TOYOTA
RESEARCH INSTITUTE

AA203: Optimal and Learning-based Control
Stanford University
May 1, 2026

Uncertainty-Aware Control at the Limits

Thomas Lew

Teamwork

- Extreme Performance Intelligent Control Team
  - John Subosits, Marcus Greiff, Makoto Suminaka, Michael Thompson, Jon Goh, Jenna Lee, Bassamul Haq
  - Alexander Davydov, Emre Adabag, Paul Brunzema

- Platform Research Team
  - Phung Nguyen, Steven Goldine, William Kettle, Geoff Budd, Zach Conybeare, Kazunori Nimura

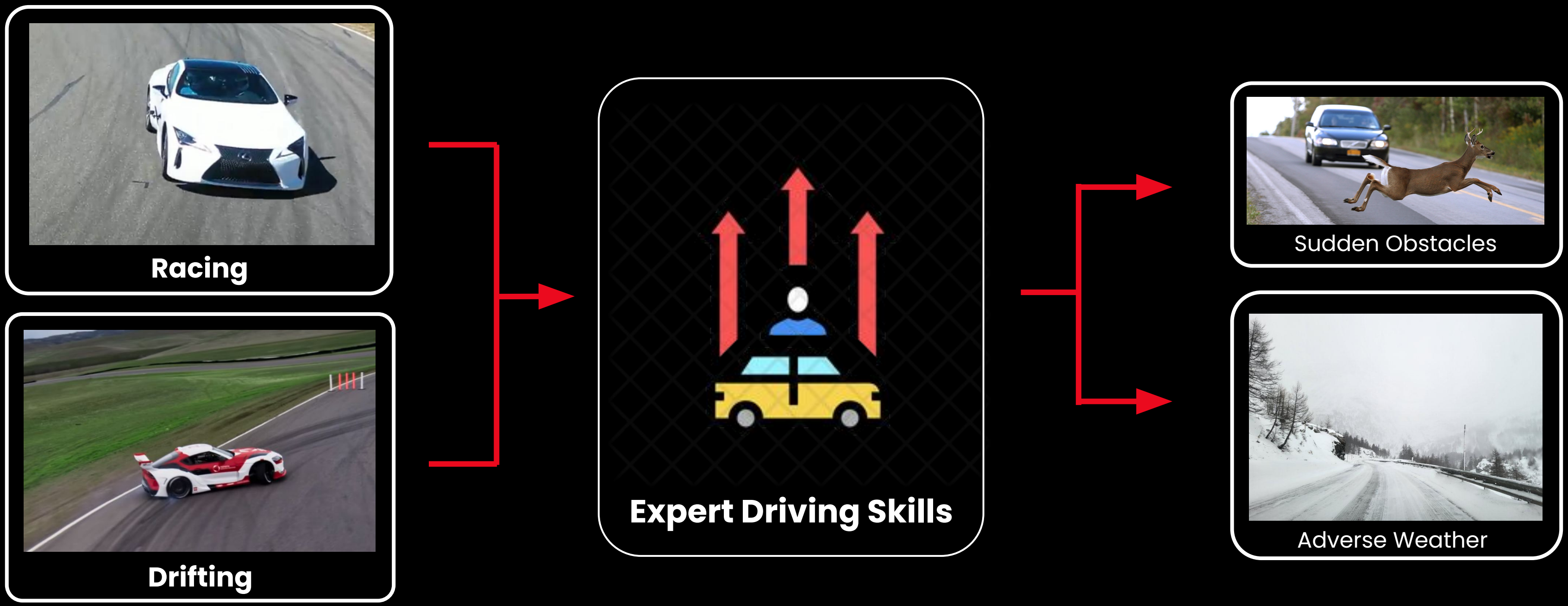
Outline

- Driving at the Limits of Handling

- Optimal Control
    - Shooting Method
    - (Stochastic) Pontryagin Maximum Principle (PMP)
- Combining Learning & Optimal Control

Driving at the Limits of Handling

Towards **expert assistance** on the road

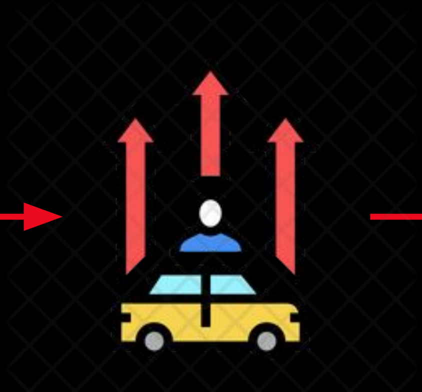


https://www.youtube.com/watch?v=MfU5_gzqPaM

Driver – Vehicle Performance & Safety

Goal: Infuse **expert level** driving skills in our AI

Challenge Problem
Autonomous Drifting



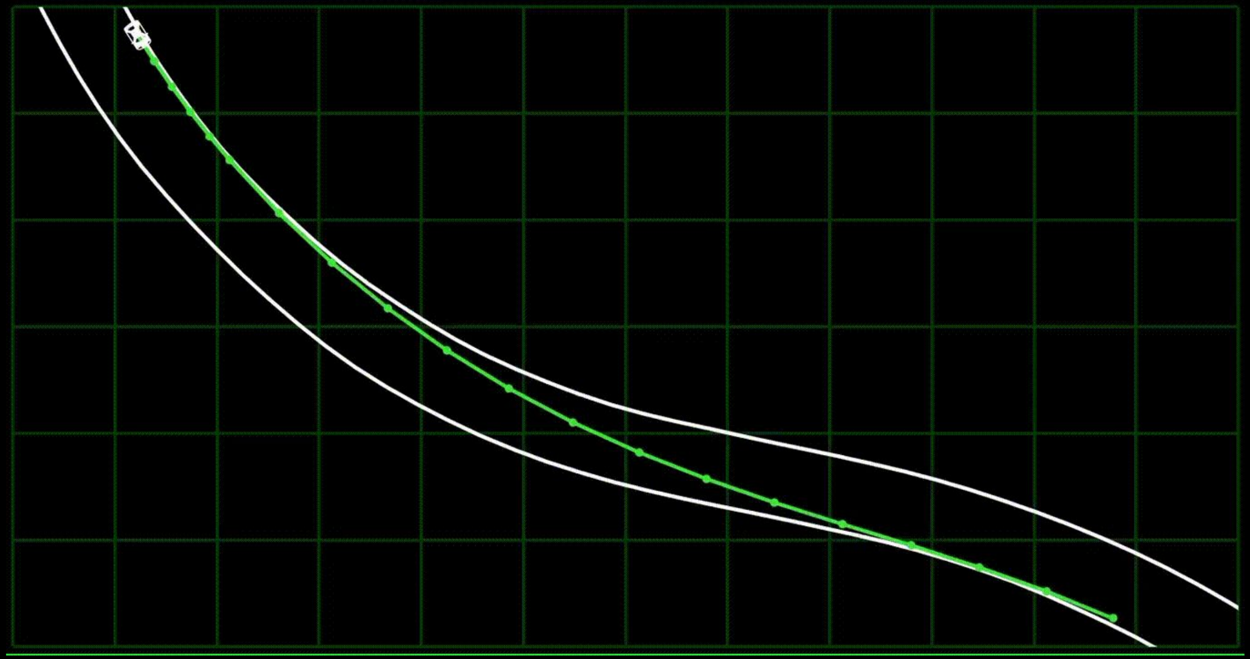
Core Technology:
AI Expert Driving Skills
(Limit Handling)

