Control Optimization for Uncertain Systems via the Koopman Operator

Jonathan Rogers, Andrew Leonard, Joey Meyers
Georgia Institute of Technology

Adam Gerlach
US Air Force Research Laboratory

Chris Rackauckas
Massachusetts Institute of Technology

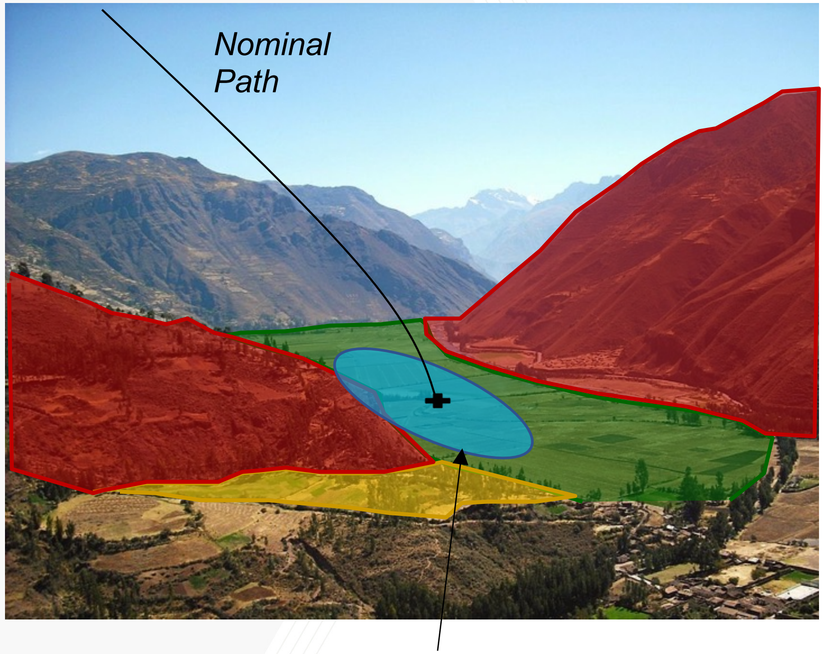
Optimal Decision-Making Under Uncertainty

- The need to make decisions under uncertainty arises often in engineering and scientific applications

Optimal Decision-Making Under Uncertainty

- The need to make decisions under uncertainty arises often in engineering and scientific applications

Airdrop Package Delivery Using Ballistic Parachute



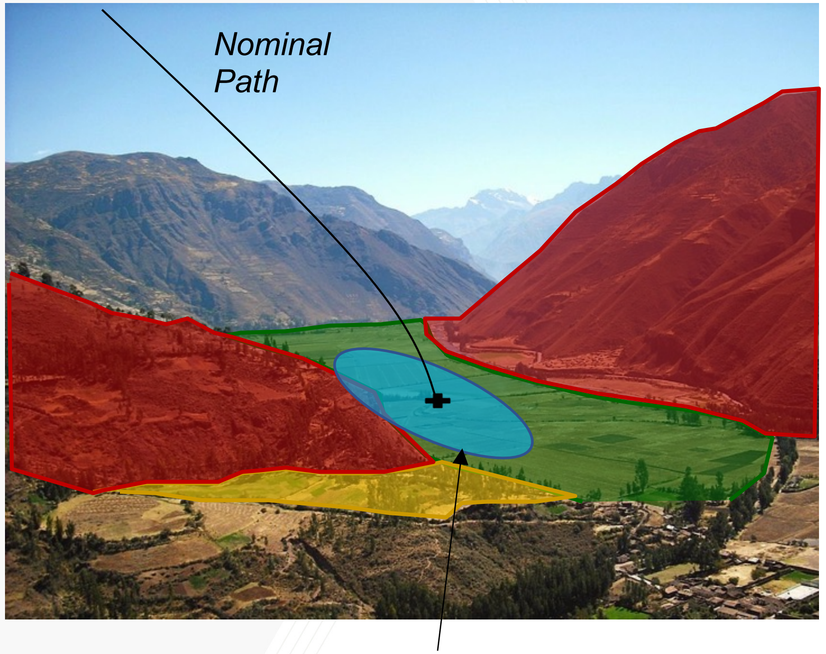
Predicted Dispersion

Q: Where should the package be dropped?

Optimal Decision-Making Under Uncertainty

- The need to make decisions under uncertainty arises often in engineering and scientific applications

Airdrop Package Delivery Using Ballistic Parachute

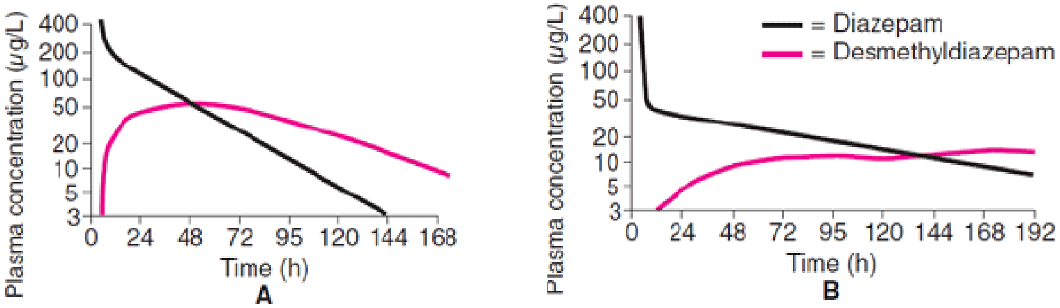


Predicted Dispersion

Q: Where should the package be dropped?

Pharmacokinetics

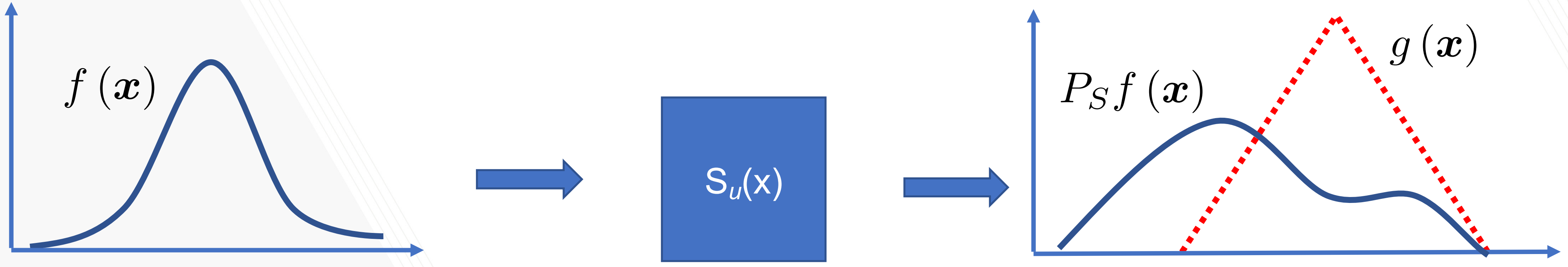
Patient A ***Patient B***



Data from Allen et al., 1980

Q: What dose should we give patients?

Optimal Decision-Making Under Uncertainty



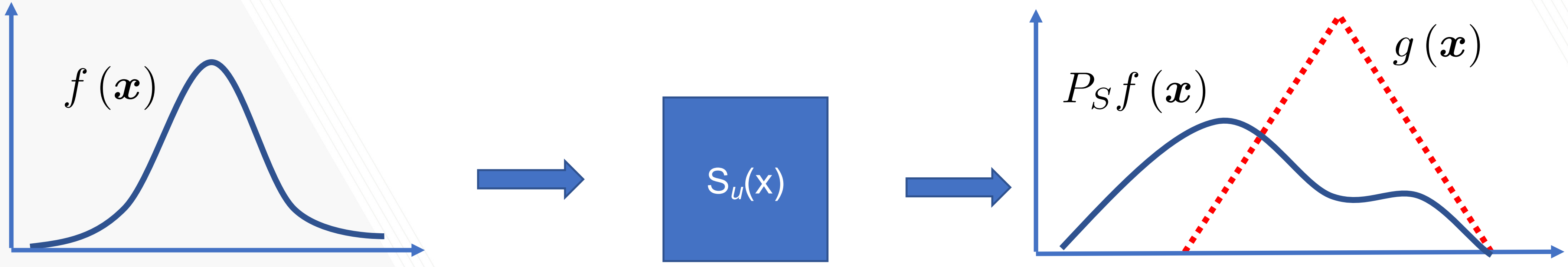
Initial density over uncertain states and parameters

State transformation under control selection u

"Pushed-forward" density and objective function

Choose u that maximizes $\mathbb{E}\left[g(X)\,|X \sim P_S f\right] = \int_{S(\Omega)} P_S f(\boldsymbol{x})\, g(\boldsymbol{x})\, d\boldsymbol{x}$

Optimal Decision-Making Under Uncertainty



Initial density over uncertain states and parameters

State transformation under control selection u

"Pushed-forward" density and objective function

Choose u that maximizes $\mathbb{E}\left[g(X)\,|X \sim P_S f\right] = \int_{S(\Omega)} P_S f(\boldsymbol{x})\, g(\boldsymbol{x})\, d\boldsymbol{x}$

Okay…but how do we compute $P_S f(x)$ for nonlinear/non-Gaussian systems?

Forward Density Propagation for Uncertain Systems

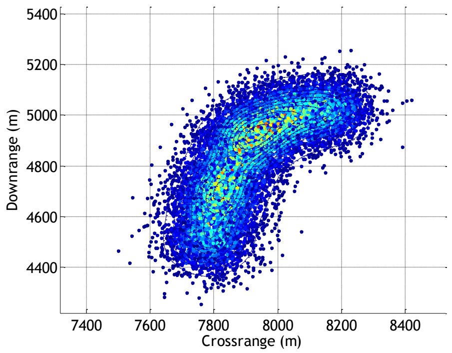
Frobenius-Perron (FP) Operator

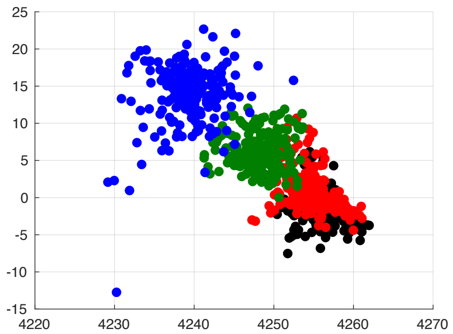
$$\int_A P_S f \, d\mu = \int_{S^{-1}(A)} f \, d\mu$$

$$S: X \to X$$

(measure preserving)

Monte Carlo Simulation





Polynomial Chaos

$$Y = \sum_{j=0}^{p} y_j \psi_j(\Xi) = \eta(x) \qquad X_P = \sum_{j=0}^{p} x_j \psi_j(\Xi)$$

The Koopman Operator

$$g : \mathbb{R}^n \to \mathbb{R}$$

Observable

$$S : \mathbb{R}^n \to \mathbb{R}^n$$

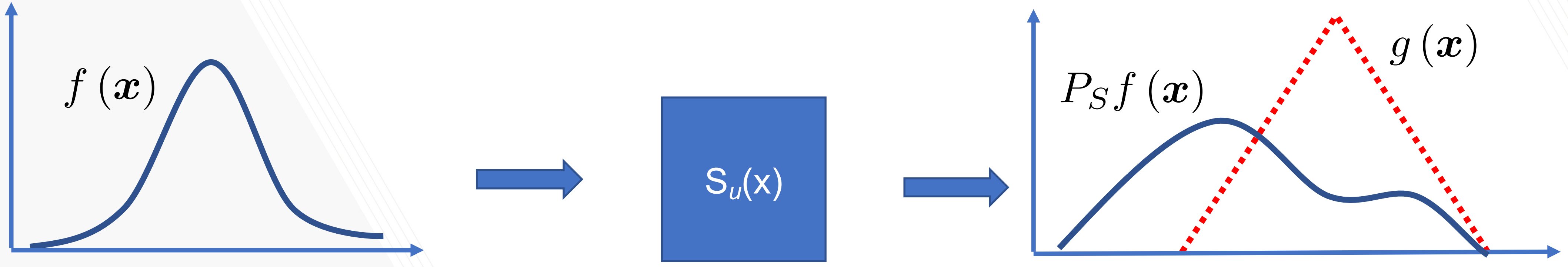
State Map

$$\mathcal{K}_S g\left(\boldsymbol{x}\right) = g\left(S\left(\boldsymbol{x}\right)\right)$$

