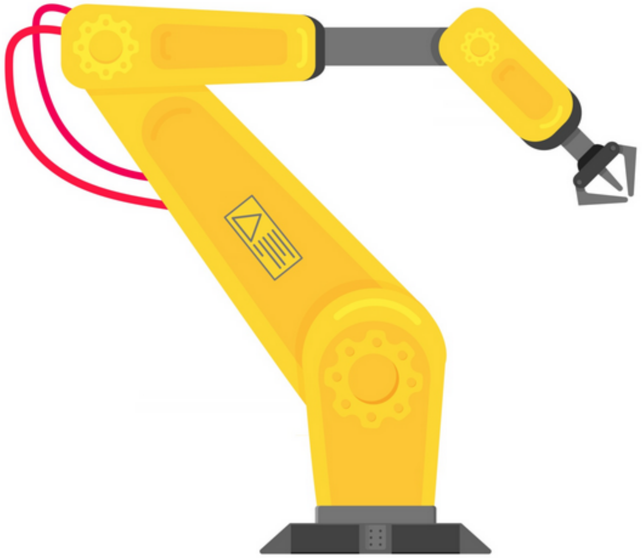
Robust Passivity-Based Control of Underactuated Systems via Neural Approximators and Bayesian Inference

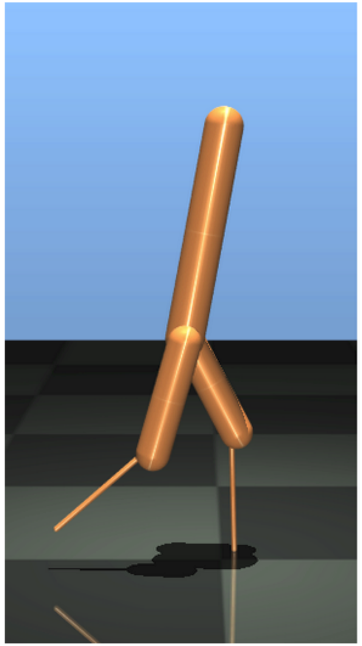
Boise State University

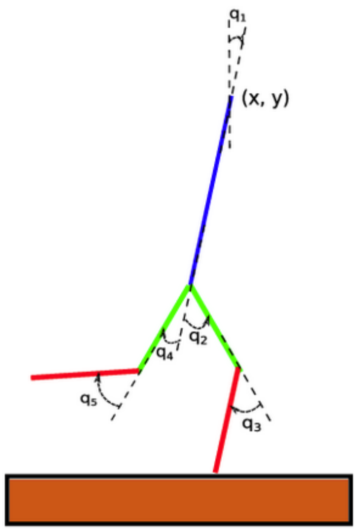
Authors: Nardos Ayele Ashenafi
Wankun Sirichotiyakul, Ph.D.
Aykut C. Satici, Ph.D.

Underactuated Robots



Torque-limited manipulators

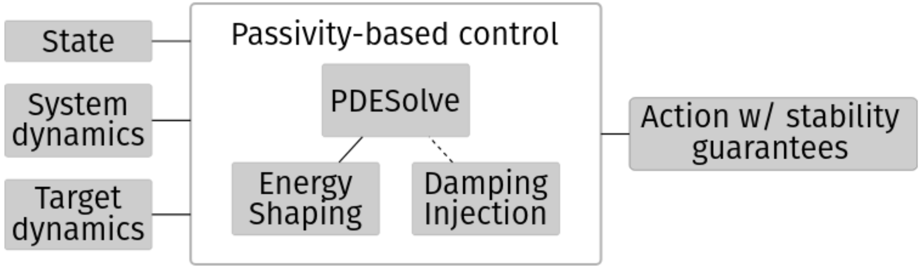




Walking robots

Existing Methods

Passivity-based Control



Strengths

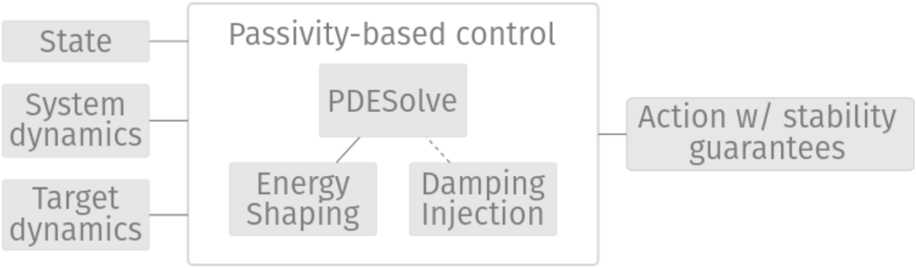
- Stability guarantees
- Closed-form policy

Weaknesses

- Model uncertainties
- Need to solve PDEs

Existing Methods

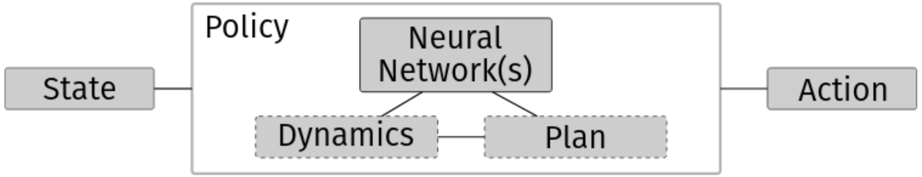
Passivity-based Control



Strengths
- Stability guarantees
- Closed-form policy

Weaknesses
- Model uncertainties
- Need to solve PDEs

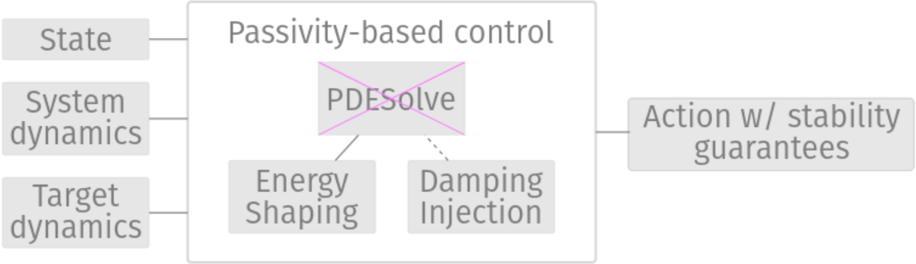
Reinforcement learning



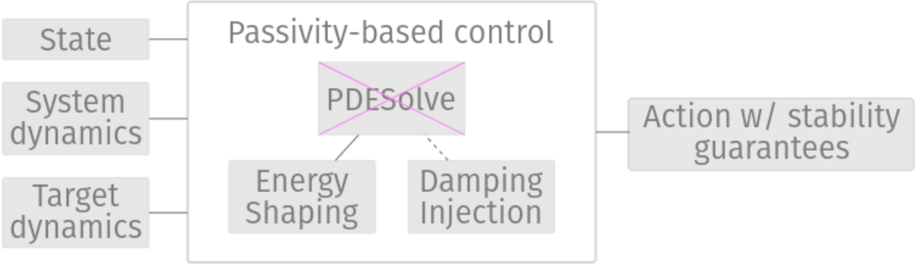
Strengths
- More general
- Unknown dynamics OK

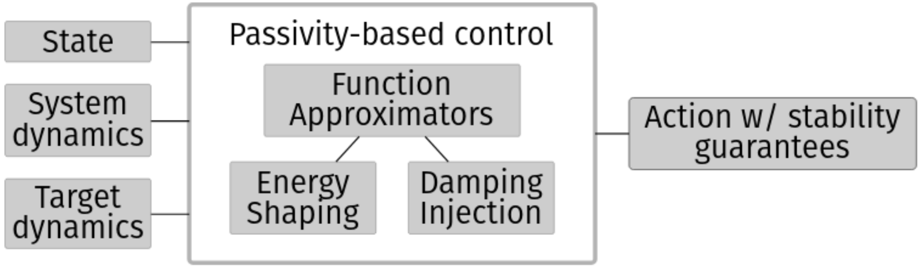
Weaknesses
- Sample complexity
- Stability guarantees?