Avoid Sudden Obstacles

Handle Icy Conditions

Driving at the limits of handling

State of the art methods leverage **model predictive control (MPC)**



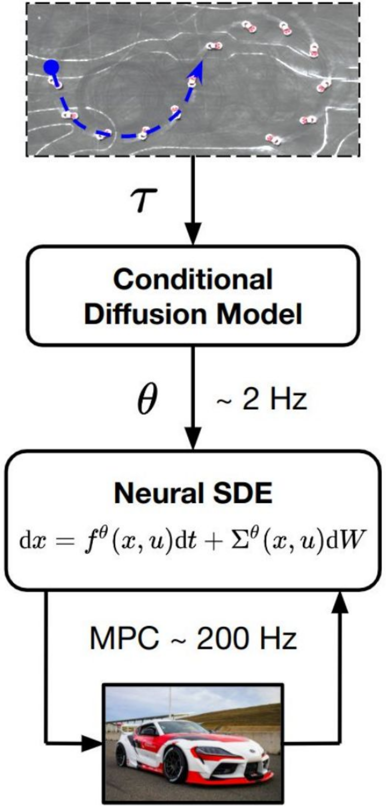
Optimal Control Problem

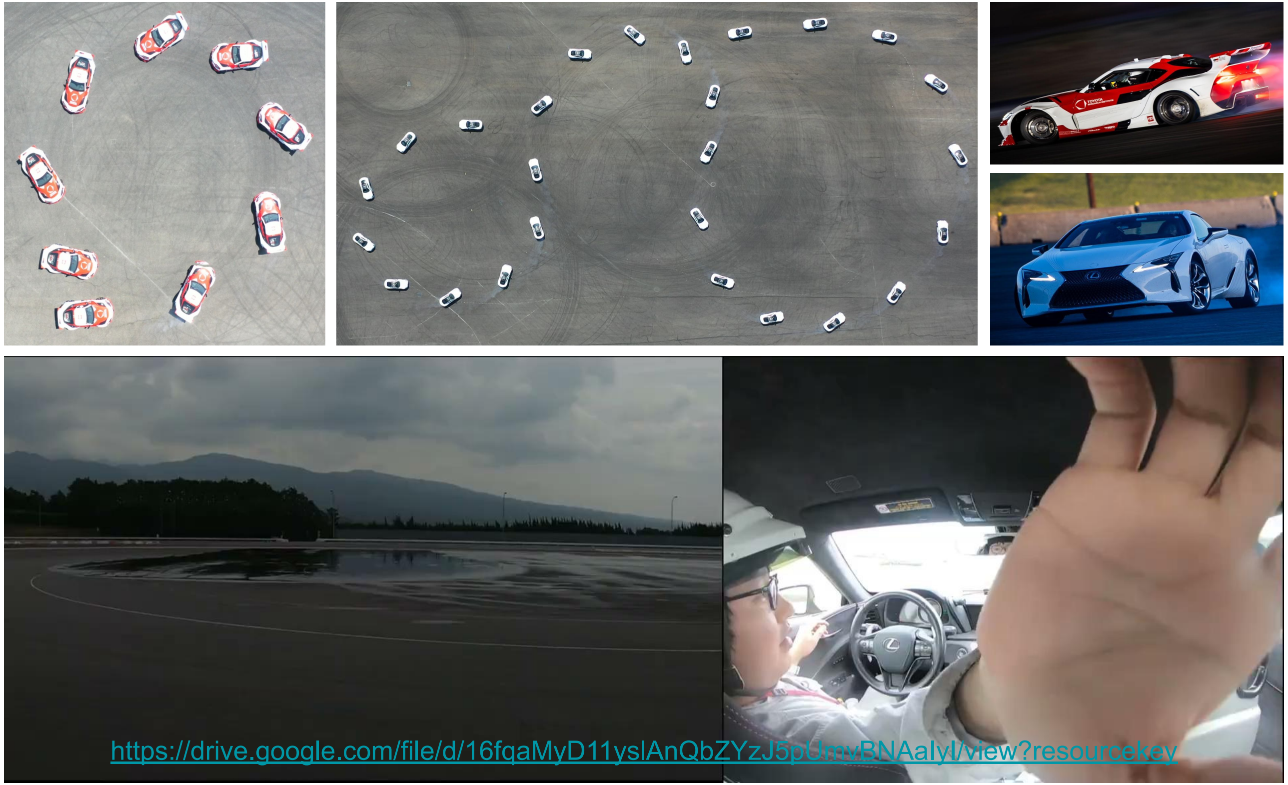
$$\min_{(x,u)} \int_0^T \ell(x_t, u_t)\,\mathrm{d}t$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t$$
$$x_0 = \bar{x}_0$$

solve times ~5-30 ms
(replan at ~30-200Hz)

MPC **generalizes** to different cars & trajectories

MPC is **data-efficient**, thanks to exploiting **structure**

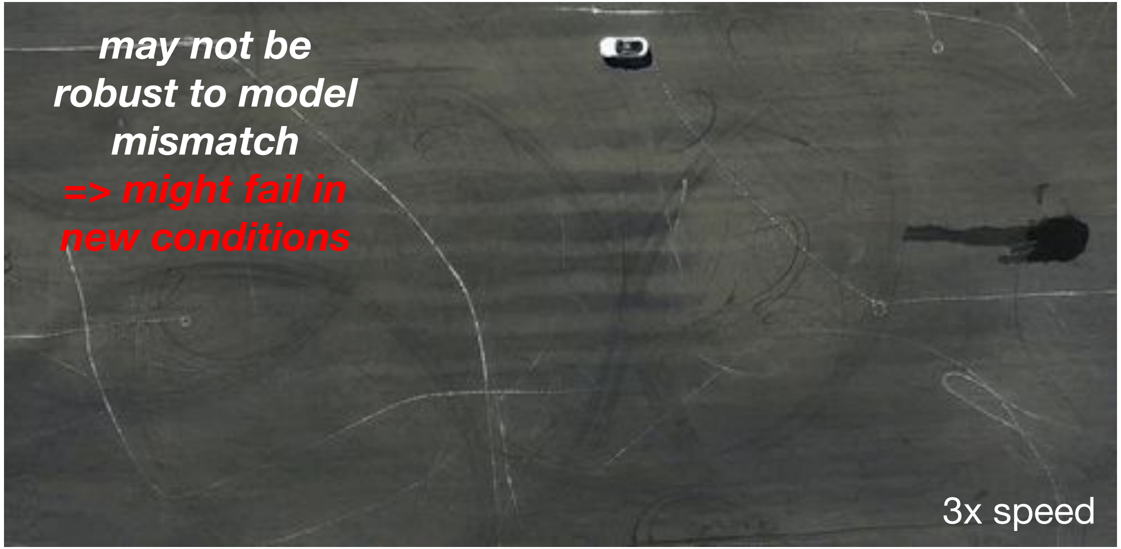




Djeumou, Lew, Ding, Thompson, Suminaka, Greiff, Subosits, "*One Model to Drift Them All*", CoRL'24 (outstanding paper award)

MPC can fail, even with learned models adapting online

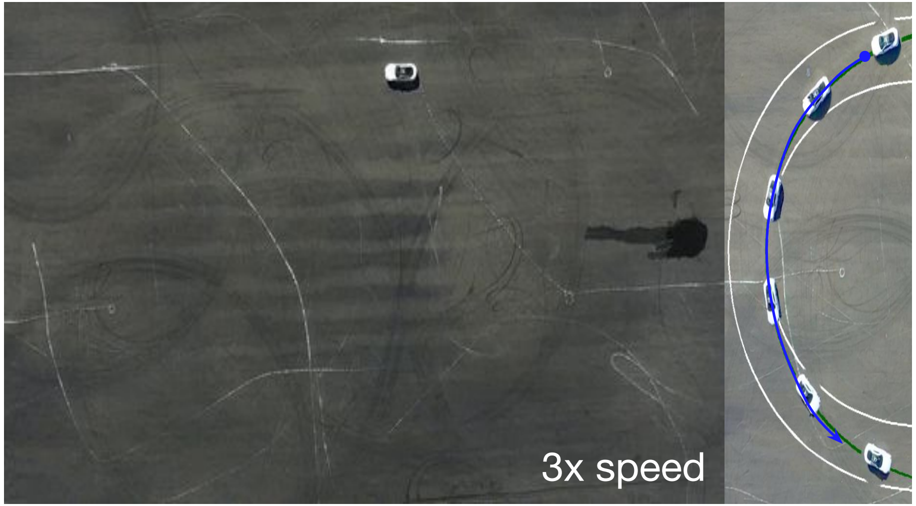
1. Online adaptation is sometimes **too slow**
   Unstable systems are difficult to simultaneously learn & control [Mania'19]
2. MPC often does not account for **uncertainty**
   MPC plans with **one model assumed exact**



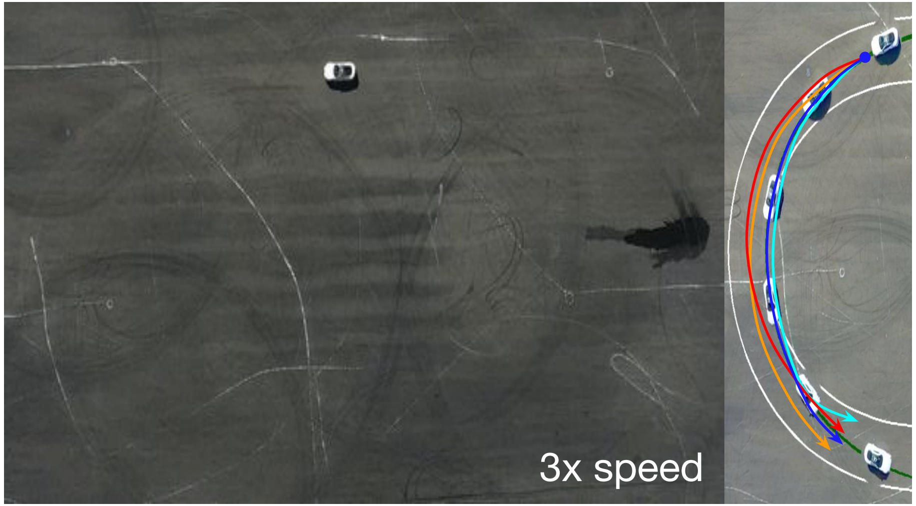
Mania, Tu, Recht. "Certainty Equivalence is Efficient for Linear Quadratic Control," NeurIPS 2019

Accounting for uncertainty increases robustness

MPC



Stochastic MPC



Optimal Control Problem

$$\min_{(x,u)} \quad \int_0^T \ell(x_t, u_t)\,\mathrm{d}t$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t$$