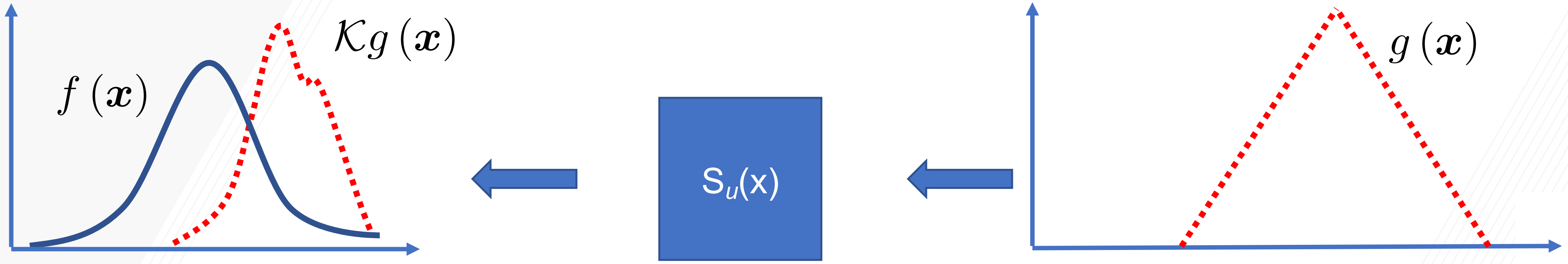
Koopman Operator

- Properties of Koopman operator of a system reveals properties of the underlying system
- Recent advancement in the literature for approximating via data-driven methods
  - Extended Dynamic Mode Decomposition (Williams *et al.* 2014, Korda and Mezic 2018)

Relationship with Uncertain Systems



$$\mathbb{E}\left[\mathcal{K}g(X)\,|X \sim f\right] = \int_{\Omega} f(\boldsymbol{x})\,\mathcal{K}g(\boldsymbol{x})\,d\boldsymbol{x} \quad = \quad \mathbb{E}\left[g(X)\,|X \sim P_S f\right] = \int_{S(\Omega)} P_S f(\boldsymbol{x})\,g(\boldsymbol{x})\,d\boldsymbol{x}$$



Benefits of the Koopman Expectation

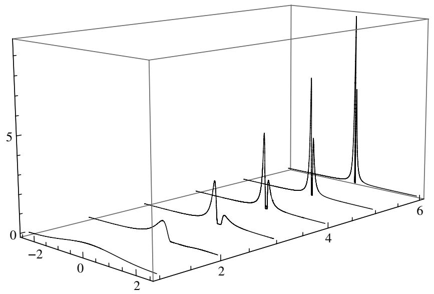
$$\mathbb{E}\left[\mathcal{K}g\left(X\right)|X\sim f\right]=\int_{\Omega}f\left(\boldsymbol{x}\right)\mathcal{K}g\left(\boldsymbol{x}\right)d\boldsymbol{x}$$

vs.

$$\mathbb{E}\left[g\left(X\right)|X\sim P_{S}f\right]=\int_{S(\Omega)}P_{S}f\left(\boldsymbol{x}\right)g\left(\boldsymbol{x}\right)d\boldsymbol{x}$$

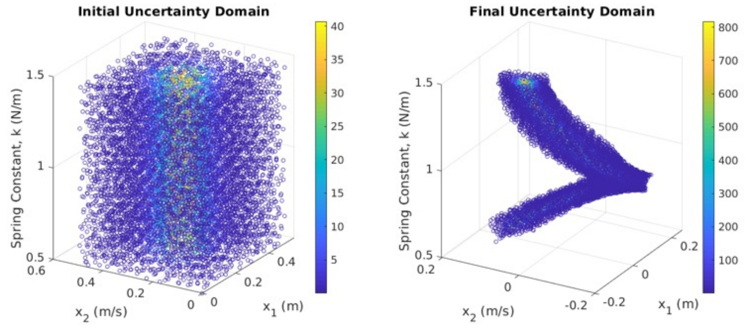
Pull-Back (Koopman) Expectation

Push-Forward (FP) Expectation



Halder and Bhattacharya, 2011

- Improved numerical stability
- Simpler evaluation
  - Domain of integration is initial domain Ω vs its image $S(\Omega)$
  - Provides well-defined structure of data, leading to simpler solution approaches (e.g., quadrature integration)



Meyers *et al.*, ACC 2019.

Benefits of the Koopman Expectation

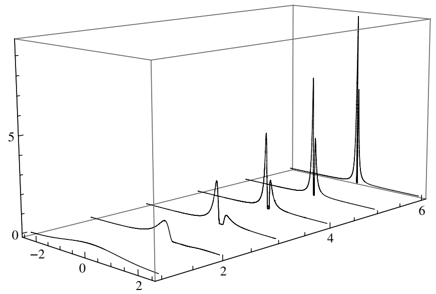
$$\mathbb{E}\left[\mathcal{K}g\left(X\right)|X\sim f\right]=\int_{\Omega}f\left(\boldsymbol{x}\right)\mathcal{K}g\left(\boldsymbol{x}\right)d\boldsymbol{x}$$

vs.

$$\mathbb{E}\left[g\left(X\right)|X\sim P_{S}f\right]=\int_{S(\Omega)}P_{S}f\left(\boldsymbol{x}\right)g\left(\boldsymbol{x}\right)d\boldsymbol{x}$$

Pull-Back (Koopman) Expectation

Push-Forward (FP) Expectation



Halder and Bhattacharya, 2011

- Improved numerical stability
- Simpler evaluation
  - Domain of integration is initial domain Ω vs its image $S(\Omega)$
  - Provides well-defined structure of data, leading to simpler solution approaches (e.g., quadrature integration)
- Computing expectation of multiple observables with varying supports in space-time
  - Pull-back each to a common domain domain ➞ Single, vector-valued expectation calculation

Meyers *et al.*, ACC 2019.

Georgia Tech
CREATING THE NEXT

Benefits of the Koopman Expectation

$$\mathbb{E}\left[\mathcal{K}g\left(X\right)|X\sim f\right]=\int_{\Omega}f\left(\boldsymbol{x}\right)\mathcal{K}g\left(\boldsymbol{x}\right)d\boldsymbol{x}$$

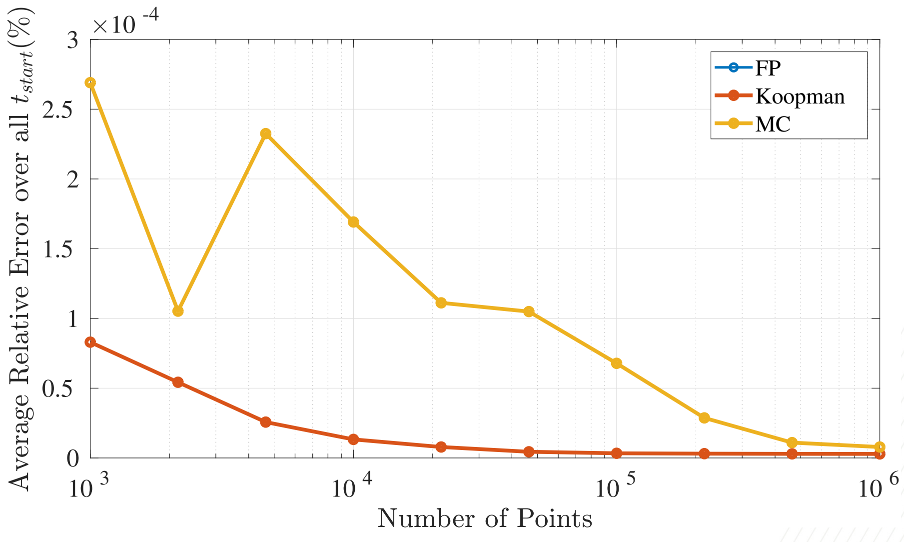
vs.

Monte Carlo Simulation

Pull-Back (Koopman) Expectation

- Faster convergence
- Error bounds / tolerancing via quadrature integration

- Downside: Koopman expectation assumes no process noise.
  - Application limited to systems which have only parametric uncertainty



Meyers *et al.*, ACC 2019.

Benefits of the Koopman Expectation

$$\mathbb{E}\left[\mathcal{K}g\left(X\right)|X\sim f\right]=\int_{\Omega}f\left(\boldsymbol{x}\right)\mathcal{K}g\left(\boldsymbol{x}\right)d\boldsymbol{x}$$

Pull-Back (Koopman) Expectation

vs.

Generalized Polynomial Chaos (gPC)

- In general, they are not the same
- However, there is an equivalence between the Koopman expectation and non-intrusive gPC when computing the mean value of an observable function $g(\boldsymbol{x})$

1

Koopman Expectation

$$\int_{\Omega}\mathcal{K}g(\boldsymbol{x})f(\boldsymbol{x})d\boldsymbol{x}=\int_{\Omega}\underbrace{\eta\left(f_p(\boldsymbol{\xi})\right)}_{g\left(S\left(f_p(\boldsymbol{\xi})\right)\right)=\mathcal{K}g(\boldsymbol{x})}\psi_0(\boldsymbol{\xi})p(\boldsymbol{\xi})d\boldsymbol{\xi}$$

gPC computation of mean of transformed RV

Benefits of the Koopman Expectation

$$\mathbb{E}\left[\mathcal{K}g\left(X\right)|X \sim f\right] = \int_{\Omega} f\left(\boldsymbol{x}\right)\mathcal{K}g\left(\boldsymbol{x}\right)d\boldsymbol{x}$$

Pull-Back (Koopman) Expectation

vs.

Generalized Polynomial Chaos (gPC)

- In general, they are not the same
- However, there is an equivalence between the Koopman expectation and non-intrusive gPC when computing the mean value of an observable function $g(\boldsymbol{x})$
- **Koopman** advantage: When computing higher-order moments, Koopman method (redefining observable) requires a lot less integrals than gPC
- **gPC** advantage: You can sample from transformed distribution (Koopman expectation only provides expected values)

Using the Koopman Expectation for Probabilistic Optimization

- In practice, we do not compute Koopman operator $\mathcal{K}_S$
- Instead, we compute *action* of the Koopman operator on observable functions of interest at discrete points in state space $\mathcal{K}_S g(x_i)$
  - Then integrals can be approximated via quadrature

$$\int_{\Omega} \mathcal{K}_S g(x) f_0(x) dx \approx \sum_{i=1}^{N} \mathcal{K}_S g(x_i) f_0(x_i) w_i$$

Note: We can also use other methods such as Monte Carlo integration to compute this as well.

$\mathcal{K}_S g(x_i)$: From each discrete sample x_i, forward simulate and compute observable function

$f_0(x_i)$: Initial uncertainty PDF evaluated at sample x_i

w_i: Quadrature weight

Meyers *et al.*, ACC 2019.

Georgia Tech
CREATING THE NEXT

Probabilistic Optimization via the Koopman Expectation

- We wish to solve the following optimization problem:

$$\boldsymbol{u}^* = \underset{\boldsymbol{u}\in\mathcal{U}}{\arg\min} \int_{\Omega} \mathcal{K}_S g(\boldsymbol{x},\boldsymbol{u}) f_0(\boldsymbol{x}) d\boldsymbol{x}$$

Minimize expected value of cost

subject to:

$$\int_{\Omega} \mathcal{K}_S \boldsymbol{c}(\boldsymbol{x},\boldsymbol{u}) f_0(\boldsymbol{x}) d\boldsymbol{x} < \boldsymbol{r}$$

Satisfy chance constraints

Meyers *et al.*, ACC 2019.

Probabilistic Optimization via the Koopman Expectation

- We wish to solve the following optimization problem:

$$\boldsymbol{u}^* = \underset{\boldsymbol{u} \in \mathcal{U}}{\arg\min} \int_{\Omega} \mathcal{K}_S g(\boldsymbol{x}, \boldsymbol{u}) f_0(\boldsymbol{x}) d\boldsymbol{x}$$

subject to:

$$\int_{\Omega} \mathcal{K}_S \boldsymbol{c}(\boldsymbol{x}, \boldsymbol{u}) f_0(\boldsymbol{x}) d\boldsymbol{x} < \boldsymbol{r}$$

Key point: Because cost and constraint functions pulled back to initial time via Koopman operator, $f_0(\boldsymbol{x})$ is never explicitly propagated forward in time.