Data-Driven Passivity-Based Control



Data-Driven Passivity-Based Control





- Systematic approach
- Prior domain knowledge
- Stability is *intrinsic*
- Model uncertainty considerations via Bayesian learning

Background

Passive System Theory and Passivity-Based Control (PBC)

Passivity

A dynamical system

$$\Sigma\colon \quad \begin{cases} \dot{x} &= f(x,u) \\ y &= h(x,u) \end{cases} \qquad x \in \mathcal{X} \subset \mathbb{R}^{2n},\, u \in \mathcal{U} \subset \mathbb{R}^{m}$$

Passivity

A dynamical system

$$\Sigma\colon \quad \begin{cases} \dot{x} &= f(x,u) \\ y &= h(x,u) \end{cases} \qquad x \in \mathcal{X} \subset \mathbb{R}^{2n},\, u \in \mathcal{U} \subset \mathbb{R}^{m}$$

is **dissipative** with respect to some supply rate s if there exists a *storage function* $\mathcal{H}\colon \mathcal{X} \to \mathbb{R}^{+}$ such that

$$\mathcal{H}(x(t_1)) \le \mathcal{H}(x(t_0)) + \int_{t_0}^{t_1} s(u(t), y(t))\,\mathrm{d}t$$

for all $x(t_0) = x_0$, all input u, and all $t_1 \ge t_0$

Passivity

A dynamical system

$$\Sigma\colon \quad \begin{cases} \dot{x} &= f(x,u) \\ y &= h(x,u) \end{cases} \qquad x \in \mathcal{X} \subset \mathbb{R}^{2n},\ u \in \mathcal{U} \subset \mathbb{R}^{m}$$

is **dissipative** with respect to some supply rate s if there exists a *storage function* $\mathcal{H}\colon \mathcal{X} \to \mathbb{R}^{+}$ such that

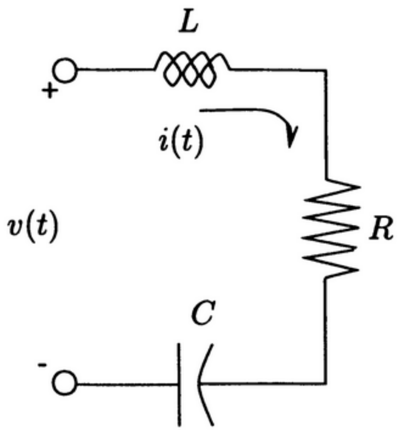
$$\mathcal{H}(x(t_1)) \leq \mathcal{H}(x(t_0)) + \int_{t_0}^{t_1} s(u(t), y(t))\,\mathrm{d}t$$

for all $x(t_0) = x_0$, all input u, and all $t_1 \geq t_0$

The system Σ is **passive** if it is dissipative with supply rate

$$s = u^{\top} y.$$

Passive System Example



Kirchoff's law

$$v = Ri + \tfrac{1}{C}\int_0^t i(\tau)\,\mathrm{d}\tau + L\,\tfrac{\mathrm{d}i}{\mathrm{d}t}$$

$$vi - Ri^2 = \tfrac{\mathrm{d}}{\mathrm{d}t}\left(\underbrace{\tfrac{1}{2C}\left(\int_0^t i(\tau)\,\mathrm{d}\tau\right)^2}_{\mathcal{V}} + \underbrace{\tfrac{1}{2}Li^2}_{\mathcal{T}}\right)$$

Let $\mathcal{H} = \mathcal{V} + \mathcal{T}$, integrate to obtain

$$\underbrace{\mathcal{H}(t)}_{\text{available}} - \underbrace{\mathcal{H}(0)}_{\text{initial}} = \underbrace{\int_0^t v(\tau)i(\tau)\,\mathrm{d}\tau}_{\text{supplied}} - \underbrace{\int_0^t Ri^2(\tau)\,\mathrm{d}\tau}_{\text{dissipated}} < \int_0^t v(\tau)i(\tau)\,\mathrm{d}\tau$$

Stability of Passive Systems

$$\Sigma\colon \quad \begin{cases} \dot{x} &= f(x,u), \qquad\qquad f(0,0) = 0, \\ y &= h(x,u), \qquad\qquad h(0,0) = 0, \end{cases}$$

Lemma (Khalil, 2002)

The origin of Σ is *stable*, i.e

$$y \equiv 0 \implies x \equiv 0$$

if Σ is passive, i.e

$$\mathcal{H} \geq 0, \quad \dot{\mathcal{H}} = \frac{\partial \mathcal{H}}{\partial x} f(x,u) \leq u^\top y$$

Passivity-Based Control (PBC)

$$\Sigma_o\colon \quad \begin{cases} \dot{x} &= f(x) + g(x)u, \\ y &= h(x) \end{cases}$$

Main idea — Select $u(x) = u_{es} + u_{di}$ that renders the closed-loop system passive.

$$\Sigma_d\colon \quad \begin{cases} \dot{x} &= f_d(x) + g(x)u_{di}, \quad f_d := f(x) + g(x)u_{es}(x) \\ y_d &= h_d(x) \end{cases}$$

Control problem is cast as a search for H_d and h_d s.t. $\dot{H}_d \leq y_d^\top u_{di}$

PBC Example - Simple Pendulum

System Dynamics

$H(q,p) = \frac{1}{2}J^{-1}p^2 + mgl(1 - \cos q)$

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \quad , \qquad y = \dot{q}$$

PBC Example - Simple Pendulum

System Dynamics

$H(q,p) = \frac{1}{2} J^{-1} p^2 + mgl(1 - \cos q)$

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \quad , \qquad y = \dot{q}$$

Choose $u = u_{es} + u_{di}$ that transforms system into a passive one with $x^\star = (q^\star, 0)$

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u_{di}, \qquad y = \dot{q}$$

PBC Example - Simple Pendulum

System Dynamics

$H(q,p) = \frac{1}{2} J^{-1} p^2 + mgl(1 - \cos q)$

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \quad , \qquad y = \dot{q}$$

Choose $u = u_{es} + u_{di}$ that transforms system into a passive one with $x^\star = (q^\star, 0)$

$Gu_{es} = \nabla_q H - \nabla_q H_d, \quad Gu_{di} = -GK_D G^\top \nabla_p H_d$

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -GK_D G^\top \end{bmatrix} \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix}, \qquad y = \dot{q}$$

PBC Example - Simple Pendulum

Control Synthesis via PBC

Choose $u = u_{es} + u_{di}$ that transforms system into a passive one with $x^\star = (q^\star, 0)$

$G u_{es} = \nabla_q H - \nabla_q H_d, \quad G u_{di} = -G K_D G^\top \nabla_p H_d$

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -G K_D G^\top \end{bmatrix} \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix}, \qquad y = \dot{q}$$