Stochastic Optimal Control Problem

$$\min_{(x,u)} \quad \mathbb{E}\left[\int_0^T \ell(x_t, u_t)\,\mathrm{d}t\right]$$
$$\text{s.t.} \quad \mathrm{d}x_t = \boxed{b(x_t, u_t)\mathrm{d}t} + \boxed{\sigma(x_t)\mathrm{d}B_t}$$

uncertain friction parameters

disturbances

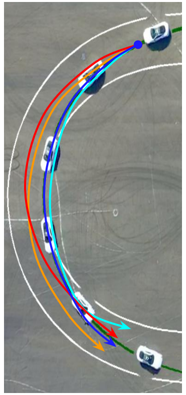
Lew et al. Risk-Averse Model Predictive Control for Racing, *ICRA 2025*

Unlocking Uncertainty-Aware MPC

Overcoming computational barriers by leveraging problem structure

Leveraging Parallel Structure & GPU Acceleration

Risk-Averse Problem



$$\begin{pmatrix} T^1 + \|x^1 - x_{\text{ref}}\| + \\ T^2 + \|x^2 - x_{\text{ref}}\| + \\ T^3 + \|x^3 - x_{\text{ref}}\| + \\ T^4 + \|x^4 - x_{\text{ref}}\| \end{pmatrix}$$

$$\begin{pmatrix} \dot{x}^1 = f(x^1, u, \theta^1) \\ \dot{x}^2 = f(x^2, u, \theta^2) \\ \dot{x}^3 = f(x^3, u, \theta^3) \\ \dot{x}^4 = f(x^4, u, \theta^4) \end{pmatrix}$$

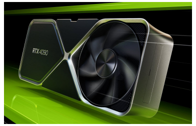
(Initial State)

(Engine & Braking Limits)

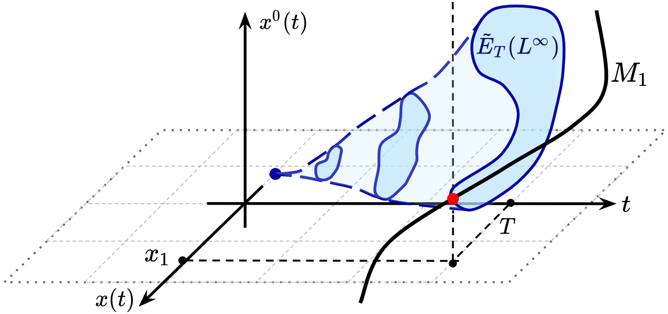
(Stay on the Track)

Solve on the GPU

$$\min_z \; q^\top z + q^\top Q q \quad \text{s.t.} \; l \leq Az \leq u$$



Leveraging Low-Dimensional Characterizations of Solutions



$$\mathrm{d}p_t = -\frac{\partial H}{\partial x}(x_t, u_t, p_t)\mathrm{d}t - \frac{\partial \sigma}{\partial x}(x_t)^\top p_t \mathrm{d}B_t$$

$$u_t = \arg\max_{v \in U} \mathbb{E}\left[H(x_t, v, p_t)\right]$$

Accounting for uncertainty increases robustness

Q: Leveraging **low-dimensional characterizations** of solutions to **efficiently solve** stochastic optimal control problems ?

1. Find informative **optimality conditions**.
2. Use them for **algorithm design**.

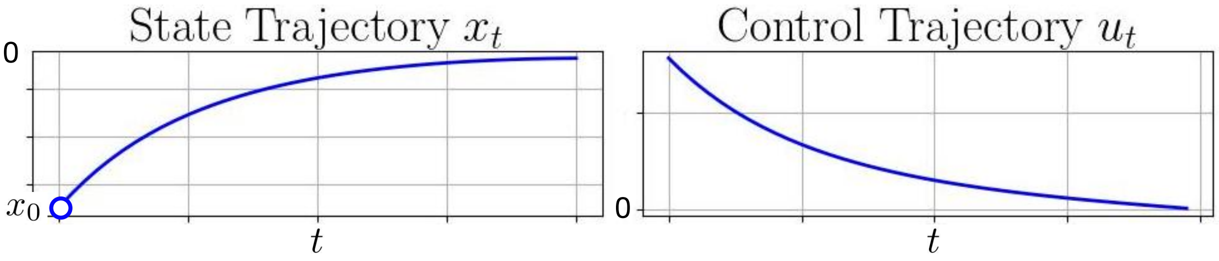
Optimal Control Problem

$$\min_{(x,u)} \quad \int_0^T \ell(x_t, u_t)\,\mathrm{d}t$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t$$

Stochastic Optimal Control Problem

$$\min_{(x,u)} \quad \mathbb{E}\Big[\int_0^T \ell(x_t, u_t)\,\mathrm{d}t\Big]$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \boxed{\sigma(x_t)\mathrm{d}B_t}$$

Optimal Control



Optimal Control Problem

$$\min_{(x,u)} \quad \int_0^T \ell(x_t, u_t)\,\mathrm{d}t$$ cost (e.g., fuel consumption)

$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t$$ dynamics

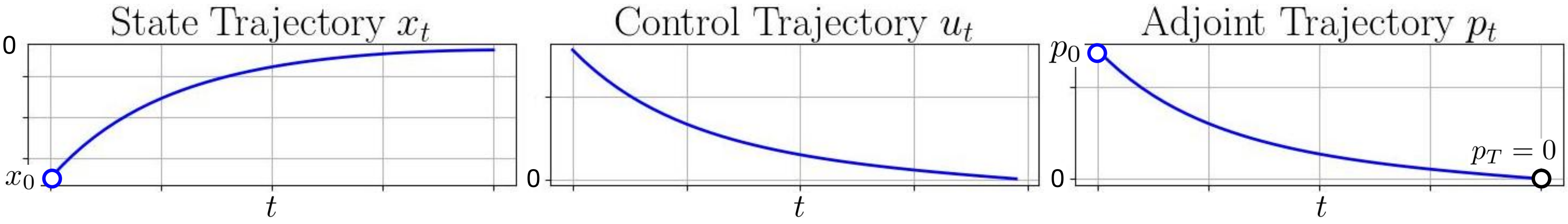
$$x_0 = \bar{x}_0$$ initial state

with $x_t \in \mathbb{R}^n$ state

$u_t \in U \subseteq \mathbb{R}^m$ control input

Optimal Control

solution structure = optimality conditions



Optimal Control Problem

$$\min_{(x,u)} \quad \int_0^T \ell(x_t, u_t)\,\mathrm{d}t$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t$$

Pontryagin Maximum Principle

If (x, u) is optimal, there is an adjoint vector p such that:

$$\mathrm{d}p_t = -\frac{\partial H}{\partial x}(x_t, u_t, p_t)\mathrm{d}t \qquad p_T = 0$$

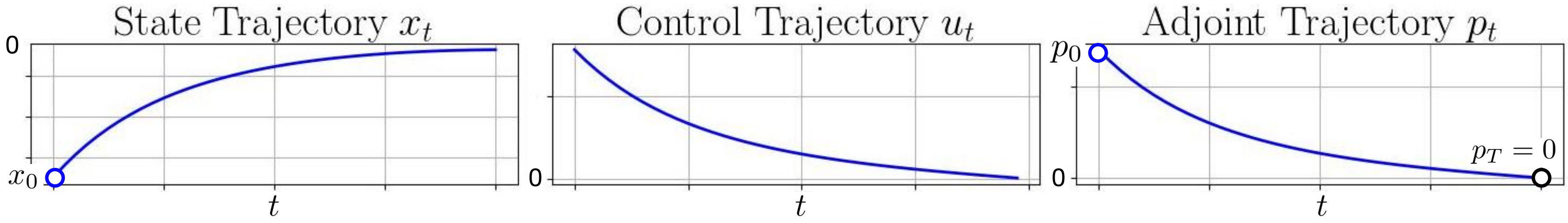
$$u_t = \arg\max_{v \in U} H(x_t, v, p_t)$$

Hamiltonian

$$H(x, u, p) := p^\top b(x, u) - \ell(x, u)$$

Optimal Control

solution structure = optimality conditions



Optimal Control Problem

$$\min_{(x,u)} \quad \int_0^T \ell(x_t, u_t)\,\mathrm{d}t$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t$$

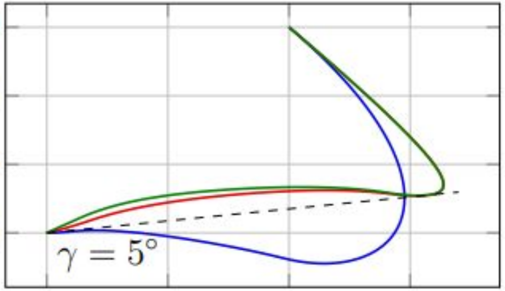
Pontryagin Maximum Principle

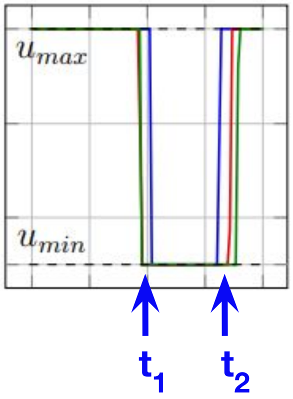
If (x, u) is optimal, there is an adjoint vector p such that:

$$\mathrm{d}p_t = -\frac{\partial H}{\partial x}(x_t, u_t, p_t)\mathrm{d}t \qquad p_T = 0$$

$$u_t = \arg\max_{v \in U} H(x_t, v, p_t)$$

Solutions to Rocket Landing are Bang-Bang 🚀

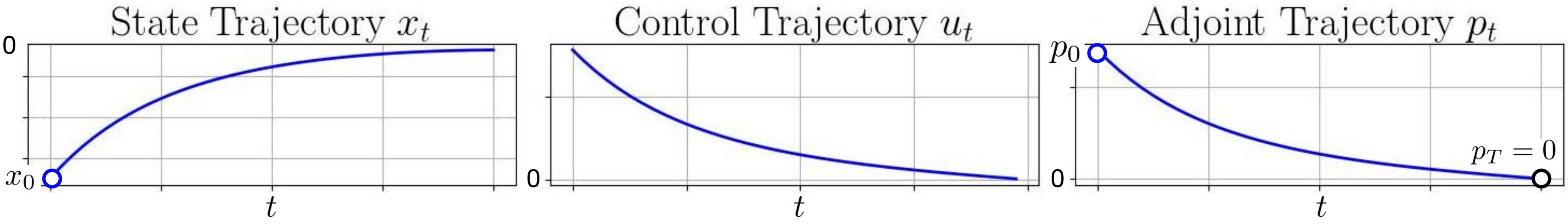




Leparoux, Hérissé & Jean
ESAIM: COCV 2022

Optimal Control

solution structure = optimality conditions



Optimal Control Problem

$$\min_{(x,u)} \int_0^T \ell(x_t, u_t)\,\mathrm{d}t$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t$$