Meyers *et al.*, ACC 2019.

Georgia Tech
CREATING THE NEXT

Example 1: Bouncing Ball

Bouncing ball in 2D with uncertain coefficient of restitution.
Compute expected cost value (no optimization).

System:

$$\ddot{\boldsymbol{x}} = \begin{bmatrix} \ddot{x} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ -g \end{bmatrix}, \qquad x_0 = 2\,\mathrm{m}, \dot{x}_0 = 2\,\mathrm{m/s}, z_0 = 50\,\mathrm{m}, \dot{z}_0 = 0\,\mathrm{m/s}$$

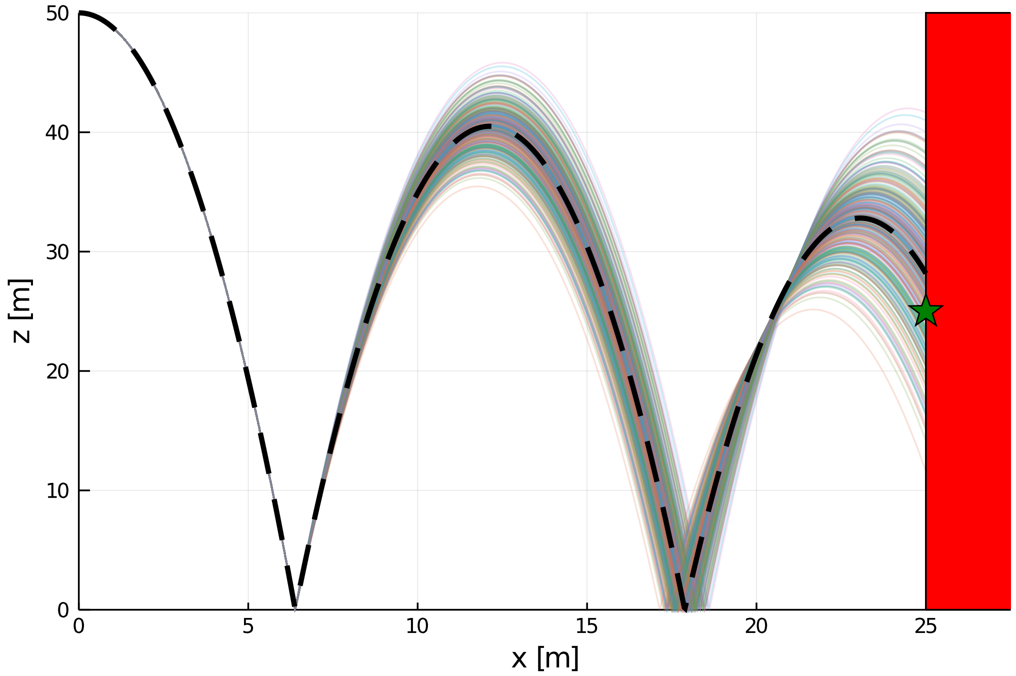
$$\dot{z}^+ = -\alpha\dot{z}^- \qquad \text{when } z = 0$$

Uncertainty:

$$\alpha \sim \mathcal{N}(0.9, 0.02) \qquad \text{truncated at 0.84 and 1}$$

Cost:

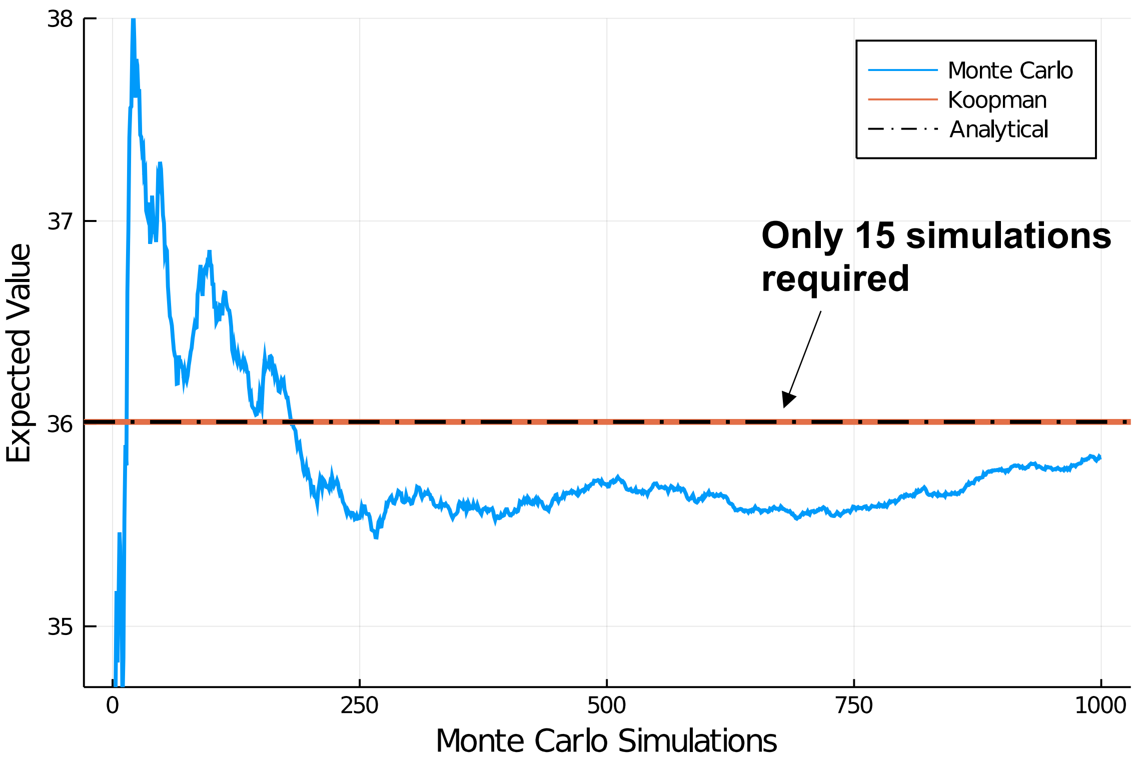
$$g(\boldsymbol{x}) = (z - z^*)^2$$

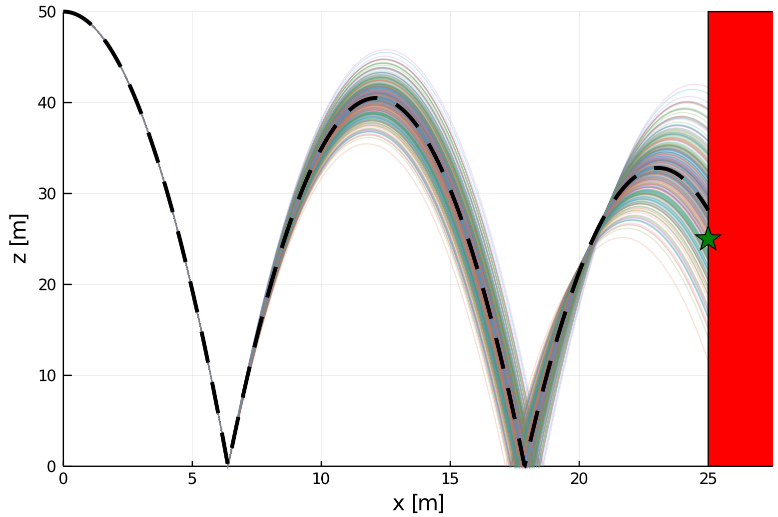


Gerlach *et al.*, 2020, https://arxiv.org/pdf/2008.08737.pdf

Example 1: Bouncing Ball

Bouncing ball in 2D with uncertain coefficient of restitution.
Compute expected cost value (no optimization).





	Analytical	Monte Carlo	Koopman
No. of Simulations	-	100,000	**15**
Exp. Value (m^2)	**36.008**	35.782	**36.008**
Computation Time (s)	-	2.060	0.0012

Gerlach *et al.*, 2020, https://arxiv.org/pdf/2008.08737.pdf

Example 1: Bouncing Ball

Bouncing ball in 2D with uncertain coefficient of restitution.

System:

$$\ddot{\boldsymbol{x}} = \begin{bmatrix} \ddot{x} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ -g \end{bmatrix}$$

$$\dot{z}^{+} = -\alpha \dot{z}^{-} \qquad \text{when } z = 0$$

Initial Conditions to Optimize:

$$x_0 \in [-100\,\mathrm{m}, 0\,\mathrm{m}]$$

$$\dot{x}_0 \in [1\,\mathrm{m/s}, 3\,\mathrm{m/s}]$$

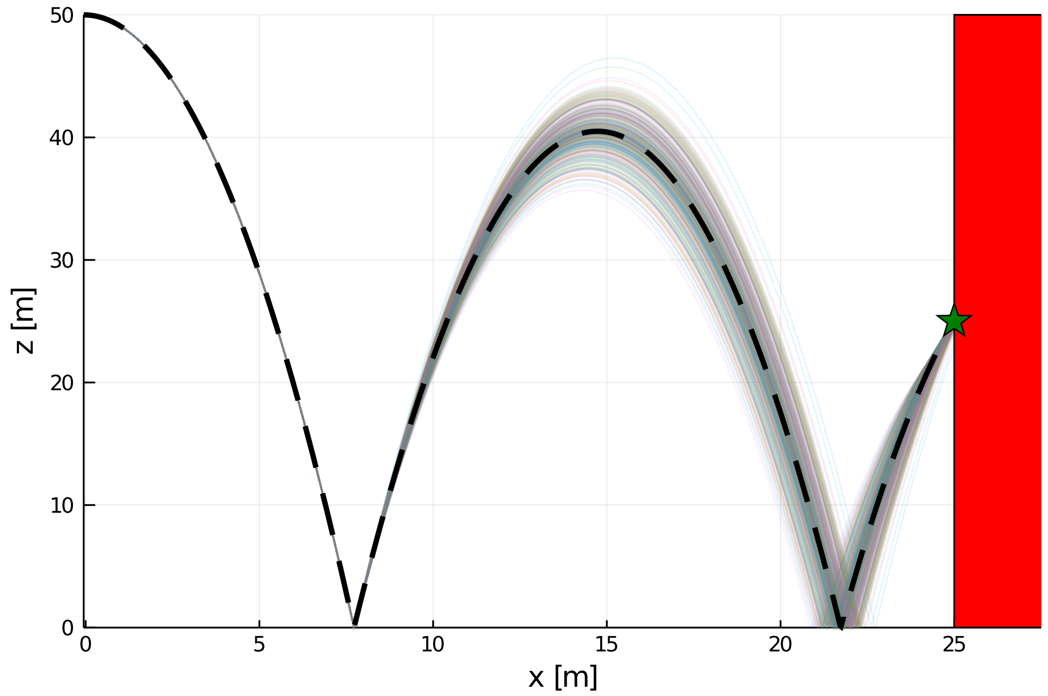
$$z_0 \in [10\,\mathrm{m}, 50\,\mathrm{m}]$$

Uncertainty:

$$\alpha \sim \mathcal{N}(0.9, 0.02) \qquad \text{truncated at 0.84 and 1}$$

Cost:

$$g(\boldsymbol{x}) = (z - z^*)^2$$



Gradient-Based optimization to solve $\boldsymbol{u}^* = \underset{\boldsymbol{u} \in \mathcal{U}}{\arg\min} \int_{\Omega} \mathcal{K}_S\, g(\boldsymbol{x}, \boldsymbol{u}) f_0(\boldsymbol{x}) d\boldsymbol{x}$

Gerlach *et al.*, 2020, https://arxiv.org/pdf/2008.08737.pdf

Example 1: Bouncing Ball

Bouncing ball in 2D with uncertain coefficient of restitution.