PBC Example - Simple Pendulum

Control Synthesis via PBC

Choose $u = u_{es} + u_{di}$ that transforms system into a passive one with $x^\star = (q^\star, 0)$

$Gu_{es} = \nabla_q H - \nabla_q H_d, \quad Gu_{di} = -GK_D G^\top \nabla_p H_d$

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -GK_D G^\top \end{bmatrix} \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix}, \qquad y = \dot{q}$$

$H_d(q,p) = \frac{1}{2} J^{-1} p^2 + V_d(q), \quad V_d(q) = \frac{1}{2} K_P (q - q^\star)^2$

PBC Example - Simple Pendulum

Control Synthesis via PBC

Choose $u = u_{es} + u_{di}$ that transforms system into a passive one with $x^\star = (q^\star, 0)$

$Gu_{es} = \nabla_q H - \nabla_q H_d, \quad Gu_{di} = -GK_D G^\top \nabla_p H_d$

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -GK_D G^\top \end{bmatrix} \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix}, \qquad y = \dot{q}$$

$H_d(q,p) = \frac{1}{2} J^{-1} p^2 + V_d(q), \quad V_d(q) = \frac{1}{2} K_P (q - q^\star)^2$

$\dot{H}_d = -K_D \left(J^{-1} p\right)^2 = y^\top u_{di} \le 0$

PBC Example - Simple Pendulum

Control Synthesis via PBC

Choose $u = u_{es} + u_{di}$ that transforms system into a passive one with $x^\star = (q^\star, 0)$

$Gu_{es} = \nabla_q H - \nabla_q H_d, \quad Gu_{di} = -GK_D G^\top \nabla_p H_d$

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -GK_D G^\top \end{bmatrix} \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix}, \qquad y = \dot{q}$$

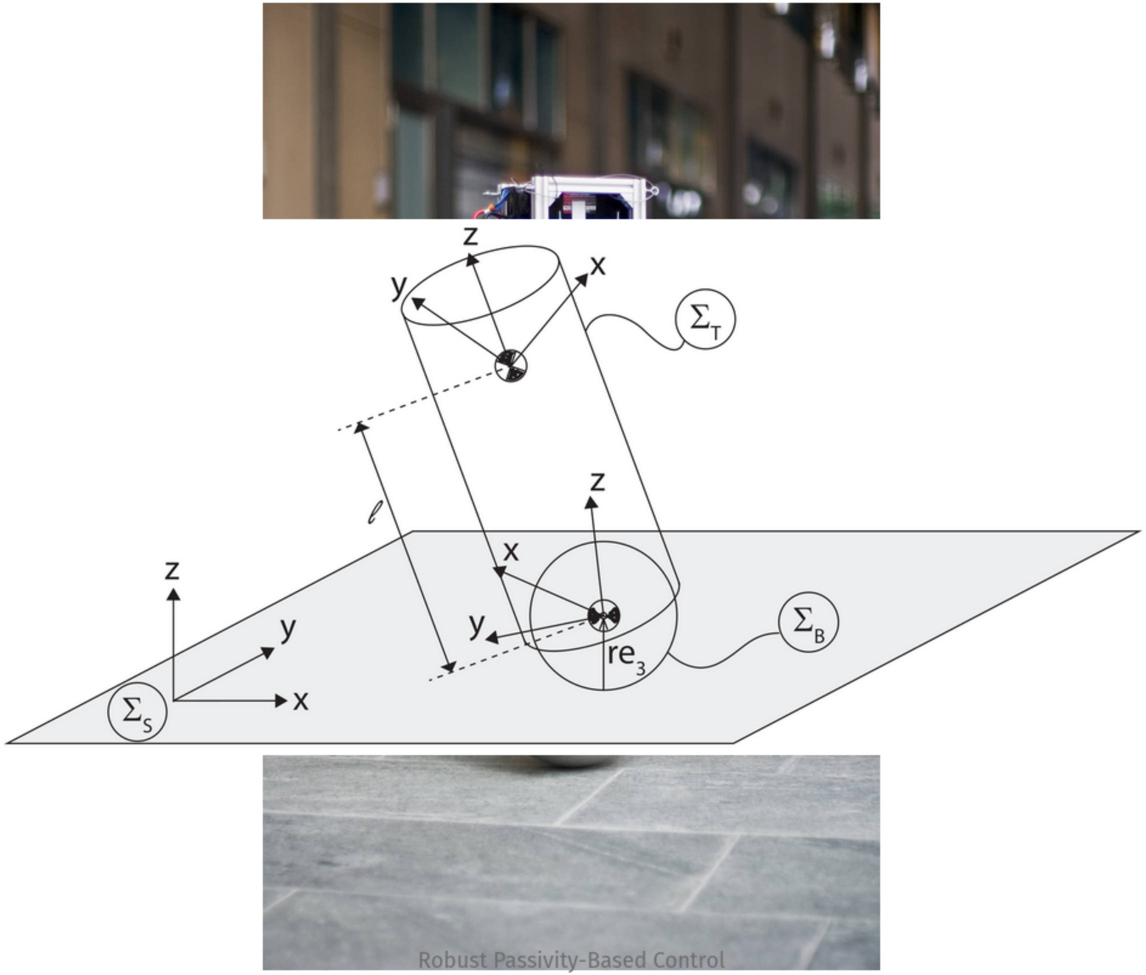
$H_d(q,p) = \frac{1}{2} J^{-1} p^2 + V_d(q), \quad V_d(q) = \frac{1}{2} K_P (q - q^\star)^2$

$\dot{H}_d = -K_D \left(J^{-1} p\right)^2 = y^\top u_{di} \leq 0$

$\boxed{u = -mgl \sin(x) - K_P(q - q^\star) - K_D \dot{q}}$

High-dimensional Problem: Ballbot

High-dimensional Problem: Ballbot



High-dimensional Problem: Ballbot

$$\dot{H}_d = (k_1 y_1 + k_2 y_2)\Bigg[\left(k_e + k_1 k_k + k_2 k_k \frac{(\alpha + \beta \cos(x))^2}{\alpha + \gamma + 2\beta \cos(x)} \right) u$$

$$+ k_2 k_k \left(-(\alpha + \beta c_x) \left(\frac{\beta s_x}{\alpha + \gamma + 2\beta c_x} \dot{x}^2 \right.\right.$$

$$\left.\left. + \frac{\mu s_x}{\alpha + \gamma + 2\beta c_x} \right) + \beta s_x \dot{x}^2 \right)$$