Pontryagin Maximum Principle

If (x, u) is optimal, there is an adjoint vector p such that:

$$\mathrm{d}p_t = -\frac{\partial H}{\partial x}(x_t, u_t, p_t)\mathrm{d}t \qquad p_T = 0$$

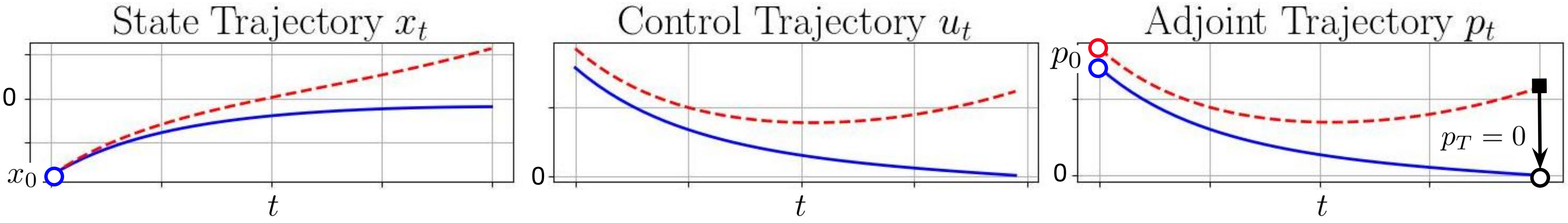
$$u_t = \arg\max_{v \in U} H(x_t, v, p_t)$$

Shooting Method

Run a Newton method (gradient descent) on p_0 until $p_T = 0$.

Optimal Control

solution structure = optimality conditions



Optimal Control Problem

$$\min_{(x,u)} \int_0^T \ell(x_t, u_t)\,\mathrm{d}t$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t$$

Pontryagin Maximum Principle

If (x, u) is optimal, there is an adjoint vector p such that:

$$\mathrm{d}p_t = -\frac{\partial H}{\partial x}(x_t, u_t, p_t)\mathrm{d}t \qquad p_T = 0$$

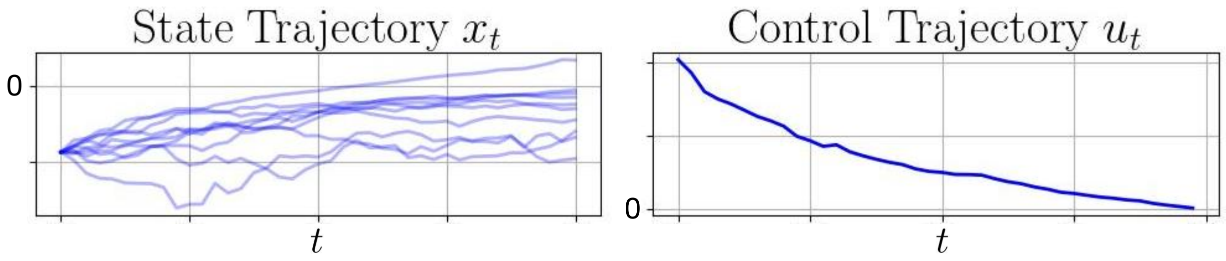
$$u_t = \arg\max_{v \in U} H(x_t, v, p_t)$$

Shooting Method

Run a Newton method (gradient descent) on p_0 until $p_T = 0$.

Stochastic Optimal Control

solution structure = optimality conditions



Optimal Control Problem

$$\min_{(x,u)} \int_0^T \ell(x_t, u_t)\,\mathrm{d}t$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t$$

Stochastic Optimal Control Problem

$$\min_{(x,u)} \mathbb{E}\left[\int_0^T \ell(x_t, u_t)\,\mathrm{d}t\right]$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \boxed{\sigma(x_t)\mathrm{d}B_t}$$

Pontryagin Maximum Principle

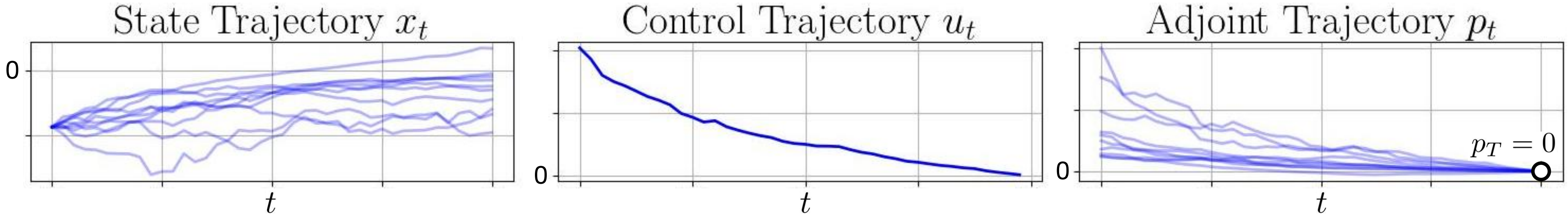
If (x, u) is optimal, there is an adjoint vector p such that:

$$\mathrm{d}p_t = -\frac{\partial H}{\partial x}(x_t, u_t, p_t)\mathrm{d}t \qquad p_T = 0$$

$$u_t = \arg\max_{v \in U} H(x_t, v, p_t)$$

Stochastic Optimal Control

solution structure = optimality conditions



Optimal Control Problem

$$\min_{(x,u)} \int_0^T \ell(x_t, u_t)\,\mathrm{d}t$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t$$

Pontryagin Maximum Principle

If (x, u) is optimal, there is an adjoint vector p such that:

$$\mathrm{d}p_t = -\frac{\partial H}{\partial x}(x_t, u_t, p_t)\mathrm{d}t \qquad p_T = 0$$

$$u_t = \arg\max_{v \in U} H(x_t, v, p_t)$$

Stochastic Optimal Control Problem

$$\min_{(x,u)} \mathbb{E}\left[\int_0^T \ell(x_t, u_t)\,\mathrm{d}t\right]$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \sigma(x_t)\mathrm{d}B_t$$

Stochastic Pontryagin Maximum Principle

If (x, u) is optimal, there are **stochastic** processes (p, q) such that:

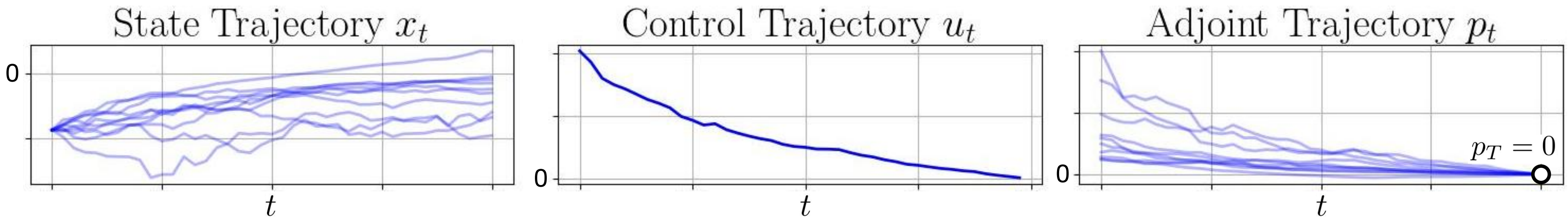
$$\mathrm{d}p_t = -\frac{\partial \widetilde{H}}{\partial x}(x_t, u_t, p_t, q_t)\mathrm{d}t + q_t \mathrm{d}B_t \qquad p_T = 0$$

$$u_t = \arg\max_{v \in U} \mathbb{E}\left[H(x_t, v, p_t)\right]$$

Yong & Zhou, *Stochastic controls (...)*, Springer 1999

Stochastic Optimal Control

solution structure = optimality conditions



Stochastic Optimal Control Problem

$$\min_{(x,u)} \quad \mathbb{E}\left[\int_0^T \ell(x_t, u_t)\,\mathrm{d}t\right]$$