System:

$$\ddot{\boldsymbol{x}} = \begin{bmatrix} \ddot{x} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ -g \end{bmatrix}$$

$$\dot{z}^{+} = -\alpha \dot{z}^{-} \qquad \text{when } z = 0$$

Initial Conditions to Optimize:

$$x_0 \in [-100\,\text{m}, 0\,\text{m}]$$

$$\dot{x}_0 \in [1\,\text{m/s}, 3\,\text{m/s}]$$

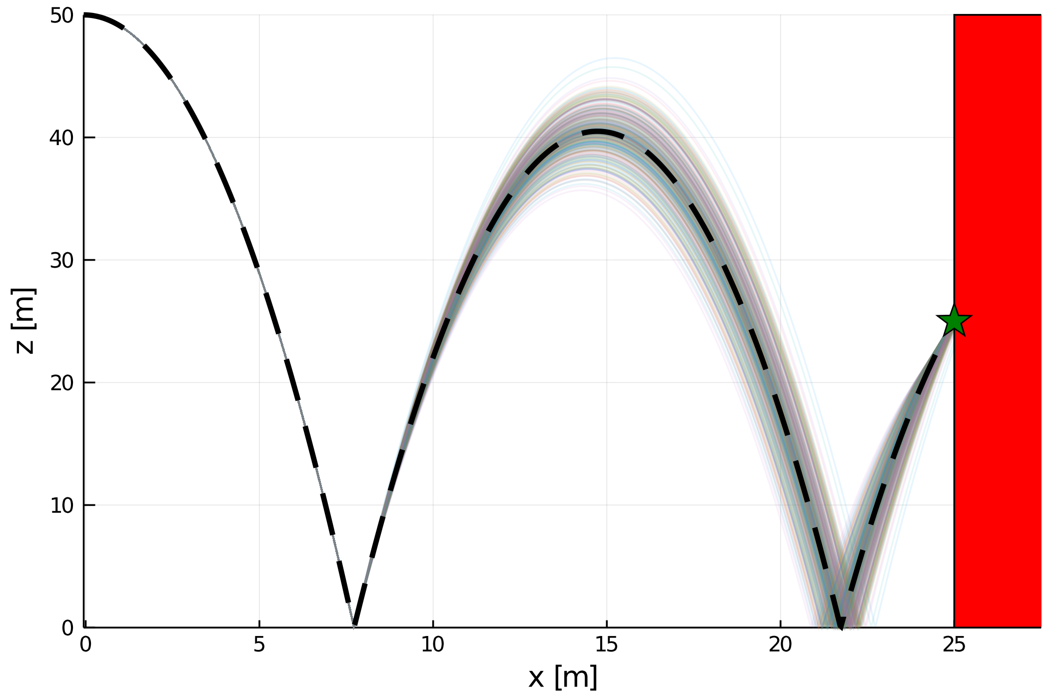
$$z_0 \in [10\,\text{m}, 50\,\text{m}]$$

Uncertainty:

$$\alpha \sim \mathcal{N}(0.9, 0.02) \qquad \text{truncated at 0.84 and 1}$$

Cost:

$$g(\boldsymbol{x}) = (z - z^*)^2$$



Optimal Solution produces expected cost of 8.3×10^{-2} in 0.12 sec (Julia implementation)

Gerlach *et al.*, 2020, https://arxiv.org/pdf/2008.08737.pdf

Georgia Tech
CREATING THE NEXT

Example 1: Bouncing Ball

Particular observable functions defined to “extract” raw moments (which can then be converted to central moments)

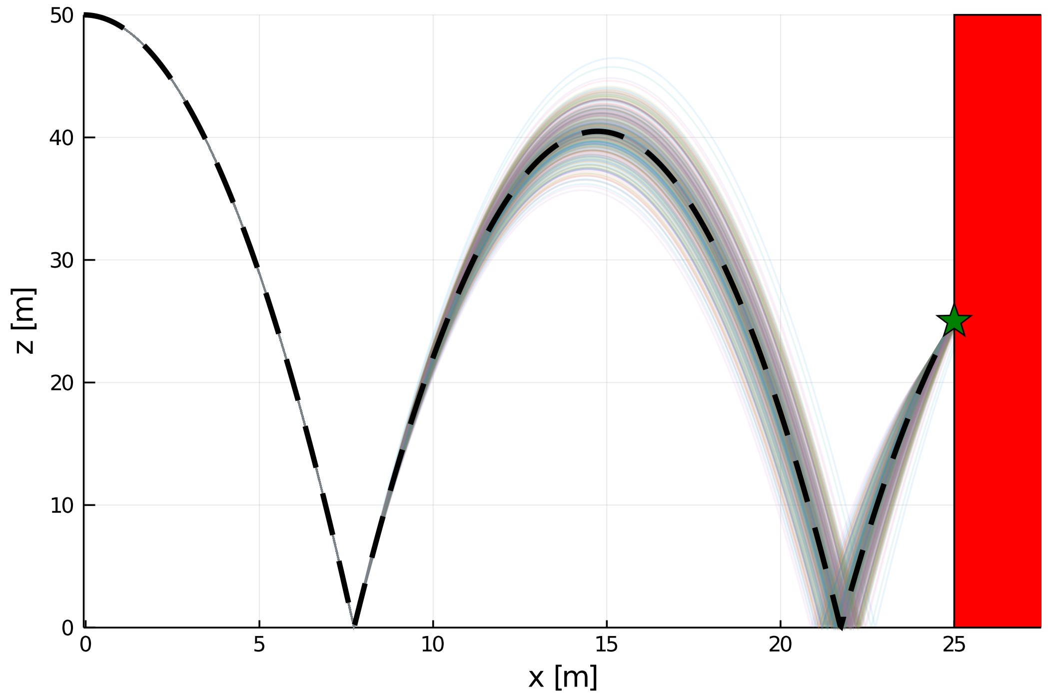
$g_1(x) = x$ mean

$g_2(x) = x^2$ 2nd raw moment

$g_3(x) = x^3$ 3rd raw moment

$\vdots$

$$\int_{\Omega} \mathcal{K}_S g_i(\boldsymbol{x}, \boldsymbol{u}) f_0(\boldsymbol{x}) d\boldsymbol{x}$$



Gerlach *et al.*, 2020, https://arxiv.org/pdf/2008.08737.pdf

Example 1: Bouncing Ball

Particular observable functions defined to "extract" raw moments (which can then be converted to central moments)

$g_1(x) = x$ mean

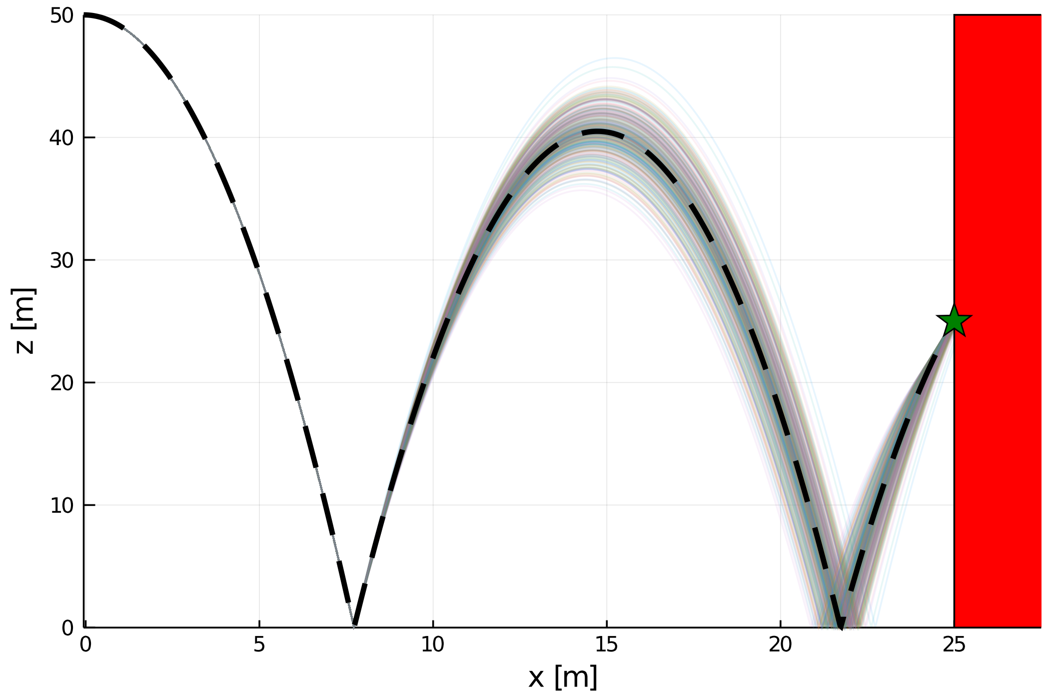
$g_2(x) = x^2$ 2nd raw moment

$g_3(x) = x^3$ 3rd raw moment

$\vdots$

$$\int_{\Omega} \mathcal{K}_S g_i(\boldsymbol{x}, \boldsymbol{u}) f_0(\boldsymbol{x}) d\boldsymbol{x}$$

Central Moment	Monte Carlo	Koopman
2	9.030e−2	9.007e−2 ± 3.878e−5
3	3.878e−1	3.924e−1 ± 1.776e−3
4	3.214	3.428 ± 1.536e−3
5	38.116	44.536 ± 3.733e−3
	10M simulations, 264 sec	225 simulations, 3.4 ms

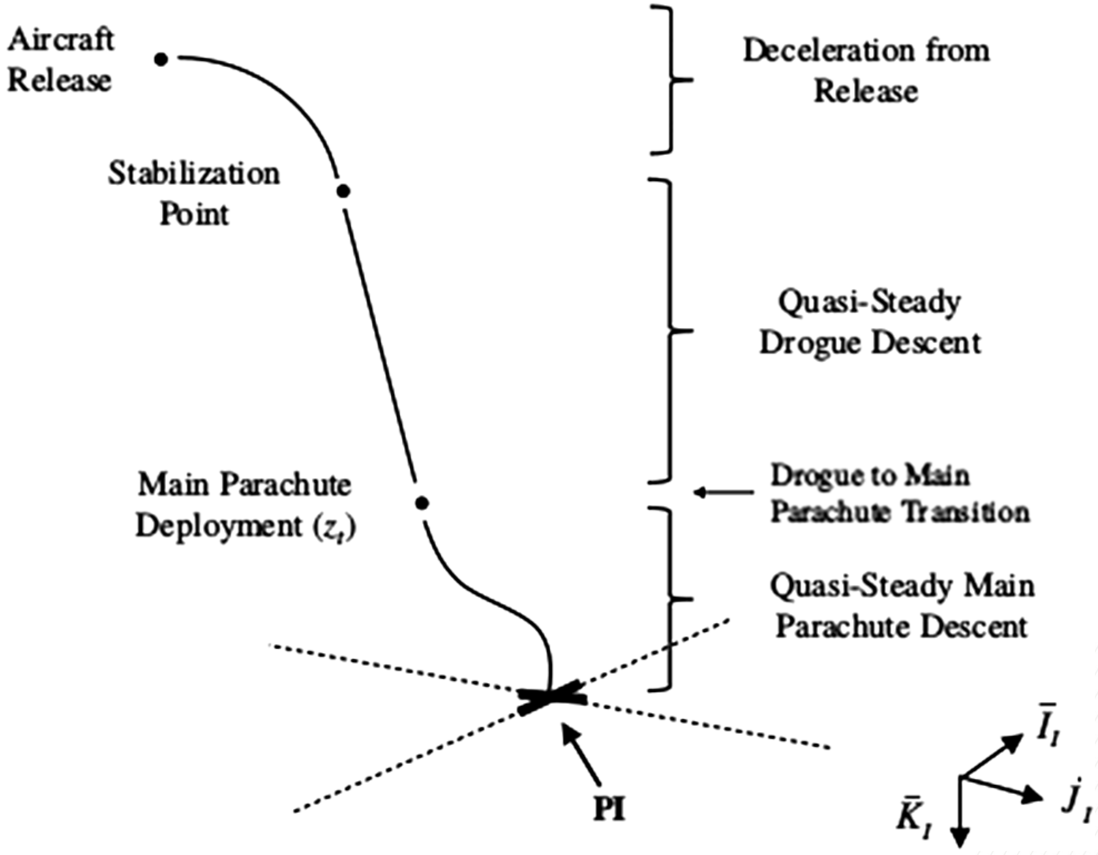


Koopman-based method produces solution with same accuracy but runs 77,000x faster.

