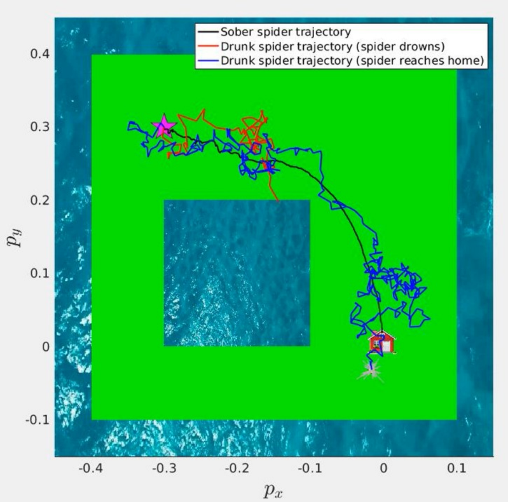
$$\Sigma\Sigma^T = \lambda G R^{-1} G^T.$$

Then, for each $(x, t) \in \mathcal{Q}$, we have

$$J(x,t) = -\lambda \log \mathbb{E}\left[\exp\left(-\frac{1}{\lambda}\int_t^{\boldsymbol{t}_f} V(\boldsymbol{x}_t)dt - \frac{1}{\lambda}\phi(\boldsymbol{x}_t)\right)\right]$$

where $\mathbb{E}[\cdot]$ is with respect to the distribution generated by the "uncontrolled" dynamics $d\boldsymbol{x}_t = fdt + \Sigma d\boldsymbol{w}_t$.
Moreover, $u^*(x,t) = -R^{-1}(x,t)G^\top(x,t)\partial_x J(x,t)$.

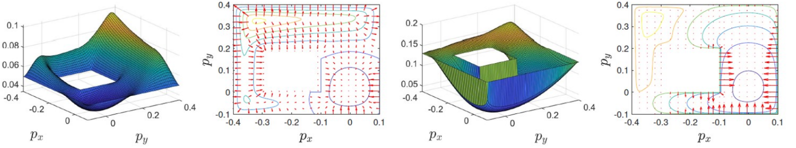
Path Integral: Example



Outline

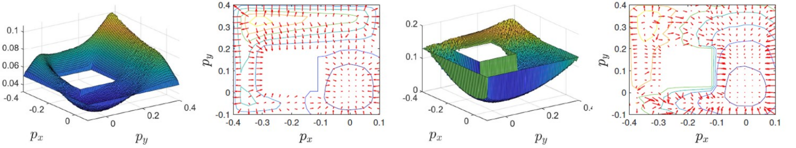
FDM vs Path Integral Method

FDM



(a) $J(x, t_0)$ for $\eta = 0.05$ (b) $u^*(x, t_0)$ for $\eta = 0.05$ (c) $J(x, t_0)$ for $\eta = 0.13$ (d) $u^*(x, t_0)$ for $\eta = 0.13$

Path Integral Method

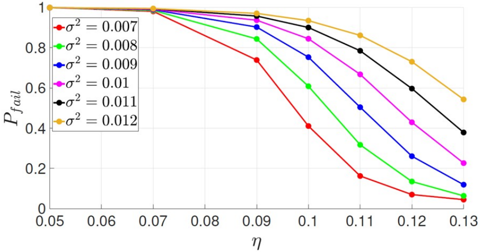


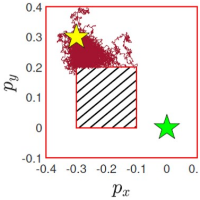
(e) $J(x, t_0)$ for $\eta = 0.05$ (f) $u^*(x, t_0)$ for $\eta = 0.05$ (g) $J(x, t_0)$ for $\eta = 0.13$ (h) $u^*(x, t_0)$ for $\eta = 0.13$