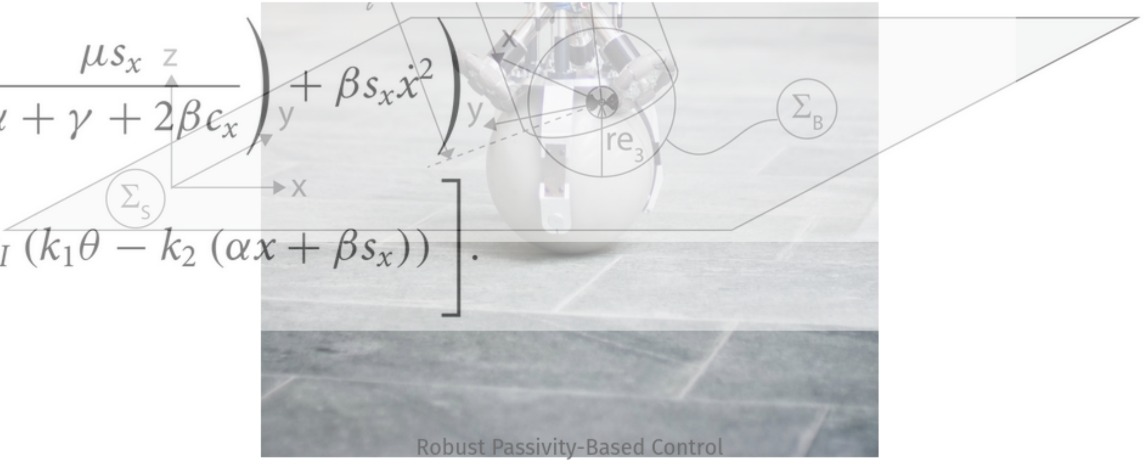
$$+ k_I \left(k_1 \theta - k_2 (\alpha x + \beta s_x) \right) \Bigg].$$

High-dimensional Problem: Ballbot

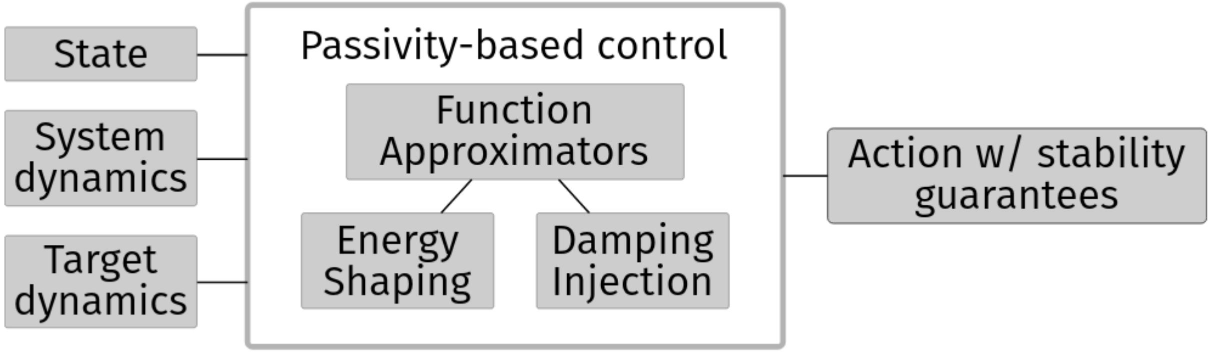
$$u = -\frac{1}{k}\Bigg[k_2 k_k\Bigg(-(\alpha+\beta c_x)\Bigg(\frac{\beta s_x}{\alpha+\gamma+2\beta c_x}\dot{x}^2 + \frac{\mu s_x}{\alpha+\gamma+2\beta c_x}\Bigg) + \beta s_x \dot{x}^2\Bigg) + k_I\left(k_1\theta - k_2\left(\alpha x + \beta s_x\right)\right) + k_p\left(k_1 y_1 + k_2 y_2\right)\Bigg]$$

$$\dot{H}_d = (k_1 y_1 + k_2$$

$$+ k_2 k_k\Bigg(- \quad + \frac{\mu s_x}{\alpha+\gamma+2\beta c_x}\Bigg) + \beta s_x \dot{x}^2\Bigg) + k_I\left(k_1\theta - k_2\left(\alpha x + \beta s_x\right)\right)\Bigg].$$

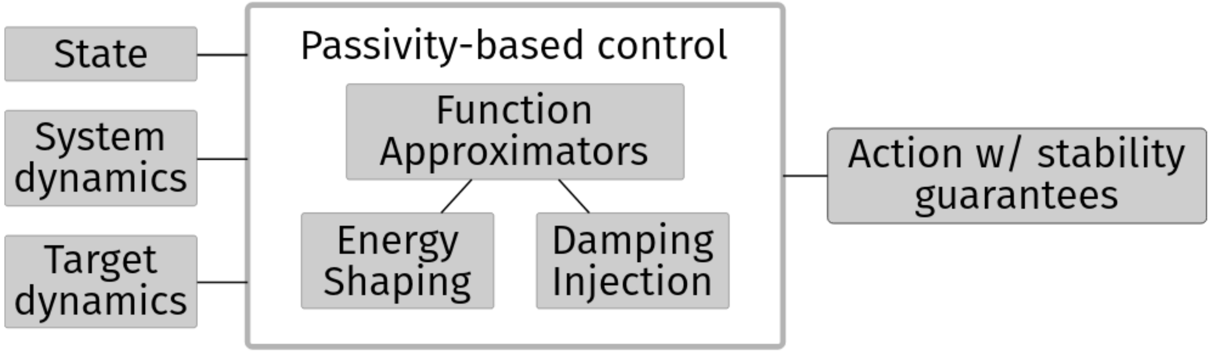


Our Methods



State
System dynamics
Target dynamics
Passivity-based control
Function Approximators
Energy Shaping
Damping Injection
Action w/ stability guarantees

Our Methods



Data-Driven Passivity-based control

$$\underset{\theta}{\text{minimize}} \quad J(\theta, x_0) = \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t$$

$$\text{subject to} \quad \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u^\theta$$

$$u^\theta = -G^\dagger \nabla_q H_d^\theta - K_D G^\top \nabla_p H_d^\theta$$

NeuralPbc Problem Statement

Data-Driven Passivity-based control

$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \qquad & J(\theta, x_0) && = \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \qquad & \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} && = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u^\theta \\
& u^\theta && = -G^\dagger \nabla_q H_d^\theta - K_D G^\top \nabla_p H_d^\theta
\end{aligned}
$$

NeuralPbc Problem Statement

Data-Driven Passivity-based control

$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \quad & J(\theta, x_0) && = \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \quad & \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} && = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u^\theta \\
& u^\theta && = -G^\dagger \nabla_q H_d^\theta - K_D G^\top \nabla_p H_d^\theta
\end{aligned}
$$

- Sampling the state space efficiently

NeuralPbc Problem Statement

Data-Driven Passivity-based control

$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \qquad J(\theta, x_0) &= \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \qquad \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u^\theta \\
u^\theta &= -G^\dagger \nabla_q H_d^\theta - K_D G^\top \nabla_p H_d^\theta
\end{aligned}
$$

- Sampling the state space efficiently
- Injecting control task into loss function design