Gerlach *et al.*, 2020, https://arxiv.org/pdf/2008.08737.pdf

Example 2: Airdrop Mission Planning

High-Altitude Low-Opening (HALO) Airdrop



Uncertainty: Winds, parachute drag, package release dynamics

Optimize: Release point, aircraft heading, opening altitude

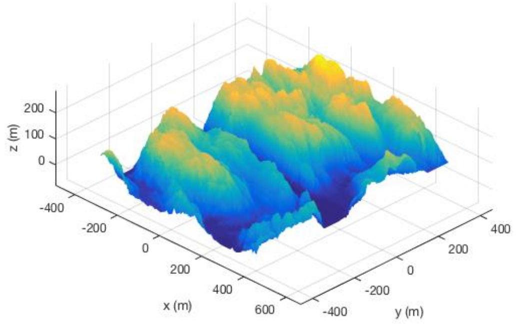
Leonard *et al.*, J. Guidance, Control, and Dynamics, 2020.

Example 2: Airdrop Mission Planning

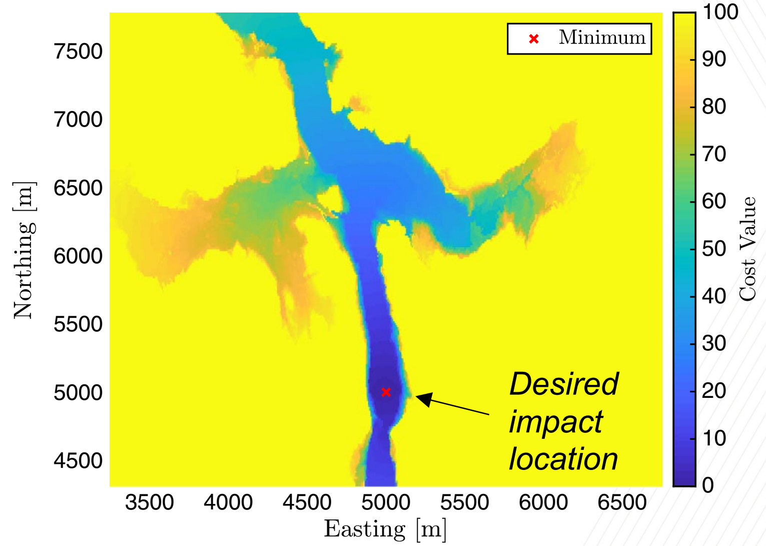
High-Altitude Low-Opening (HALO) Airdrop

Uncertainty: Winds, parachute drag, package release dynamics

Optimize: Release point, aircraft heading, opening altitude



Cost Function $g(x, y)$



$$\mathbb{E}_{\mathcal{D}}[G(\boldsymbol{X})|\boldsymbol{u}] = \int_{\Omega} G(\boldsymbol{x}) f_{xy}(x, y|\boldsymbol{u}) f_{C_d}(C_d) f_{\hat{w}_m}(\hat{w}_m) f_{\hat{x}_\psi}(\hat{w}_\psi)\, \mathrm{d}\boldsymbol{x}$$

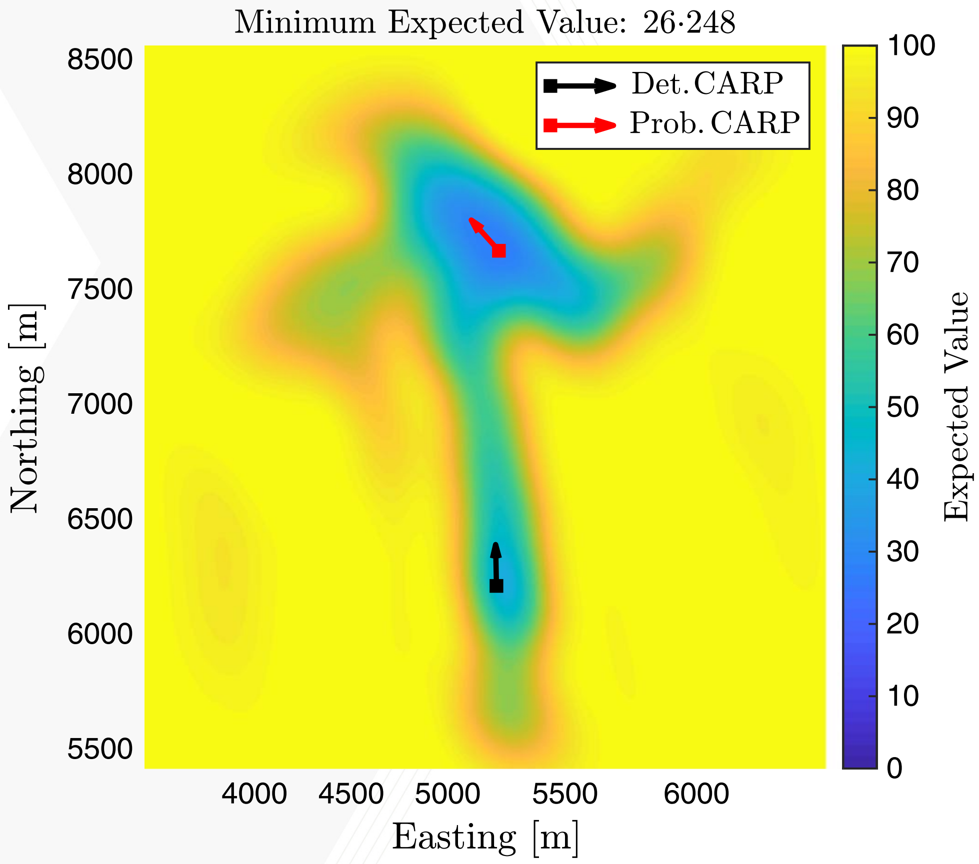
Choose drop location (x, y) that minimizes this expected value

Leonard *et al.*, J. Guidance, Control, and Dynamics, 2020.

Georgia Tech
CREATING THE NEXT

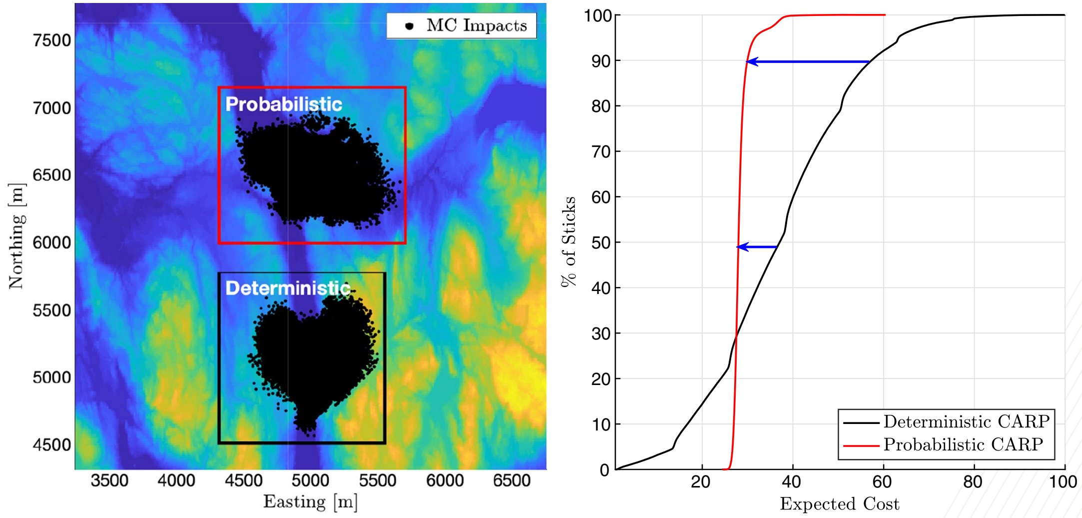
Example 2: Airdrop Mission Planning

High-Altitude Low-Opening (HALO) Airdrop



Leonard *et al.*, J. Guidance, Control, and Dynamics, 2020.

- **Deterministic planner** does not account for uncertainty, drops straight into canyon (lots of bad outcomes)
- **Probabilistic planner** drops in flatter region – gives up best-case performance to protect against lots of poor outcomes



Example 3: Maneuver-Based Trajectory Planning

Use library of (uncertain) maneuvers to construct path that minimizes expected cost while satisfying chance constraints

$$\min_{\mu \in U} E\left[J\left(H_\mu(x_0, t_0)\right)\right]$$

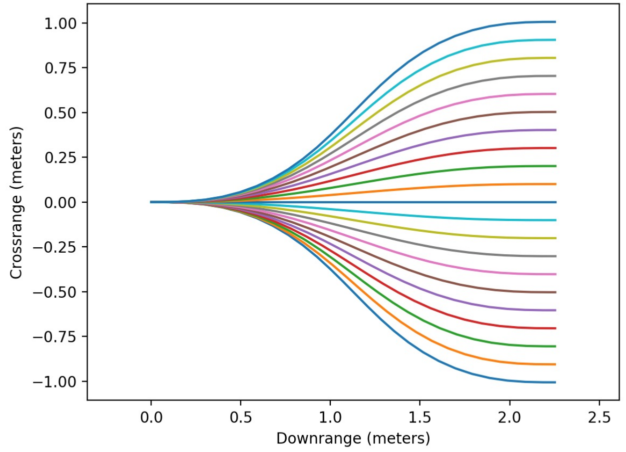
s.t. $P\left(H_\mu(x_0, t_0) \notin F\right) \leq r$

H_μ gives the state history under the controller μ

Koopman operator used to pull-back expected cost and constraint values for each maneuver

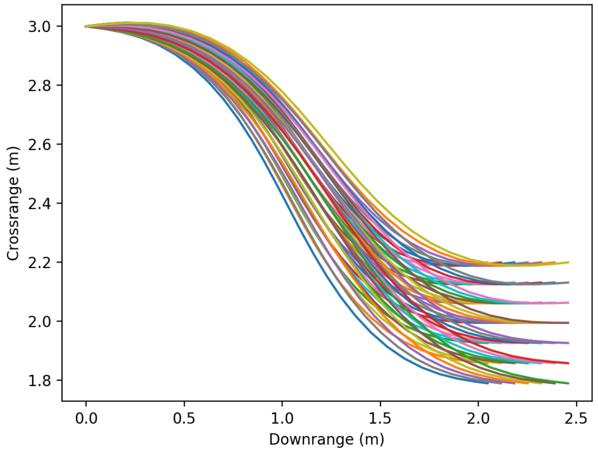
Gutow and Rogers, IEEE RAL, 2020.

Library:



Single Maneuver Under Parameter Uncertainty:

Each realization has probability of occurring given joint distribution on parameters or ICs

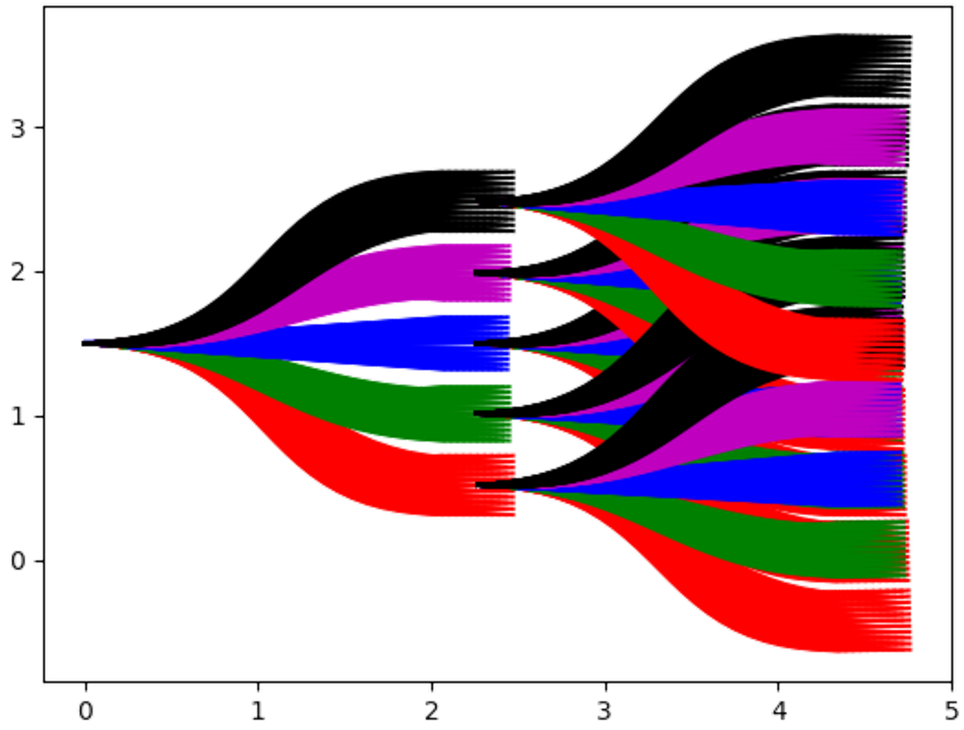


Georgia Tech
CREATING THE NEXT

Example 3: Maneuver-Based Trajectory Planning

"Expected State Planner"