$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \sigma(x_t)\mathrm{d}B_t$$

Stochastic Pontryagin Maximum Principle

If (x, u) is optimal, there are **stochastic** processes (p, q) such that:

$$\mathrm{d}p_t = -\frac{\partial \widetilde{H}}{\partial x}(x_t, u_t, p_t, q_t)\mathrm{d}t + q_t \mathrm{d}B_t \qquad p_T = 0$$

$$u_t = \arg\max_{v \in U} \mathbb{E}\left[H(x_t, v, p_t)\right]$$

Yong & Zhou, *Stochastic controls (...)*, Springer 1999

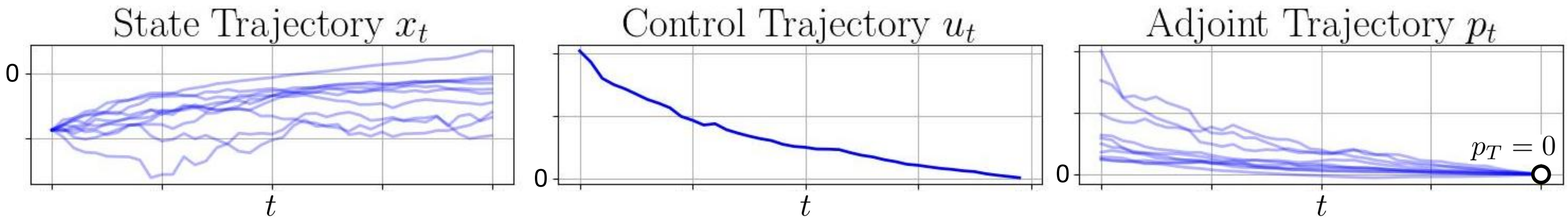
A shooting method is out of reach…

- **Unknowns are** (p, q)
- Forward-backward SDEs are difficult to simulate (except in special cases)
- In practice, use deep learning…

Deep learning for BSDEs: Weinan, Han, Jentzen, 2017

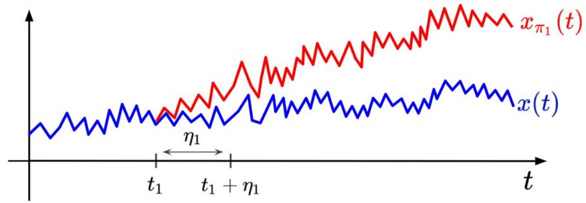
Stochastic Optimal Control

solution structure = optimality conditions



Rough path theory

- **Idea**: Study SDEs *pathwise*



- **Recent** theory of stochastic calculus (**T. Lyons, 1990s**) actively studied today (Friz, Hairer, …)
- Many applications

Stochastic Optimal Control Problem

$$\min_{(x,u)} \quad \mathbb{E}\left[\int_0^T \ell(x_t, u_t)\,\mathrm{d}t\right]$$

$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \boxed{\sigma(x_t)\mathrm{d}B_t}$$

Rough Stochastic Pontryagin Maximum Principle

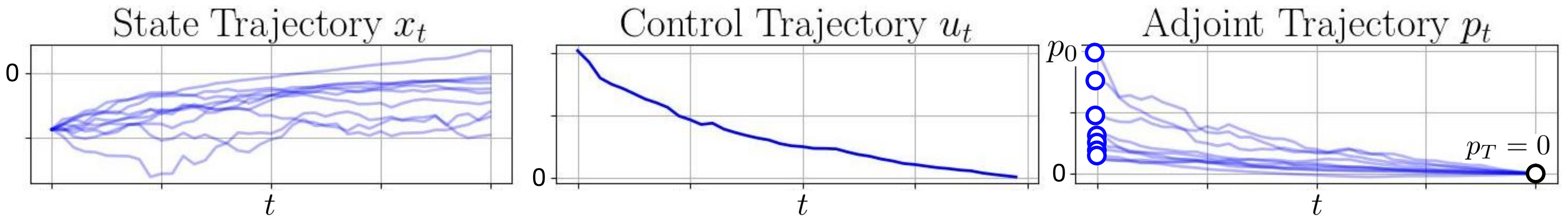
If (x, u) is optimal, there is a **stochastic** adjoint vector p such that:

$$\mathrm{d}p_t = -\frac{\partial H}{\partial x}(x_t, u_t, p_t)\mathrm{d}t \boxed{- \frac{\partial \sigma}{\partial x}(x_t)^\top p_t \mathrm{d}B_t} \qquad p_T = 0$$

$$u_t = \arg\max_{v \in U} \mathbb{E}\left[H(x_t, v, p_t)\right]$$

Stochastic Optimal Control

solution structure = optimality conditions



A Stochastic Indirect Shooting Method!

Use **Monte Carlo** approximation.

Run a **Newton method** (gradient descent) on **samples** of p_0 until $p_T = 0$ for all samples.

Stochastic Optimal Control Problem

$$\min_{(x,u)} \quad \mathbb{E}\left[\int_0^T \ell(x_t, u_t)\,\mathrm{d}t\right]$$

$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \sigma(x_t)\mathrm{d}B_t$$

Rough Stochastic Pontryagin Maximum Principle

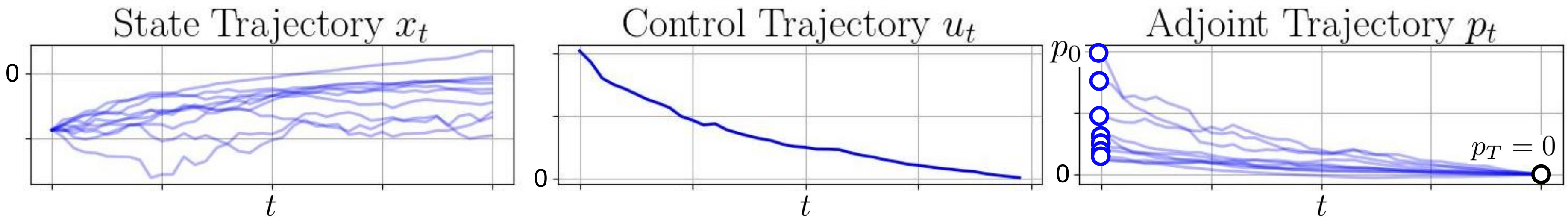
If (x, u) is optimal, there is a **stochastic** adjoint vector p such that:

$$\mathrm{d}p_t = -\frac{\partial H}{\partial x}(x_t, u_t, p_t)\mathrm{d}t - \frac{\partial \sigma}{\partial x}(x_t)^\top p_t \mathrm{d}B_t \qquad p_T = 0$$

$$u_t = \arg\max_{v \in U} \mathbb{E}\left[H(x_t, v, p_t)\right]$$

Stochastic Optimal Control

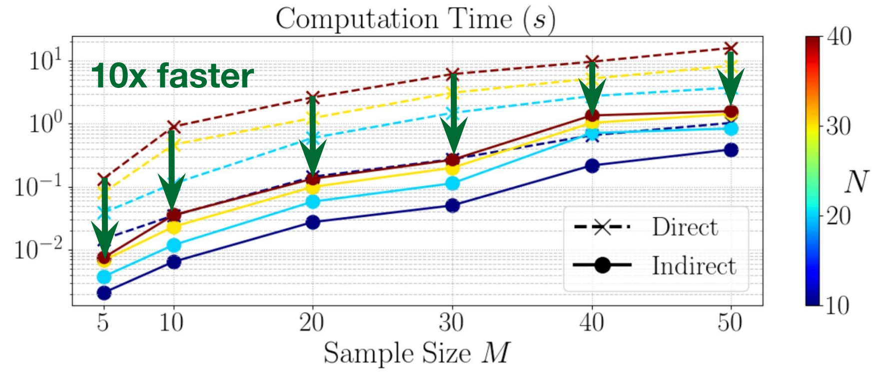
solution structure = optimality conditions



A Stochastic Indirect Shooting Method!

Use **Monte Carlo** approximation.

Run a **Newton method** (gradient descent) on **samples** of p_0 until $p_T = 0$ for all samples.