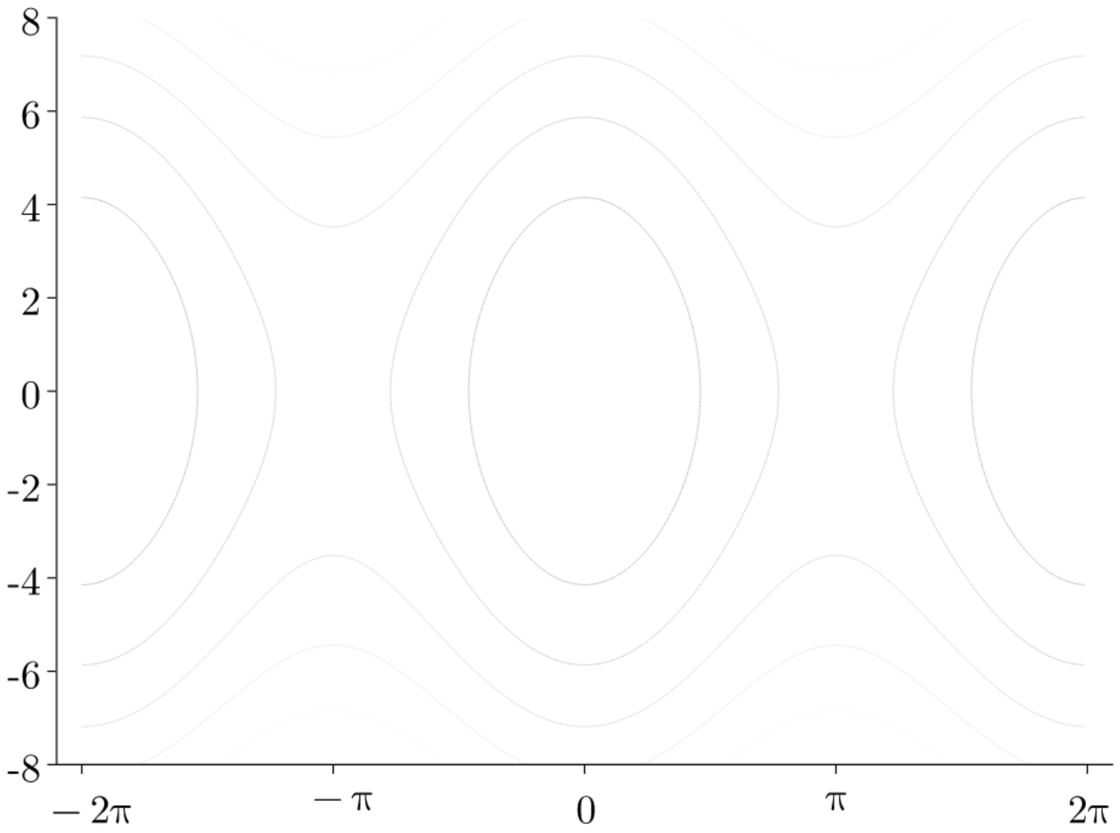
NeuralPbc Problem Statement

Data-Driven Passivity-based control

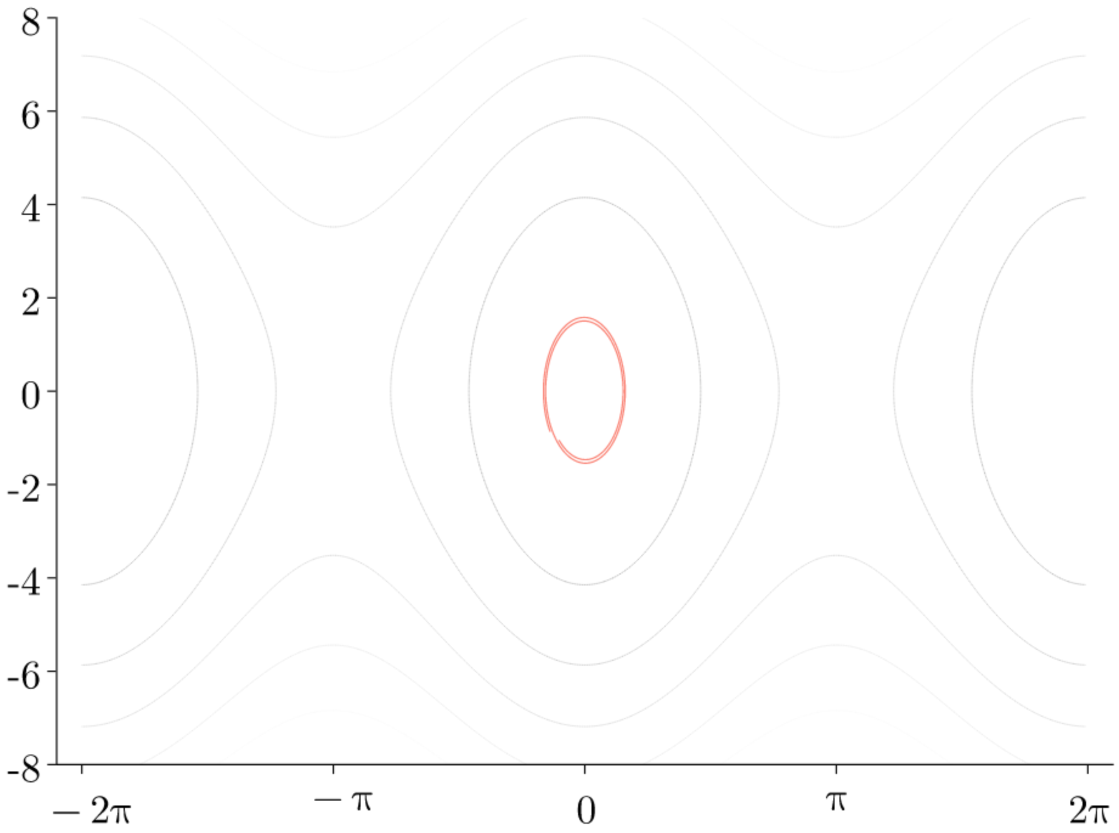
$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \qquad J(\theta, x_0) &= \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \qquad \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u^\theta \\
u^\theta &= -G^\dagger \nabla_q H_d^\theta - K_D G^\top \nabla_p H_d^\theta
\end{aligned}
$$

- Sampling the state space efficiently
- Injecting control task into loss function design
- Backprop through closed-loop trajectories