- Chain together next primitive from expected state of last one
- Use Koopman operator to pull back expected costs and constraint violations of candidate paths
- Use of primitives + Koopman allows UQ without real-time simulation

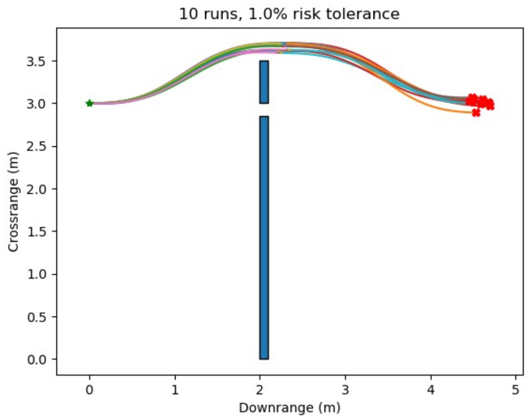


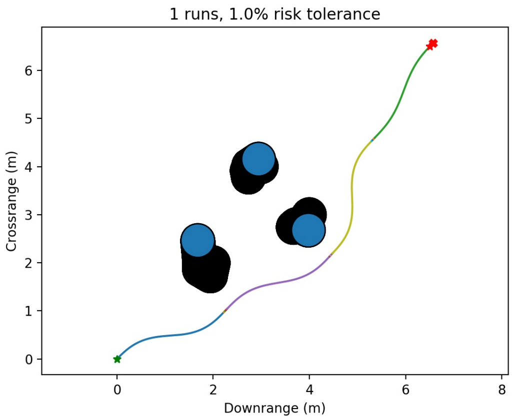
A* or dynamic programming can be used to solve for optimal path.

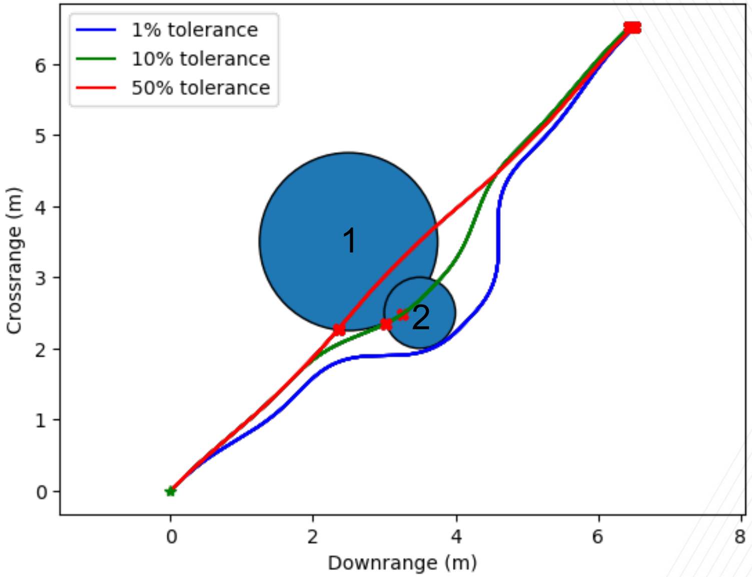
Gutow and Rogers, IEEE RAL, 2020.

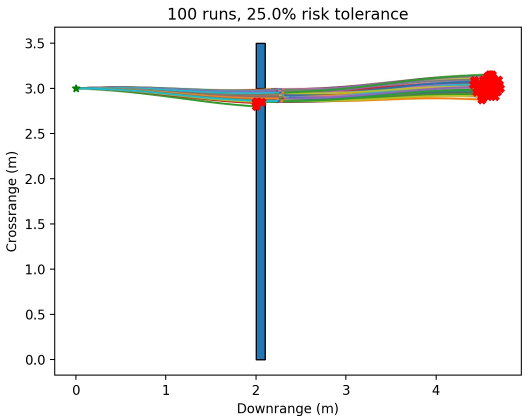
Example 3: Maneuver-Based Trajectory Planning

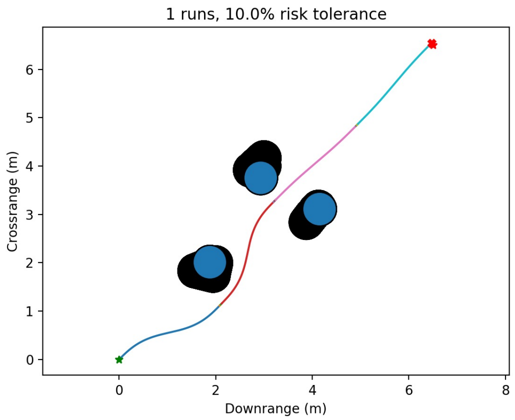
Yields planner with tunable risk thresholds











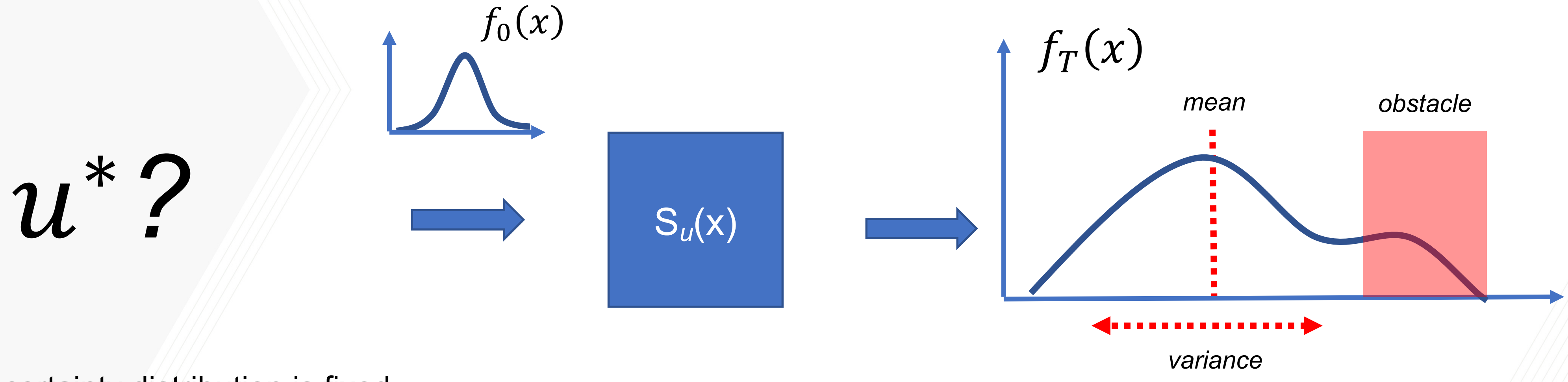
- Vehicle has 40% chance of being destroyed every 0.25 sec inside region 1
- Vehicle has 2.5% chance of being destroyed every 0.025 sec inside region 2
- Trajectory adapts based on risk tolerance

Gutow and Rogers, IEEE RAL, 2020.

Georgia Tech
CREATING THE NEXT

Probabilistic Inverse Problems

So far, we have tried to optimize vector of initial inputs given desired expected values of observables



- Initial uncertainty distribution is fixed
- We are allowed to pick the system

OR

- Form of initial uncertainty distribution is fixed
- We are allowed to set its parameters

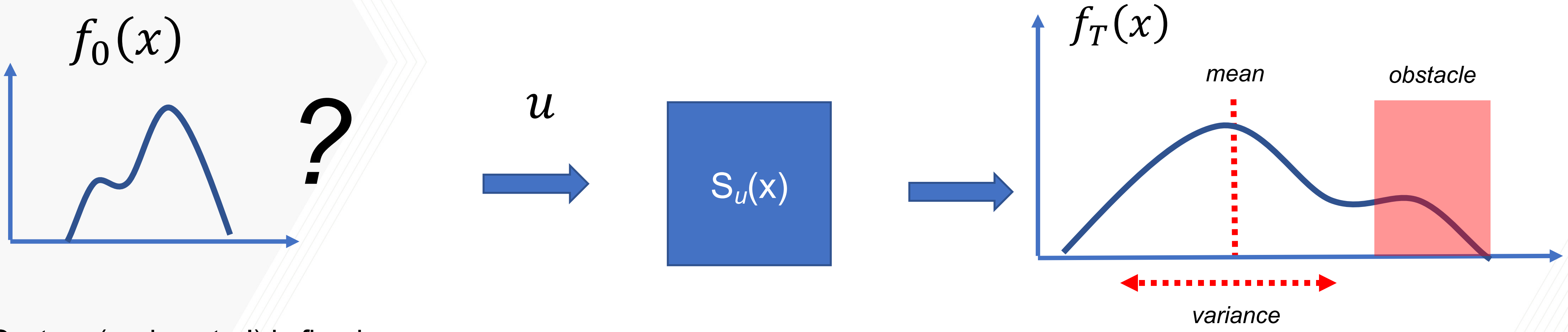
$$\int_\Omega g_1(x,u) f_T(\boldsymbol{x}) d\boldsymbol{x} - c_1 = 0$$

$$\int_\Omega g_2(x,u) f_T(\boldsymbol{x}) d\boldsymbol{x} - c_2 = 0$$

$$\vdots$$

Probabilistic Inverse Problems

So far, we have tried to optimize vector of initial inputs given desired expected values of observables



- System (and control) is fixed
- What is the initial uncertainty distribution that meets desired expected values?

- Engineering design problems, drug design/dosing, disease modeling, biological population modeling…

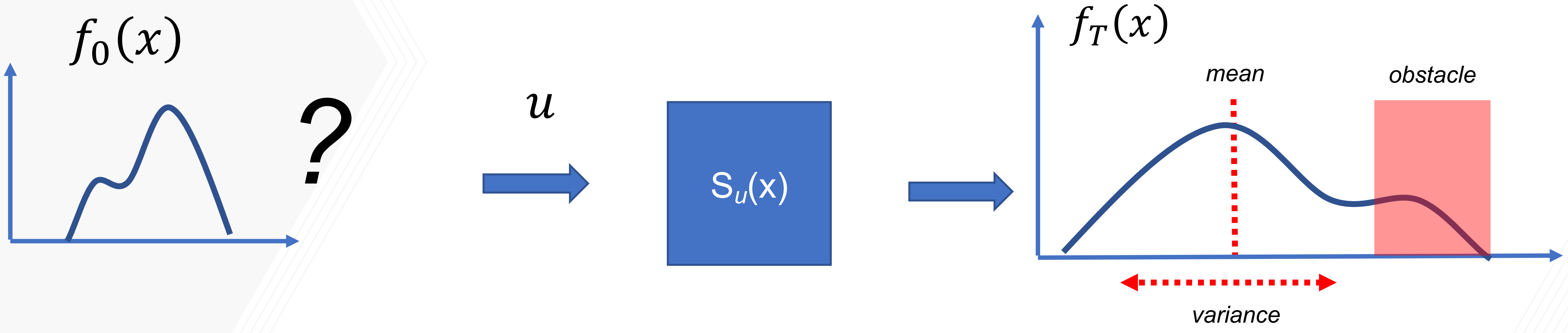
$$\int_{\Omega} g_1(x,u) f_T(\boldsymbol{x}) d\boldsymbol{x} - c_1 = 0$$

$$\int_{\Omega} g_2(x,u) f_T(\boldsymbol{x}) d\boldsymbol{x} - c_2 = 0$$

$$\vdots$$

Probabilistic Inverse Problems

So far, we have tried to optimize vector of initial inputs given desired expected values of observables



Probabilistic inverse problem: Given expectations of observables of the output, what is a valid input distribution?

$$\int_{\Omega} g_1(x,u) f_T(\boldsymbol{x}) d\boldsymbol{x} - c_1 = 0$$

$$\int_{\Omega} g_2(x,u) f_T(\boldsymbol{x}) d\boldsymbol{x} - c_2 = 0$$

$$\vdots$$

Probabilistic Inverse Problems

Problem statement:

Find $f_0(\boldsymbol{x})$ s.t.:

$$1 = \int_{supp(f_0)} f_0(\mathbf{x})\, d\mathbf{x}$$

Integrate to 1 constraint

$$c_i = \int_{supp(f_0)} f_0(\mathbf{x}) U_i g_i(\mathbf{x})\, d\mathbf{x} \qquad i = 1, \ldots, p$$

EV equality constraints

$$c_j < \int_{supp(f_0)} f_0(\mathbf{x}) U_j g_j(\mathbf{x})\, d\mathbf{x} \qquad j = p+1, \ldots, K$$

EV inequality constraints

U_i is Koopman operator that pulls observable function back from time t_i to t_0

Meyers *et al.*, J. Comp. Phys., 2021.