Outline

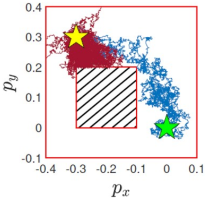
P_{fail} vs η: Simulation study

Blue: Safe trajectories
Red: Failed trajectories

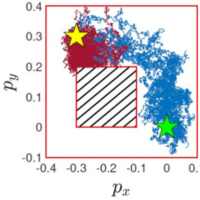




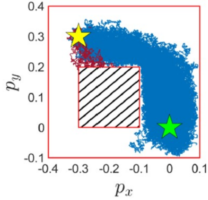
(a) $\eta = 0.05$



(b) $\eta = 0.07$

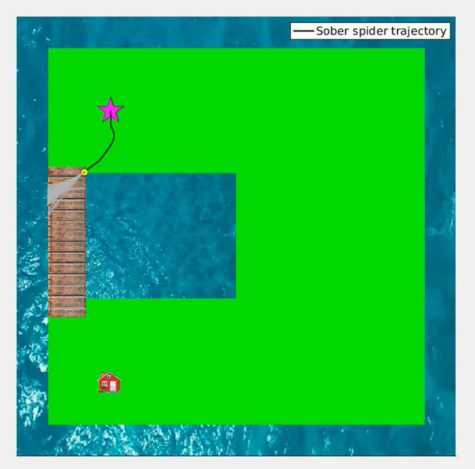


(c) $\eta = 0.09$



(d) $\eta = 0.13$

Path Integral Example



Outline

Summary

- We presented an approach to solve a risk-constrained SOC problem for nonlinear system dynamics and cost functions. We considered continuous-time, end-to-end risk without any conservative approximation.
- Risk-constrained control problem is formulated using the notion of exit time and converted it to a risk-minimizing SOC problem which has a time-additive cost function
- We showed that risk-minimizing control synthesis is equivalent to solving an HJB PDE with Dirichlet boundary condition which can be tuned appropriately to achieve a desired level of safety.
- The proposed risk-minimizing control problem can be viewed as a generalization of the risk-estimation problem.
- Compared simulation results of FDM and path integral

Outline

Future Work

- Connection between Δ and η (hard vs soft chance constraints)

---

[1] Patil et al., "Risk-Minimizing Two-Player Zero-Sum Stochastic Differential Game via Path Integral Control", submitted to ACC 2023

[2] Yoon et al., "Sampling complexity of path integral methods for trajectory optimization", ACC 2022

Future Work

- Connection between Δ and η (hard vs soft chance constraints)
- Chance-constrained stochastic games[1]

[1] Patil et al., "Risk-Minimizing Two-Player Zero-Sum Stochastic Differential Game via Path Integral Control", submitted to ACC 2023

[2] Yoon et al., "Sampling complexity of path integral methods for trajectory optimization", ACC 2022

Future Work

- Connection between Δ and η (hard vs soft chance constraints)
- Chance-constrained stochastic games[1]
- Partially observable systems, mean-field games

[1] Patil et al., "Risk-Minimizing Two-Player Zero-Sum Stochastic Differential Game via Path Integral Control", submitted to ACC 2023

[2] Yoon et al., "Sampling complexity of path integral methods for trajectory optimization", ACC 2022

Future Work

- Connection between Δ and η (hard vs soft chance constraints)
- Chance-constrained stochastic games[1]
- Partially observable systems, mean-field games
- When exactly is path integral control better than alternatives (e.g., DDP, DRL)?

[1] Patil et al., "Risk-Minimizing Two-Player Zero-Sum Stochastic Differential Game via Path Integral Control", submitted to ACC 2023

[2] Yoon et al., "Sampling complexity of path integral methods for trajectory optimization", ACC 2022

Future Work

- Connection between Δ and η (hard vs soft chance constraints)
- Chance-constrained stochastic games[1]
- Partially observable systems, mean-field games
- When exactly is path integral control better than alternatives (e.g., DDP, DRL)?
- Sample complexity analysis of path integral control[2]

[1] Patil et al., "Risk-Minimizing Two-Player Zero-Sum Stochastic Differential Game via Path Integral Control", submitted to ACC 2023

[2] Yoon et al., "Sampling complexity of path integral methods for trajectory optimization", ACC 2022

Thank you!