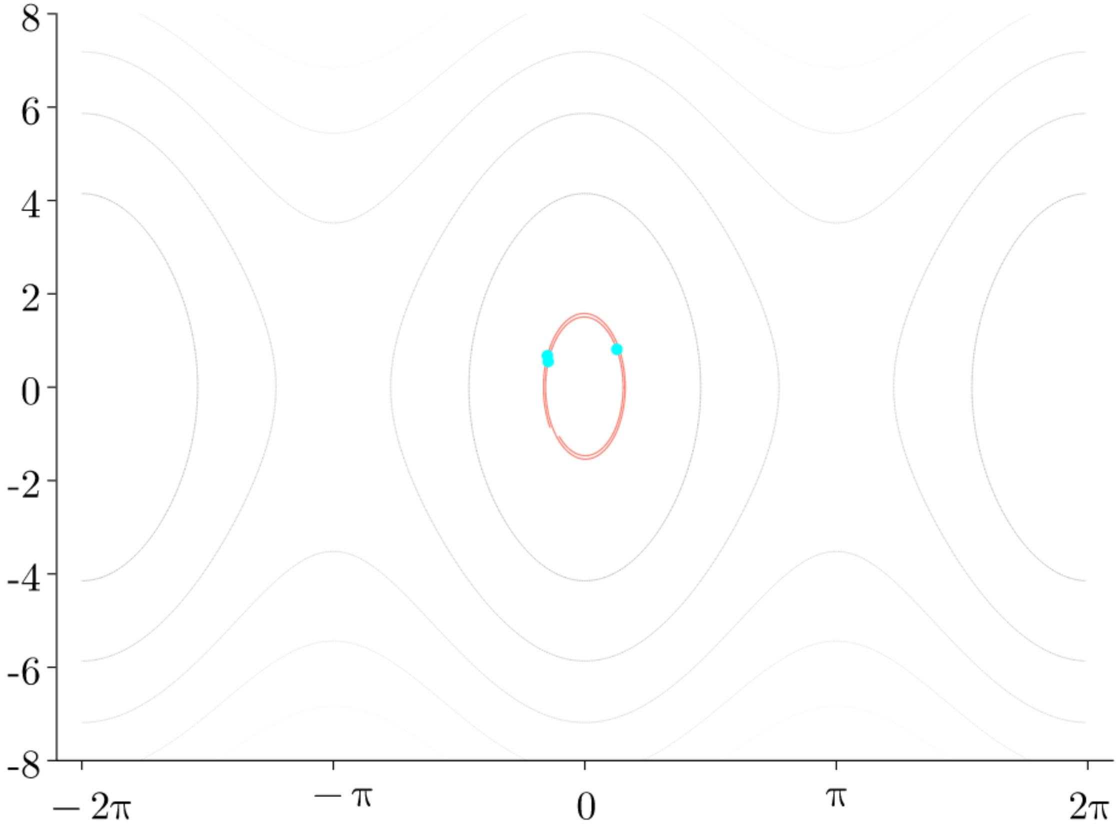
NEURALPBC Sampling State Space 📉



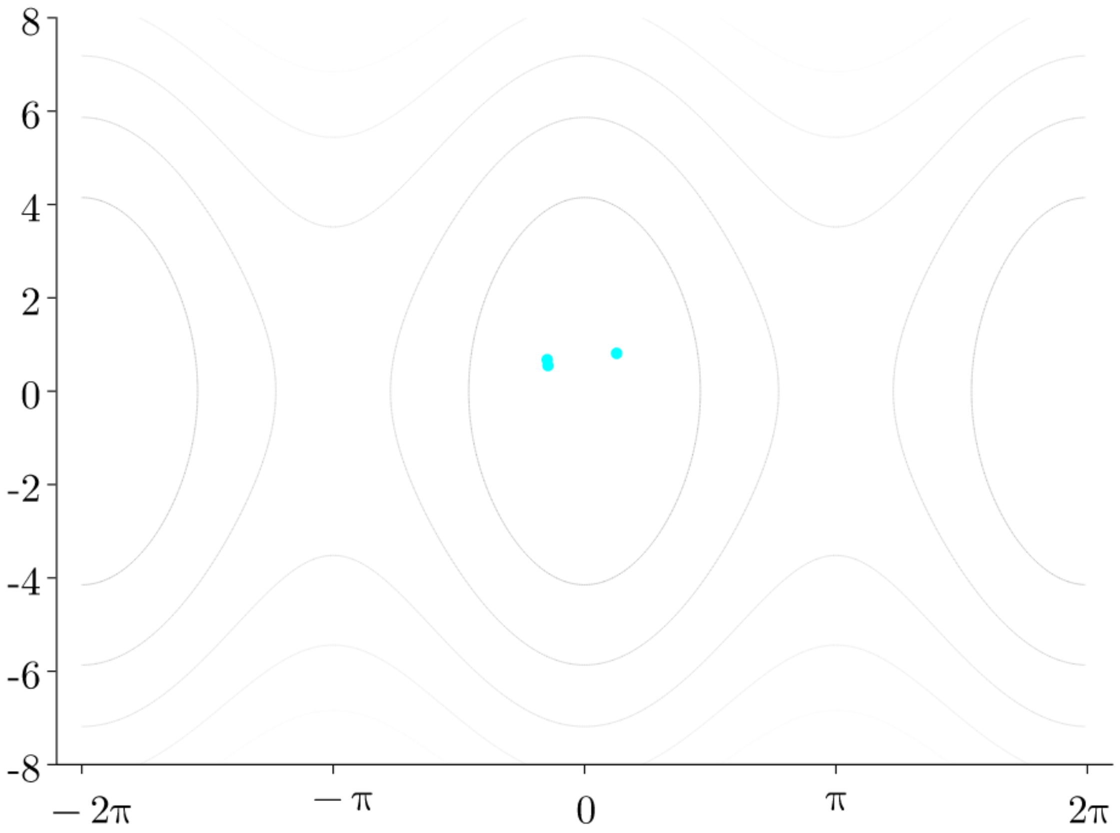
NeuralPbc Sampling State Space 📉



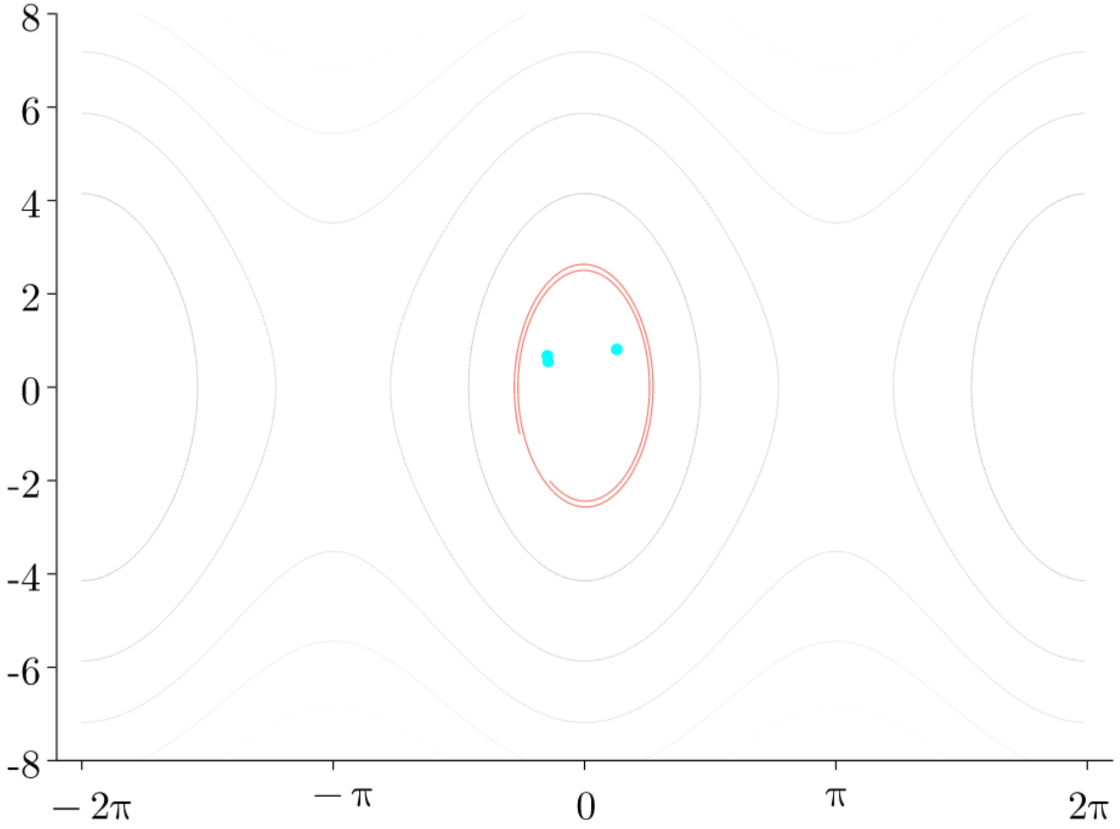