Georgia Tech
CREATING THE NEXT

Probabilistic Inverse Problems

Problem statement:

Find $f_0(\boldsymbol{x})$ s.t.:

$$1 = \int_{supp(f_0)} f_0(\mathbf{x})\, d\mathbf{x}$$

Integrate to 1 constraint

$$c_i = \int_{supp(f_0)} f_0(\mathbf{x}) U_i g_i(\mathbf{x})\, d\mathbf{x} \qquad i = 1, \ldots, p$$

EV equality constraints

$$c_j < \int_{supp(f_0)} f_0(\mathbf{x}) U_j g_j(\mathbf{x})\, d\mathbf{x} \qquad j = p+1, \ldots, K$$

EV inequality constraints

Optimize over the space of L^1 functions…this is hard.

Meyers *et al.*, J. Comp. Phys., 2021.

Georgia Tech CREATING THE NEXT

Probabilistic Inverse Problems

Problem statement:

Find $f_0(\boldsymbol{x})$ s.t.:

$$1 = \int_{supp(f_0)} f_0(\mathbf{x})\, d\mathbf{x}$$

Integrate to 1 constraint

$$c_i = \int_{supp(f_0)} f_0(\mathbf{x}) U_i g_i(\mathbf{x})\, d\mathbf{x} \qquad i = 1, \dots, p$$

EV equality constraints

$$c_j < \int_{supp(f_0)} f_0(\mathbf{x}) U_j g_j(\mathbf{x})\, d\mathbf{x} \qquad j = p+1, \dots, K$$

EV inequality constraints

Optimize over the space of L^1 functions…this is hard.

This is an ill-posed problem.
So we will need regularization.

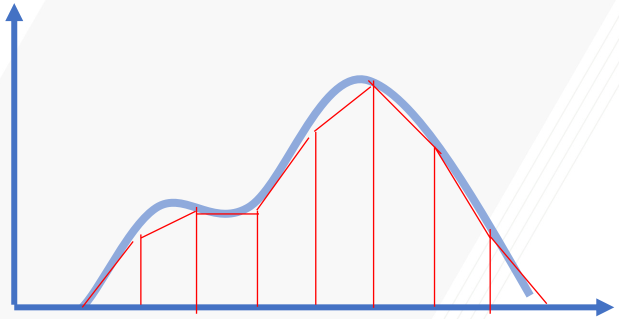
Meyers *et al.*, J. Comp. Phys., 2021.

Georgia Tech
CREATING THE NEXT

Probabilistic Inverse Problems

Formulation as a quadratic program:

Approximate $f_0(\boldsymbol{x})$ as piecewise linear over grid



$$\hat{c}_i \approx \sum_{k=0}^{n} w_k \hat{f}_0(\mathbf{x}_k) g_i(S_i(\mathbf{x}_k))$$

Quadrature approximation of desired EVs

$$\underset{\mathbf{f}\in\mathbb{R}^n}{\operatorname{argmin}} ||G\mathbf{f} - \mathbf{c}||_2^2 + \lambda^2 ||L\mathbf{f}||_2^2$$

EV targets (LS cost)

Regularization

$$\mathbf{w}^T\mathbf{f} = 1$$ Integrate to 1 constraint

$$G_{eq}\mathbf{f} = \mathbf{c}_{eq}$$ EV equality constraints

$$G_{ineq}\mathbf{f} \geq \mathbf{c}_{ineq}$$ EV inequality constraints

$$\mathbf{f} \geq \mathbf{0}$$

Meyers *et al.*, J. Comp. Phys., 2021.

Probabilistic Inverse Problems

Formulation as a quadratic program:

- Non-negative constrained least-squares problem
- Cast as a convex quadratic program
- Use QP solver to find vector **f** which approximates initial distribution

$$\underset{\mathbf{f}\in\mathbb{R}^n}{\operatorname{argmin}} ||G\mathbf{f}-\mathbf{c}||_2^2 + \lambda^2||L\mathbf{f}||_2^2$$

EV targets (LS cost)

Regularization

$\mathbf{w}^T\mathbf{f} = 1$	Integrate to 1 constraint
$G_{eq}\mathbf{f} = \mathbf{c}_{eq}$	EV equality constraints
$G_{ineq}\mathbf{f} \geq \mathbf{c}_{ineq}$	EV inequality constraints
$\mathbf{f} \geq \mathbf{0}$	

Meyers *et al.*, J. Comp. Phys., 2021.

Probabilistic Inverse Problems

Formulation as a quadratic program:

- Non-negative constrained least-squares problem
- Cast as a convex quadratic program
- Use QP solver to find vector **f** which approximates initial distribution

$$\underset{\mathbf{f}\in\mathbb{R}^n}{\operatorname{argmin}} ||G\mathbf{f}-\mathbf{c}||_2^2 + \lambda^2 ||L\mathbf{f}||_2^2$$

EV targets (LS cost)

Regularization

$\mathbf{w}^T\mathbf{f} = 1$	Integrate to 1 constraint
$G_{eq}\mathbf{f} = \mathbf{c}_{eq}$	EV equality constraints
$G_{ineq}\mathbf{f} \geq \mathbf{c}_{ineq}$	EV inequality constraints
$\mathbf{f} \geq \mathbf{0}$	

Made possible because we formulated problem using Koopman expectations!

Meyers *et al.*, J. Comp. Phys., 2021.

Georgia Tech
CREATING THE NEXT

Inverse Problem Example: Reentry Vehicle

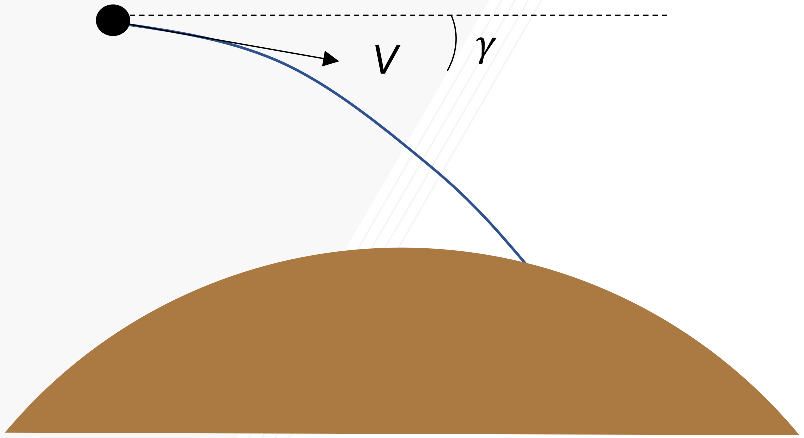
- Vinh's Equations

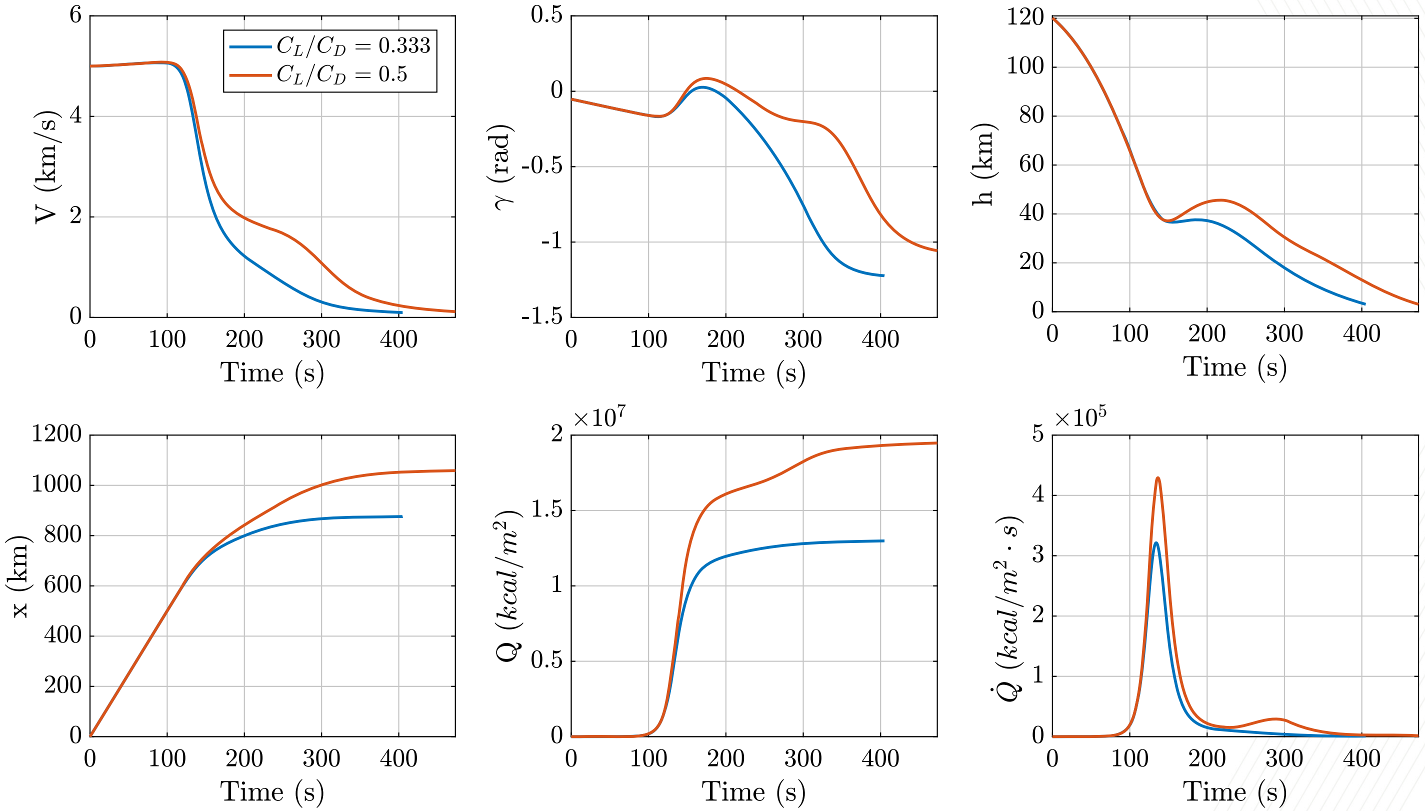
$$\dot{x} = V\cos\gamma\,,$$

$$\dot{r} = V\sin\gamma\,,$$

$$\dot{V} = \frac{-\rho S C_D}{2m}V^2 + g_0\left(\frac{r_0}{r}\right)^2\sin\gamma\,,$$

$$\dot{\gamma} = \frac{\rho S C_L}{2m}V + \left(\frac{V}{r} - \frac{g_0}{V}\left(\frac{r_0}{r}\right)^2\right)\cos\gamma$$





Meyers *et al.*, J. Comp. Phys., 2021.