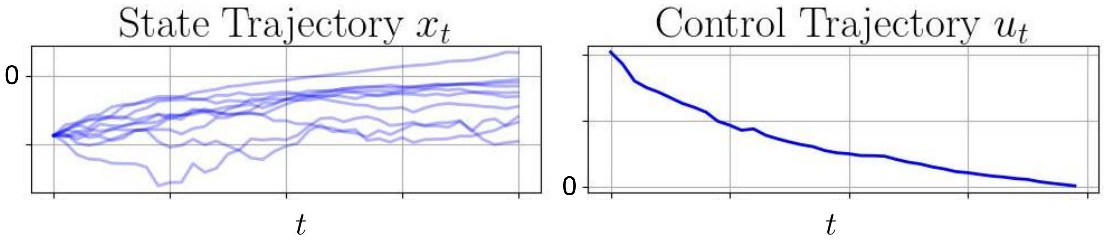
Results on rigid body stabilization task

Stochastic Optimal Control: Proof of the PMP

$$\min_{u \in L^\infty([0,T],U)} \quad \mathbb{E}\left[\int_0^T \ell(x_t, u_t)\,\mathrm{d}t\right]$$

$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \sigma(x_t)\mathrm{d}\mathbf{B}_t, \quad t \in [0,T].$$

- Open-loop controls $u \in L^\infty([0,T],U)$ (deterministic controls)
- Control bounds $u_t \in U$ (e.g., limited engine torque)
- Equality constraints $\mathbb{E}[h(x_T)] = 0$ (e.g., goal reaching)

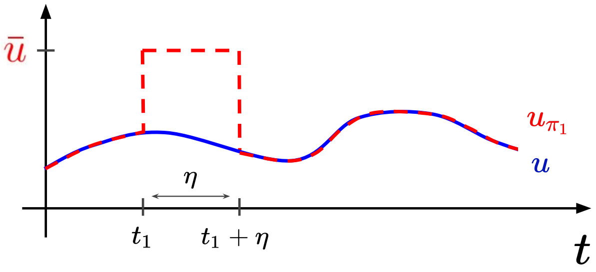


Linearized Rough Differential Equations (RDEs)

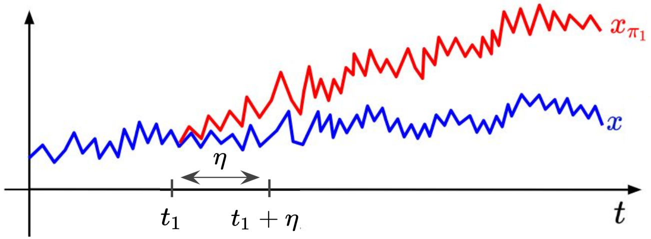
Needle-like variation

Perturb the optimal control

$$u_t^\pi = \begin{cases} \bar{u} & \text{if } t \in [t_1, t_1 + \eta], \\ u_t & \text{otherwise.} \end{cases}$$



=> How do the solutions change?



Error bound: Consider

$$\mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \sigma(x_t)\mathrm{d}\mathbf{B}_t$$

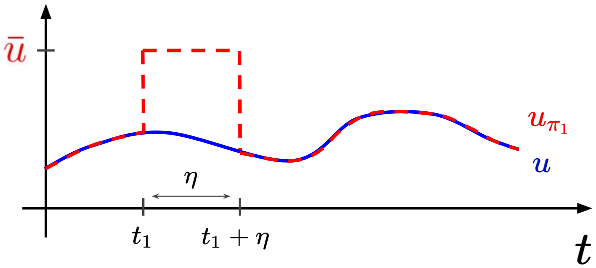
$$\mathrm{d}x_t^\pi = b(x_t^\pi, u_t^\pi)\mathrm{d}t + \sigma(x_t^\pi)\mathrm{d}\mathbf{B}_t$$

Linearized Rough Differential Equations (RDEs)

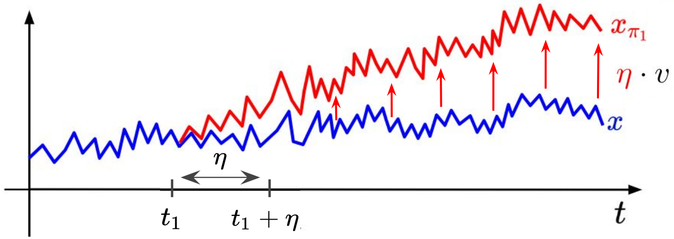
Needle-like variation

Perturb the optimal control

$$u_t^\pi = \begin{cases} \bar{u} & \text{if } t \in [t_1, t_1 + \eta], \\ u_t & \text{otherwise.} \end{cases}$$



=> How do the solutions change?



Error bound: Consider

$$\mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \sigma(x_t)\mathrm{d}\mathbf{B}_t$$

$$\mathrm{d}x_t^\pi = b(x_t^\pi, u_t^\pi)\mathrm{d}t + \sigma(x_t^\pi)\mathrm{d}\mathbf{B}_t$$

and the linearized RDE

$$\mathrm{d}v_t = \nabla b(x_t, u_t)v_t\mathrm{d}t + \nabla\sigma(x_t)v_t\mathrm{d}\mathbf{B}_t$$

from $v_{t_1} = b(x_{t_1}, \bar{u}) - b(x_{t_1}, u_{t_1})$.

Then,

$$\|x^\pi - x\| \leq C_{\mathbf{B}} \cdot \eta$$

$$\|x^\pi - x - \eta \cdot v\| \leq C_{\mathbf{B}} \cdot \eta^2$$

Relies on key results by
Cass, Litterer, Lyons (2013) and Friz & Riedel (2013)

- Integrable bounds
- Directional derivative

Proof of the PMP

Step 1/3) Hyperplane Separation Argument

$$\min_u \quad \mathbb{E}\left[\int_0^T \ell(x_t, u_t)\,\mathrm{d}t\right]$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \sigma(x_t)\mathrm{d}\mathbf{B}_t$$

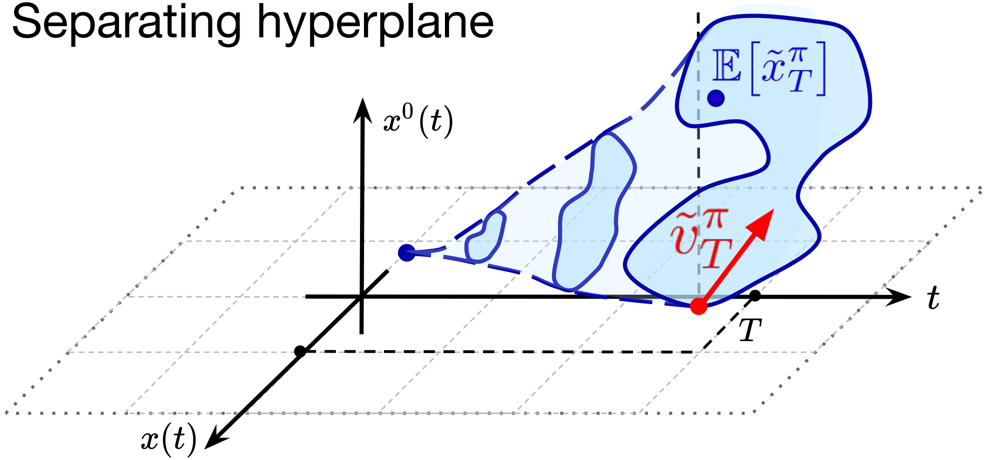
- Cost-augmented system

$$\mathrm{d}\tilde{x}_t = \begin{bmatrix}\mathrm{d}x_t \\ \mathrm{d}x_t^0\end{bmatrix} = \begin{bmatrix}b(x_t, u_t) \\ \ell(x_t, u_t)\end{bmatrix}\mathrm{d}t + \begin{bmatrix}\sigma(x_t) \\ 0\end{bmatrix}\mathrm{d}\mathbf{B}_t$$
$$= \tilde{b}(x_t, u_t)\ \mathrm{d}t + \tilde{\sigma}(x_t)\ \mathrm{d}\mathbf{B}_t$$

Terminal cost value

$$\mathbb{E}\left[x_T^0\right] = \mathbb{E}\left[\int_0^T \ell(x_t, u_t)\mathrm{d}t\right]$$

- Separating hyperplane



Any needle-like variation u_t^π is suboptimal.

Proof of the PMP

Step 1/3) Hyperplane Separation Argument

$$\min_u \quad \mathbb{E}\Big[\int_0^T \ell(x_t,u_t)\,\mathrm{d}t\Big]$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t,u_t)\mathrm{d}t + \sigma(x_t)\mathrm{d}\mathbf{B}_t$$

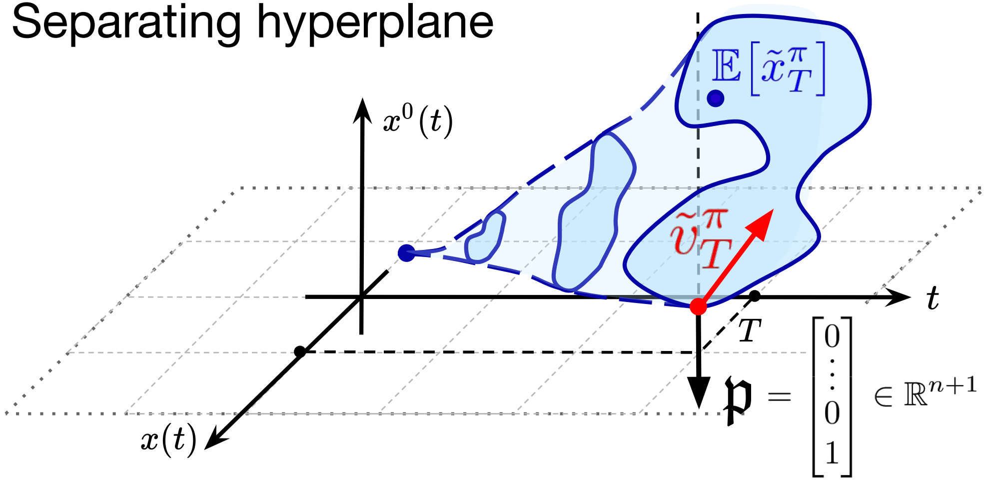
- Cost-augmented system

$$\mathrm{d}\tilde{x}_t = \begin{bmatrix}\mathrm{d}x_t\\ \mathrm{d}x_t^0\end{bmatrix} = \begin{bmatrix}b(x_t,u_t)\\ \ell(x_t,u_t)\end{bmatrix}\mathrm{d}t + \begin{bmatrix}\sigma(x_t)\\ 0\end{bmatrix}\mathrm{d}\mathbf{B}_t$$
$$= \quad \tilde{b}(x_t,u_t)\ \mathrm{d}t + \ \tilde{\sigma}(x_t)\ \mathrm{d}\mathbf{B}_t$$