NeuralPbc Sampling State Space 📉



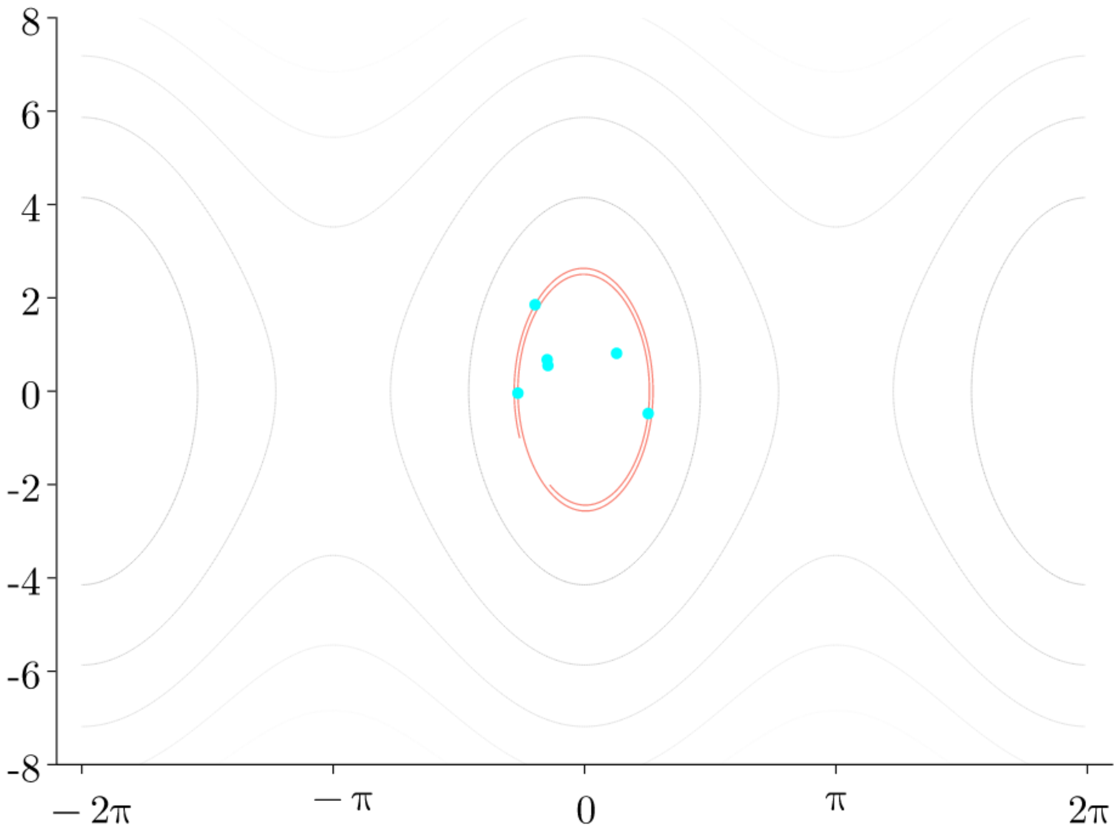
NeuralPbc Sampling State Space 📉



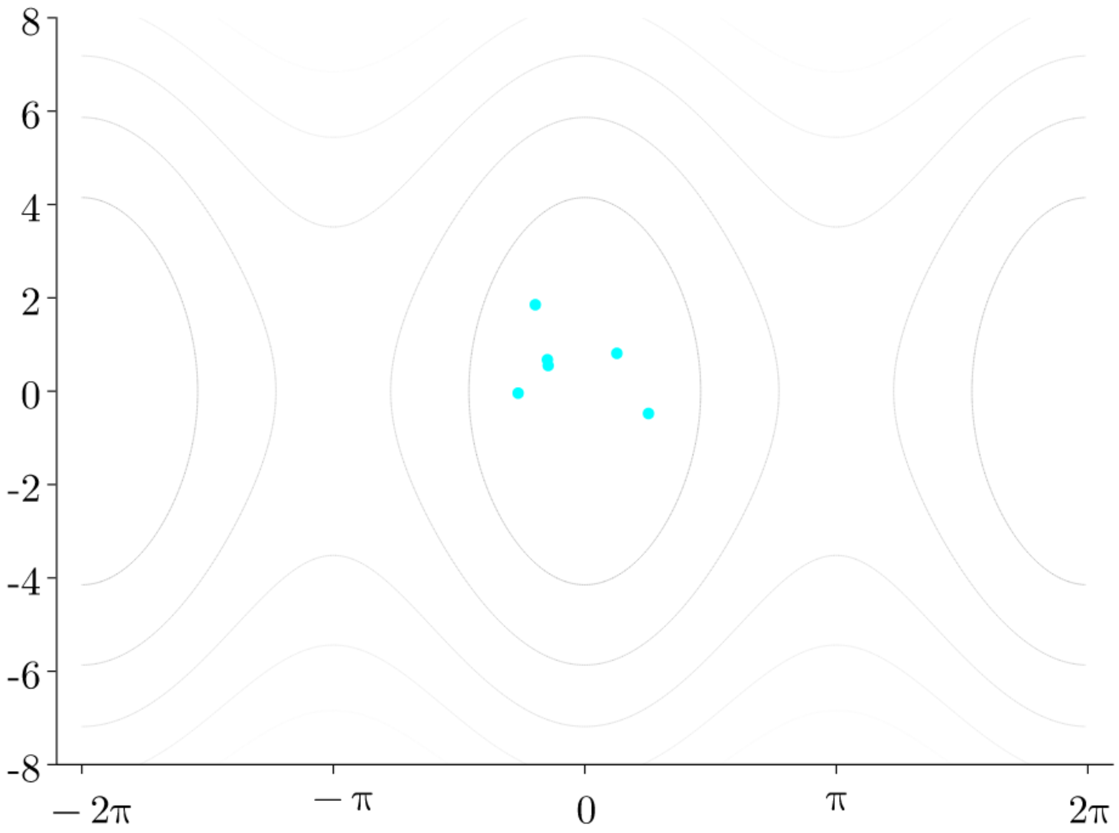
NeuralPbc Sampling State Space 📉



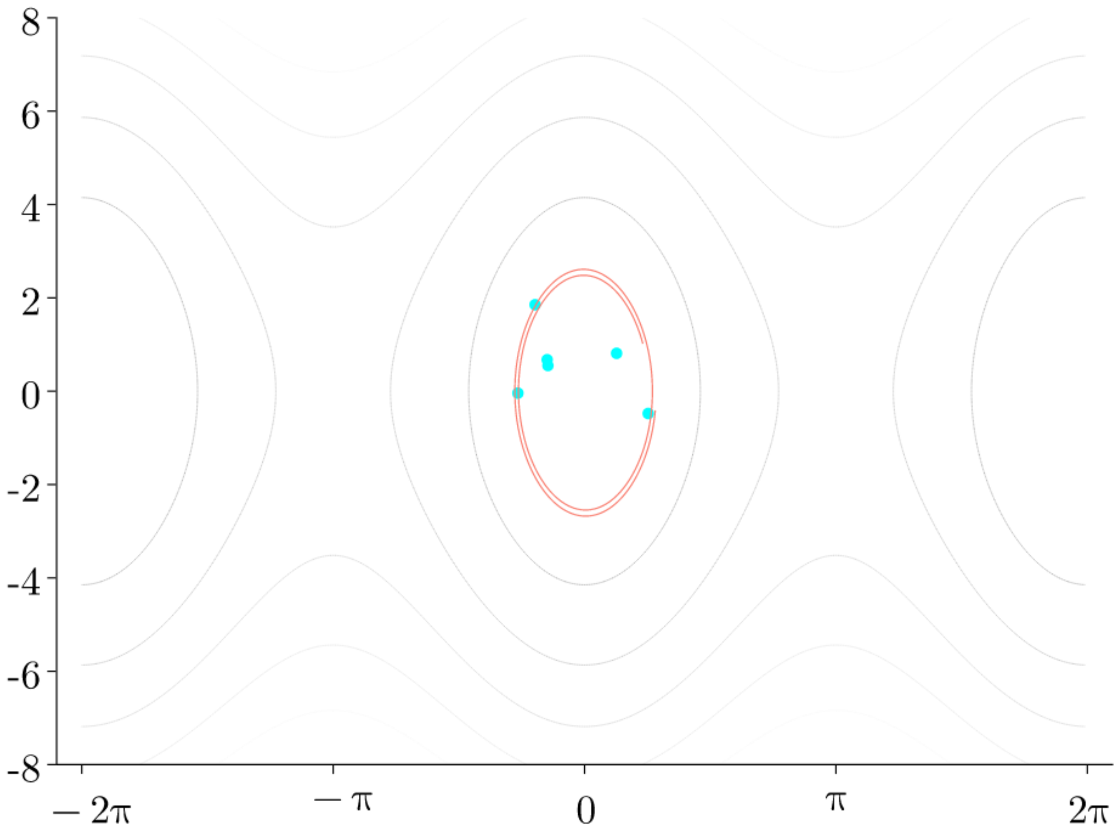