Inverse Problem Example: Reentry Vehicle

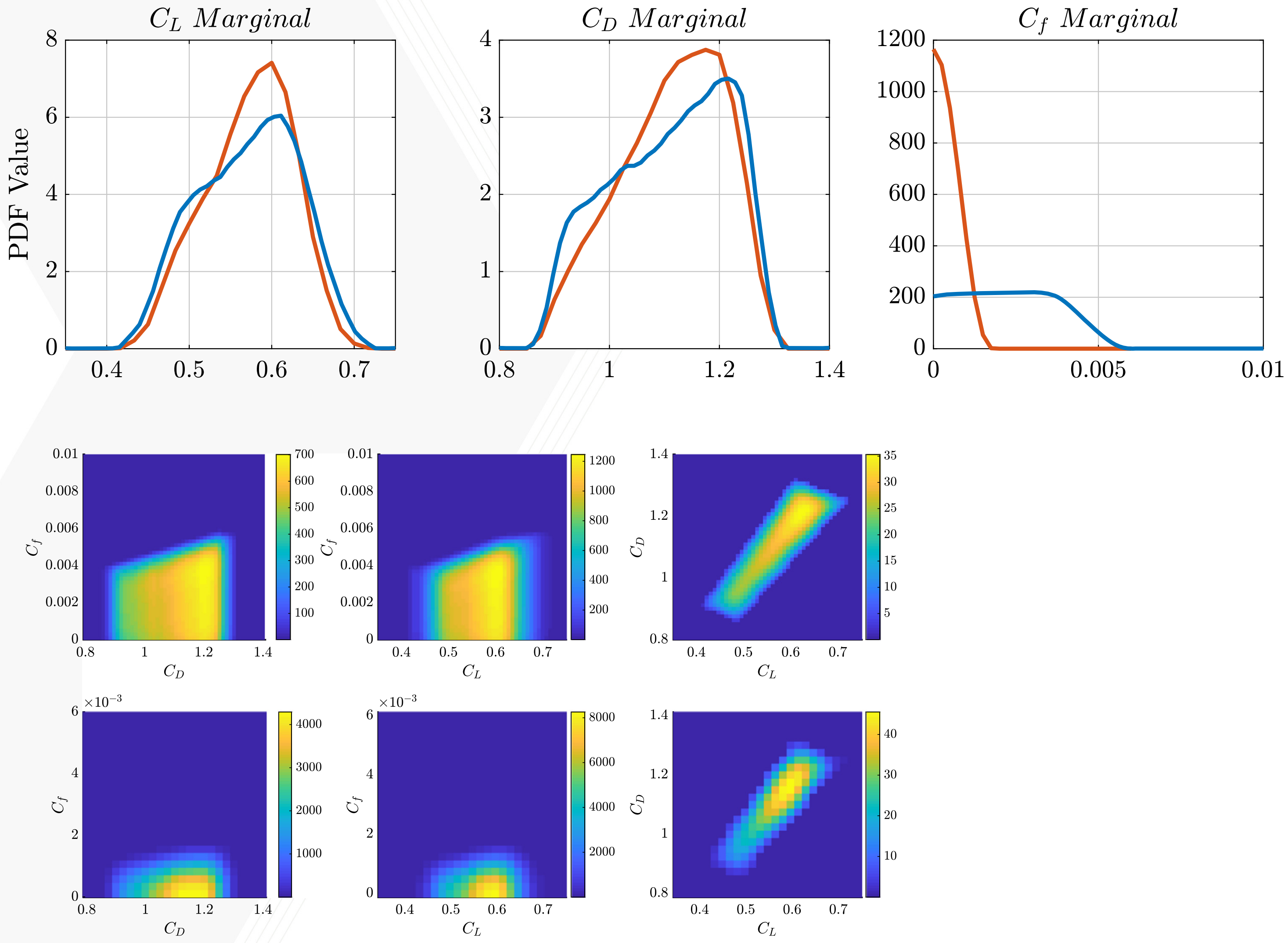
Case	Expected Value	Constraint	
Case 1	$Pr(1000 \le x(T) \le 1150\ \mathrm{km})$	≥ 0.99	Final **position** constraint
	$Pr(0.09 \le V(T) \le 0.11\ \mathrm{km/s})$	≥ 0.99	Final **velocity** constraint
	$Pr(Q(T) < 1.5\times10^7\ \mathrm{kcal/m})$	≥ 0.99	Final **integrated heat load** constraint
	$Pr\left(\max \dot{Q}(T) < 3\times10^5\ \mathrm{kcal/m^2/s}\right)$	≥ 0.99	Maximum **heating rate** constraint (allowable range)
Case 2	$Pr(1000 \le x(T) \le 1150\ \mathrm{km})$	≥ 0.99	
	$Pr(0.09 \le V(T) \le 0.11\ \mathrm{km/s})$	≥ 0.99	
	$E\left[\max \dot{Q}(T)\right]\quad (\mathrm{kcal/m^2/s})$	$= 3\times10^5$	Maximum **heating rate** equality constraint

Uncertainty in lift coefficient (C_L), drag coefficient (C_D), heating coefficient (C_f)

What are allowable distributions for them?

Meyers *et al.*, J. Comp. Phys., 2021.

Inverse Problem Example: Reentry Vehicle

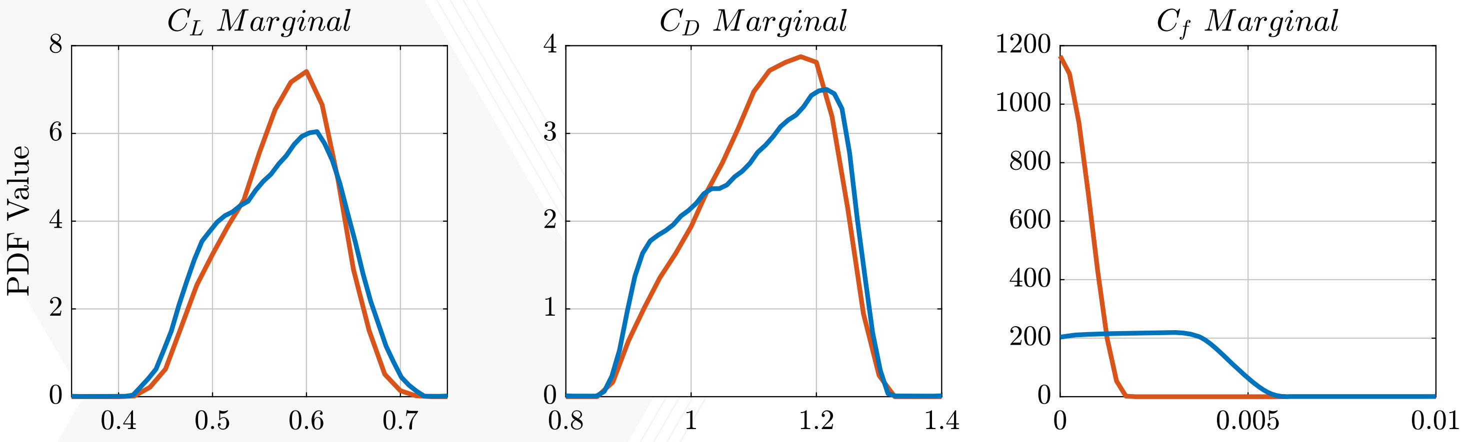


Case 1: Allowable range of heating rates

Case 2: Maximum heating rate enforced

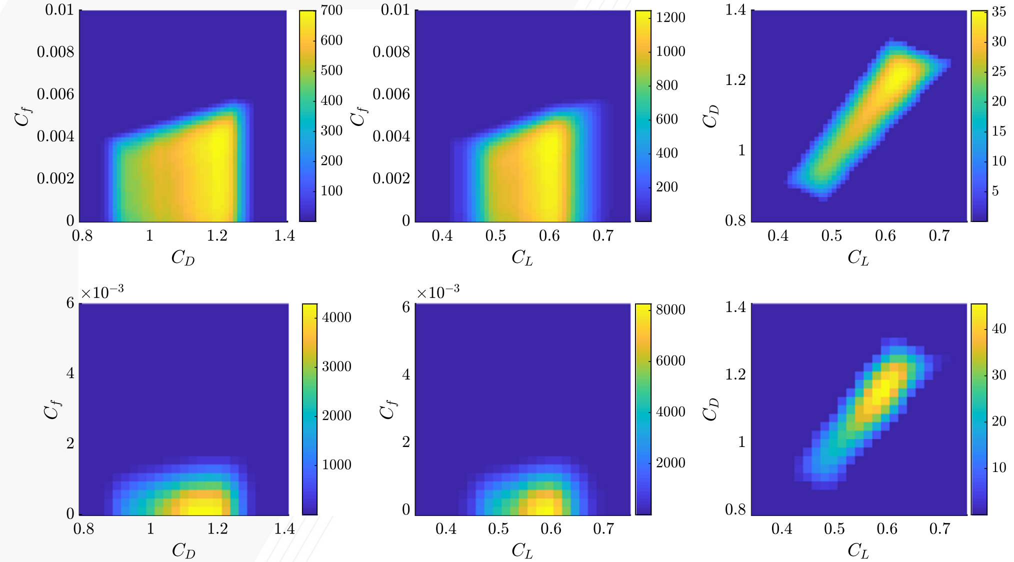
Meyers *et al.*, J. Comp. Phys., 2021.

Inverse Problem Example: Reentry Vehicle



Case 1: Allowable range of heating rates

Case 2: Maximum heating rate enforced



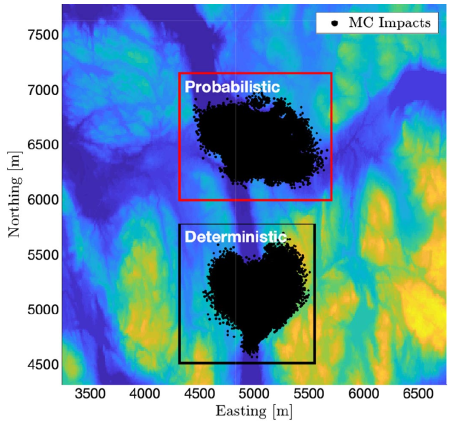
- Multi-dimensional distributions computed using 125,000 points (Case 1) and 15,625 points (Case 2)
- Monte Carlo simulations verify that desired EV constraints were met using computed distributions

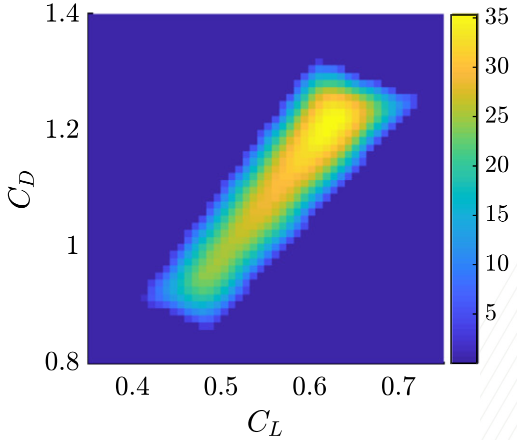
Meyers *et al.*, J. Comp. Phys., 2021.

Conclusion

- Koopman operator provides powerful mechanism for optimization under parametric uncertainty
- Unique computational advantages compared to MC and other explicit UQ methods
- Approach has been demonstrated in optimization of discrete control decisions and initial uncertainty distributions
- Potential extensions to systems with process noise and cases involving optimization of continuous-time controllers

$$\int_{\Omega} \mathrm{P_S} f(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x} = \int_{\Omega} f(\boldsymbol{x}) \mathrm{U_S} g(\boldsymbol{x}) d\boldsymbol{x}$$





Relevant Publications

A. Gerlach, A. Leonard, J. Rogers, C. Rackauckas, The Koopman Expectation: An Operator Theoretic Method for Efficient Analysis and Optimization of Uncertain Hybrid Dynamical Systems," Arxiv Preprint, https://arxiv.org/abs/2008.08737

J. Meyers, J. Rogers, A. Gerlach, "Koopman Operator Method for Solution of Probabilistic Inverse Problems," *Journal of Computational Physics*, Vol. 428, 2021, pp. 1-21.

G. Gutow, J. Rogers, "Koopman Operator Method for Chance-Constrained Motion Primitive Planning," *IEEE Robotics and Automation Letters*, Vol. 5, No. 2, 2020, pp. 1572-1578.

J. Meyers, A. Leonard, J. Rogers, A. Gerlach, "Koopman Operator Approach to Optimal Control Selection Under Uncertainty," 2019 American Control Conference, Philadelphia, PA, July 10-12, 2019.

A. Leonard, J. Rogers, A. Gerlach, "Koopman Operator Approach to Airdrop Mission Planning Under Uncertainty," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 11, 2019, pp. 2382-2398.

A. Leonard, J. Rogers, A. Gerlach, "Probabilistic Release Point Optimization for Airdrop with Variable Transition Altitude," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 8, 2020, pp. 1-11.

Georgia Tech
CREATING THE NEXT