Terminal cost value

$$\mathbb{E}\left[x_T^0\right] = \mathbb{E}\left[\int_0^T \ell(x_t,u_t)\mathrm{d}t\right]$$

- Separating hyperplane



Any needle-like variation u_t^π is suboptimal, so

$$\mathbb{E}\left[\mathfrak{p}^\top \tilde{v}_T^\pi\right] \leq 0$$

Proof of the PMP

Step 2/3) Adjoint equation

$$\min_u \quad \mathbb{E}\Big[\int_0^T \ell(x_t,u_t)\,\mathrm{d}t\Big]$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t,u_t)\mathrm{d}t + \sigma(x_t)\mathrm{d}\mathbf{B}_t$$

- Adjoint equation, defined backwards in time (compare with Itô & FBSDEs)

$$\mathrm{d}\tilde{p}_t = -\nabla\tilde{b}(\tilde{x}_t,u_t)^\top\tilde{p}_t\mathrm{d}t - \nabla\tilde{\sigma}(\tilde{x}_t)^\top\tilde{p}_t\mathrm{d}\mathbf{B}_t \qquad \tilde{p}_T = \mathfrak{p}$$

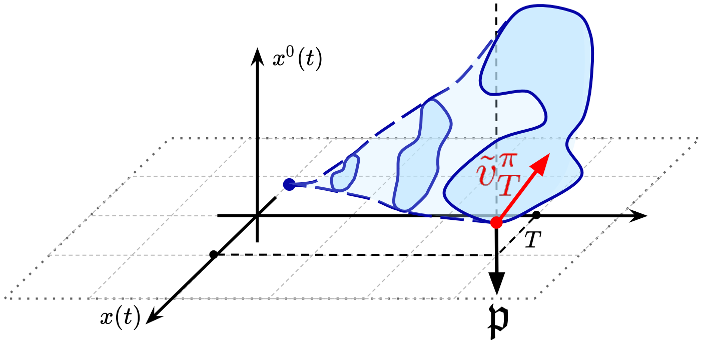
- Recall the variations

$$\mathrm{d}\tilde{v}_t^\pi = \nabla\tilde{b}(\tilde{x}_t,u_t)\tilde{v}_t^\pi\mathrm{d}t + \nabla\tilde{\sigma}(\tilde{x}_t)\tilde{v}_t^\pi\mathrm{d}\mathbf{B}_t$$

- By **Itô's formula** for geometric rough paths:

$$\tilde{p}_t^\top\tilde{v}_t^\pi = \tilde{p}_T^\top\tilde{v}_T^\pi \quad \text{for all times, a.s.}$$

=> $\mathbb{E}\left[\tilde{p}_t^\top\tilde{v}_t^\pi\right] \leq 0$ for all times.



$$\mathbb{E}\left[\mathfrak{p}^\top\tilde{v}_T^\pi\right] \leq 0$$

Proof of the PMP

Step 3/3) Maximality Condition

$$\min_u \quad \mathbb{E}\left[\int_0^T \ell(x_t, u_t)\,\mathrm{d}t\right]$$
$$\text{s.t.} \quad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \sigma(x_t)\mathrm{d}\mathbf{B}_t$$

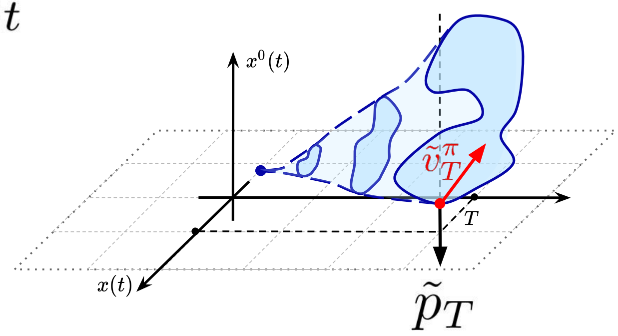
- Adjoint equation

$$\mathrm{d}\tilde{p}_t = -\nabla\tilde{b}(x_t, u_t)^\top \tilde{p}_t \mathrm{d}t - \nabla\tilde{\sigma}(x_t)^\top \tilde{p}_t \mathrm{d}\mathbf{B}_t$$

- Maximality condition

$$u_t = \arg\max_{v \in U} \mathbb{E}\left[H(\tilde{x}_t, v, \tilde{p}_t)\right]$$

with $H(\tilde{x}, u, \tilde{p}) = \tilde{p}^\top \tilde{b}(\tilde{x}, u)$.



By contradiction, find a better control:

$$\mathbb{E}\left[H(\tilde{x}_t, u_t^\pi, \tilde{p}_t) - H(\tilde{x}_t, u_t, \tilde{p}_t)\right]$$
$$= \mathbb{E}\left[\tilde{p}_t^\top\left(\tilde{b}(\tilde{x}_t, u_t^\pi) - \tilde{b}(\tilde{x}_t, u_t)\right)\right] = \mathbb{E}\left[\tilde{p}_t^\top \tilde{v}_t^\pi\right] > 0$$ => contradicts $\mathbb{E}\left[\tilde{p}_t^\top \tilde{v}_t^\pi\right] \leq 0$

Summary: Stochastic PMP

- **Optimal Control via** (pathwise) **rough path theory**

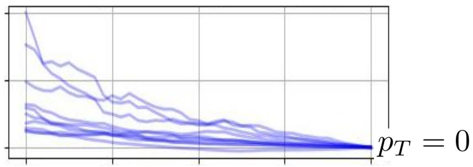
$$\min_{u \in L^\infty([0,T],U)} \quad \mathbb{E}\left[\int_0^T \ell(x_t, u_t)\,\mathrm{d}t\right]$$
$$\text{s.t.} \qquad \mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \sigma(x_t)\mathrm{d}\mathbf{B}_t$$

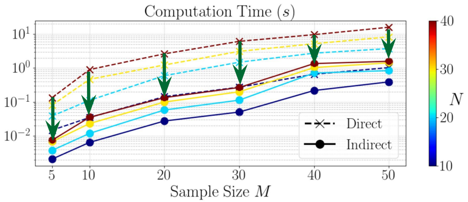
- **Pontryagin Maximum Principle**

$$\mathrm{d}p_t = -\frac{\partial H}{\partial x}(x_t, u_t, p_t)\mathrm{d}t - \frac{\partial \sigma}{\partial x}(x_t)^\top p_t \mathrm{d}B_t \qquad p_T = 0$$

$$u_t = \arg\max_{v \in U} \mathbb{E}\left[H(x_t, v, p_t)\right]$$

- **Shooting Method**



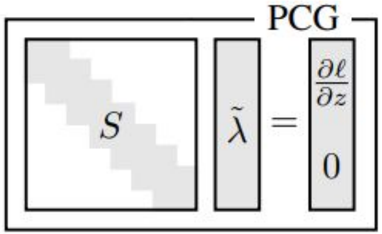


Questions

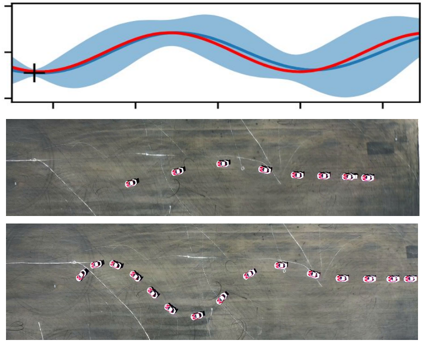
- Other formulations ? (risk-averse, …)
- Control in the diffusion ?
- Analysis of the shooting method ?
- A different path towards extending the deterministic optimal control literature

Augmenting Optimal Control with Learning ?

Differentiable MPC on the GPU (+ RL)



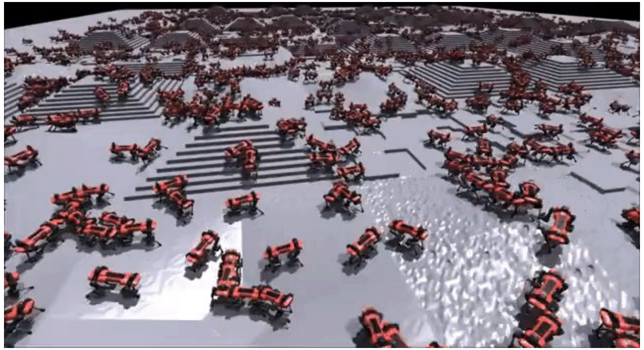
Bayesian Learning



Perception & Control

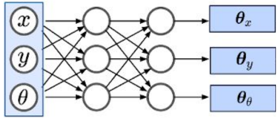
Robustness to Uncertainty via Reinforcement Learning

Reinforcement Learning
in Simulation

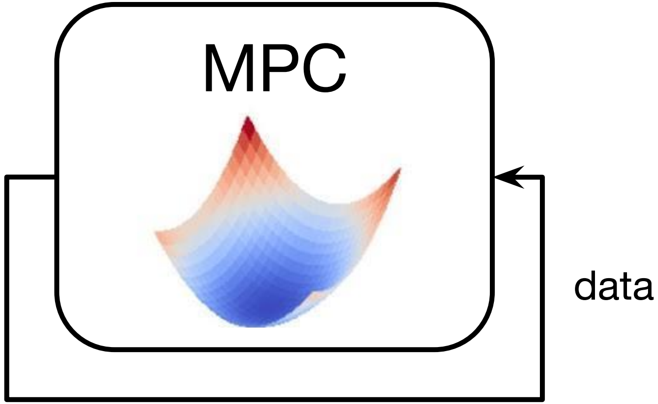


Rudin et al, CoRL'21

Black-Box
Neural Networks

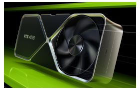


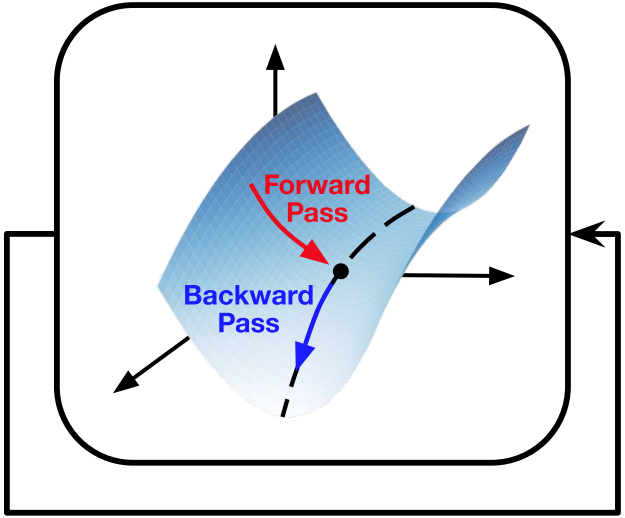
Differentiable MPC as a
structured policy class



- Make MPC **trainable** & combine with learning ?
- Make MPC **scalable** on the GPU ?
- Leverage MPC's **structure** ?

Differentiable **MPC on the** **GPU**

- DiffMPC as a **structured, control-oriented** learning backbone
  github.com/ToyotaResearchInstitute/diffmpc
- Solvers on the **GPU** => we can **scale up**
  - **Hardware-algorithm co-design** 
- DiffMPC + RL to handle uncertainty

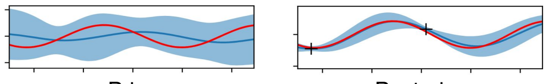


Data
- Reinforcement learning
- Imitation learning
- System identification
- Perception & control
- Auto-tuning

https://www.youtube.com/watch?v=r42iJBw-L4E

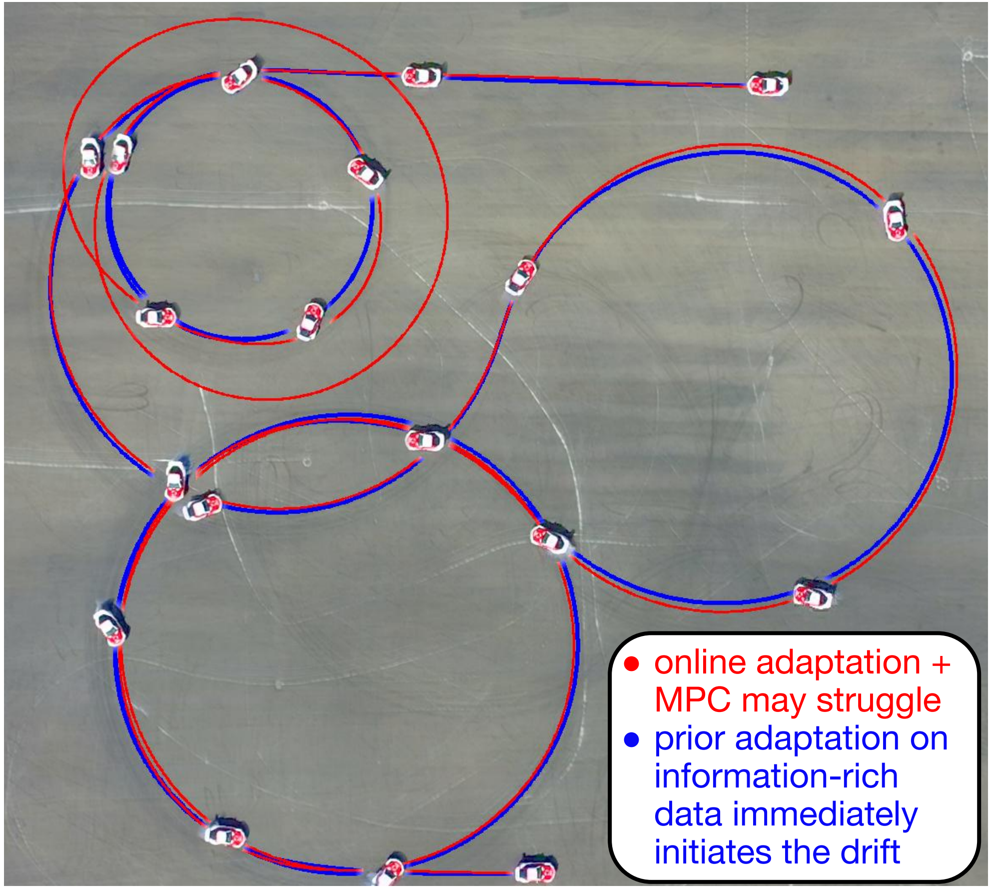
Bayesian Learning & Information Gathering

Bayesian vehicle dynamics model



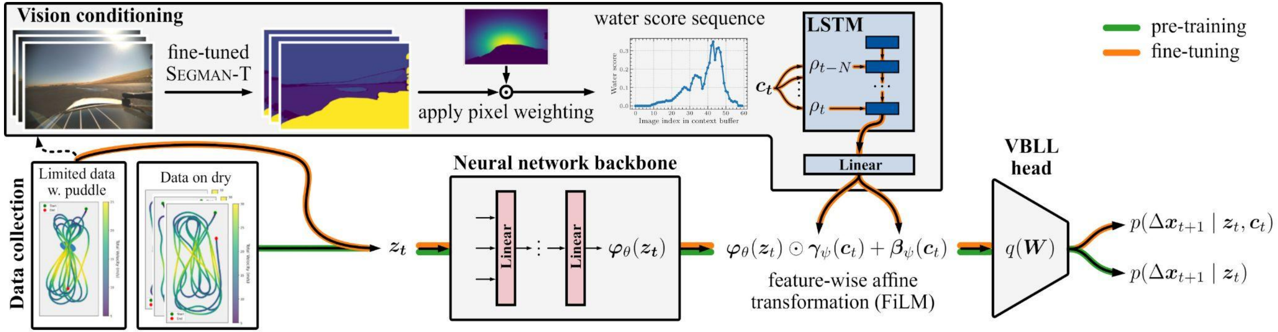
MPC →

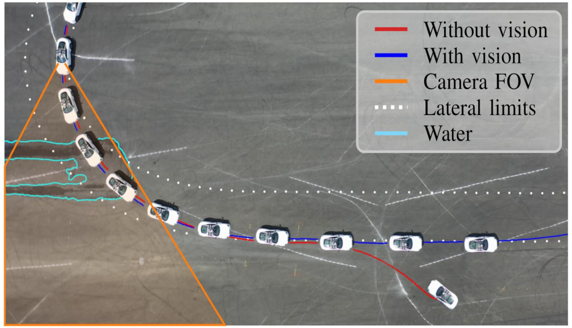
↓ **Active Learning**



Davydov et al, “*First, Learn What You Don’t Know*”, RA-L’25

Vision-Conditioned Bayesian Dynamics Models

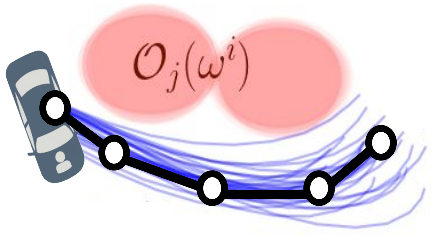




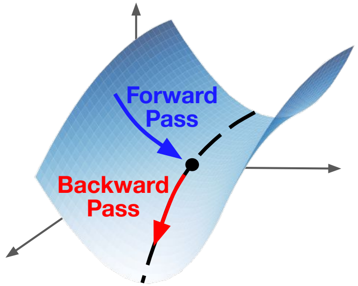
Brunzema et al, “*Vision-Conditioned VBLLs*”, Arxiv’26

Uncertainty-Aware Control at the Limits

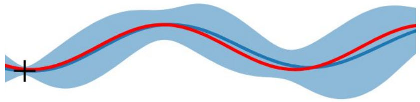
Uncertainty-Aware Optimal Control

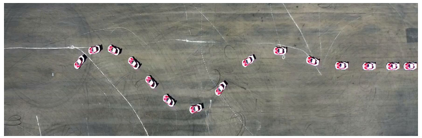


Differentiable MPC on the GPU + RL



Bayesian Learning & Control





Perception & Control

By turning **optimal control** into a **scalable** & **trainable** decision-making engine that **reasons about uncertainty**, we improve the ability of intelligent vehicles to drive safely near their performance limits.

Thank You