NeuralPbc Sampling State Space 📉



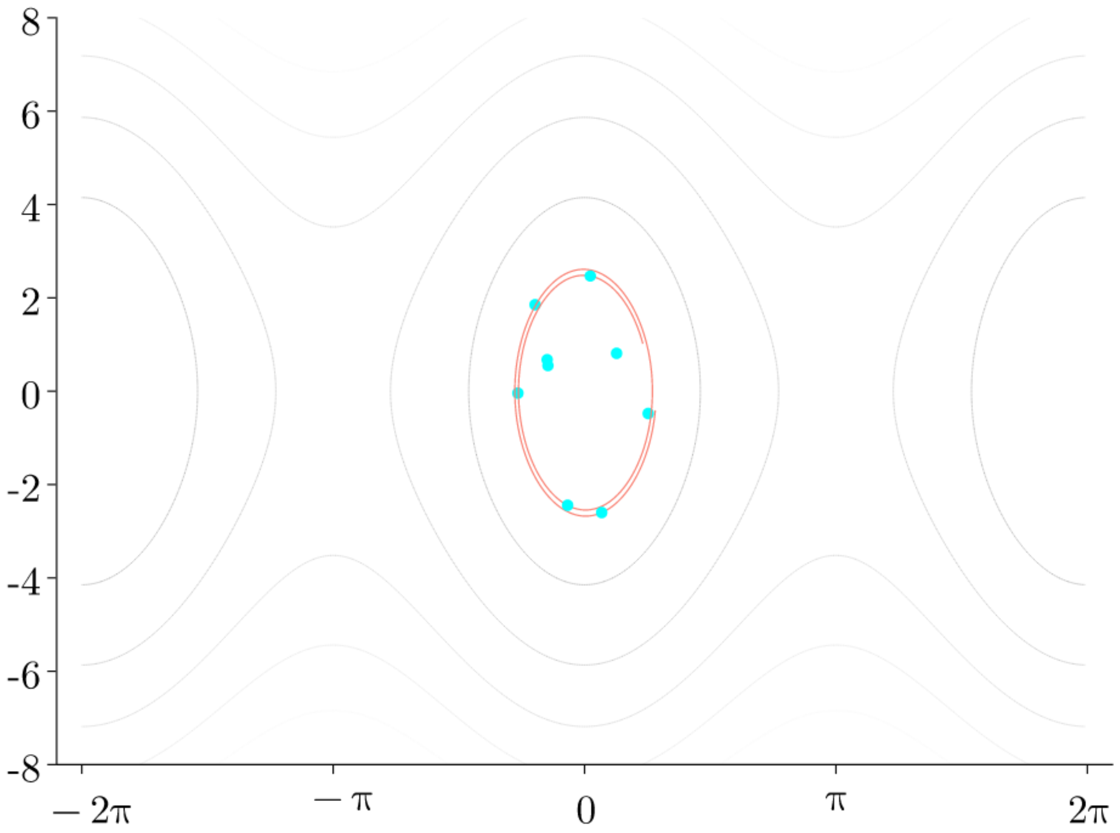
NeuralPbc Sampling State Space 📉



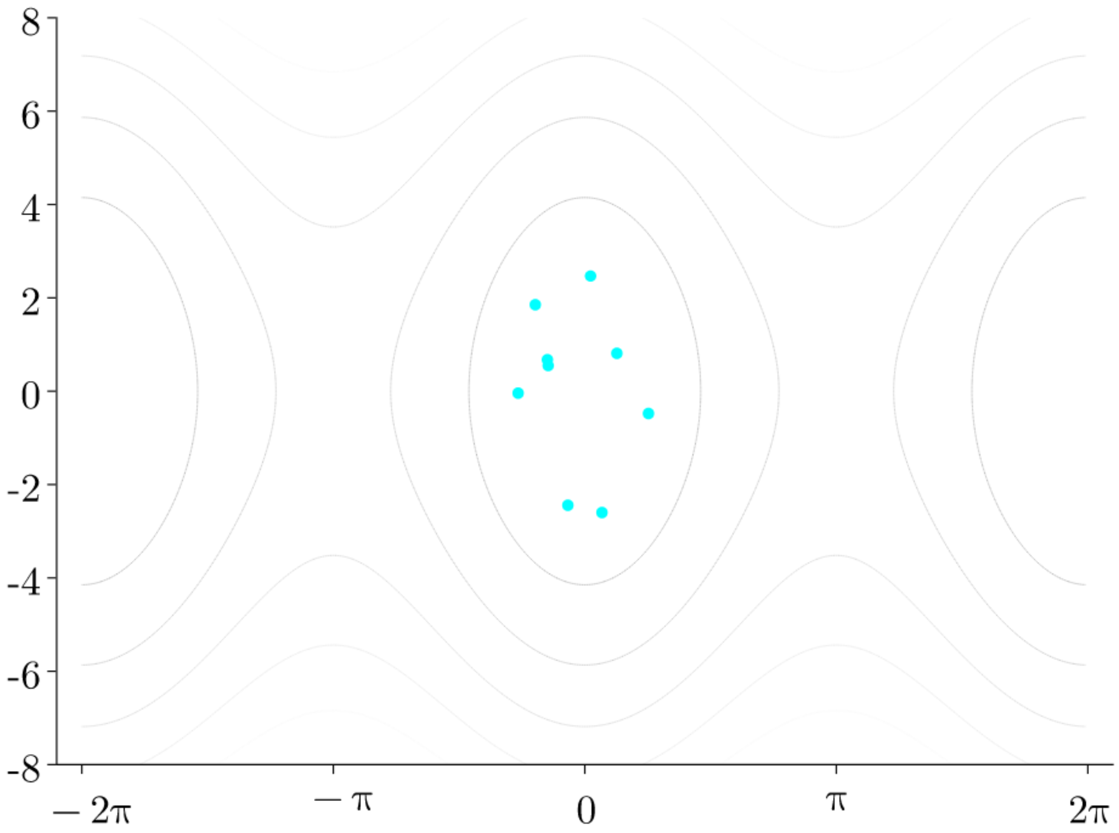
NeuralPbc Sampling State Space 📉



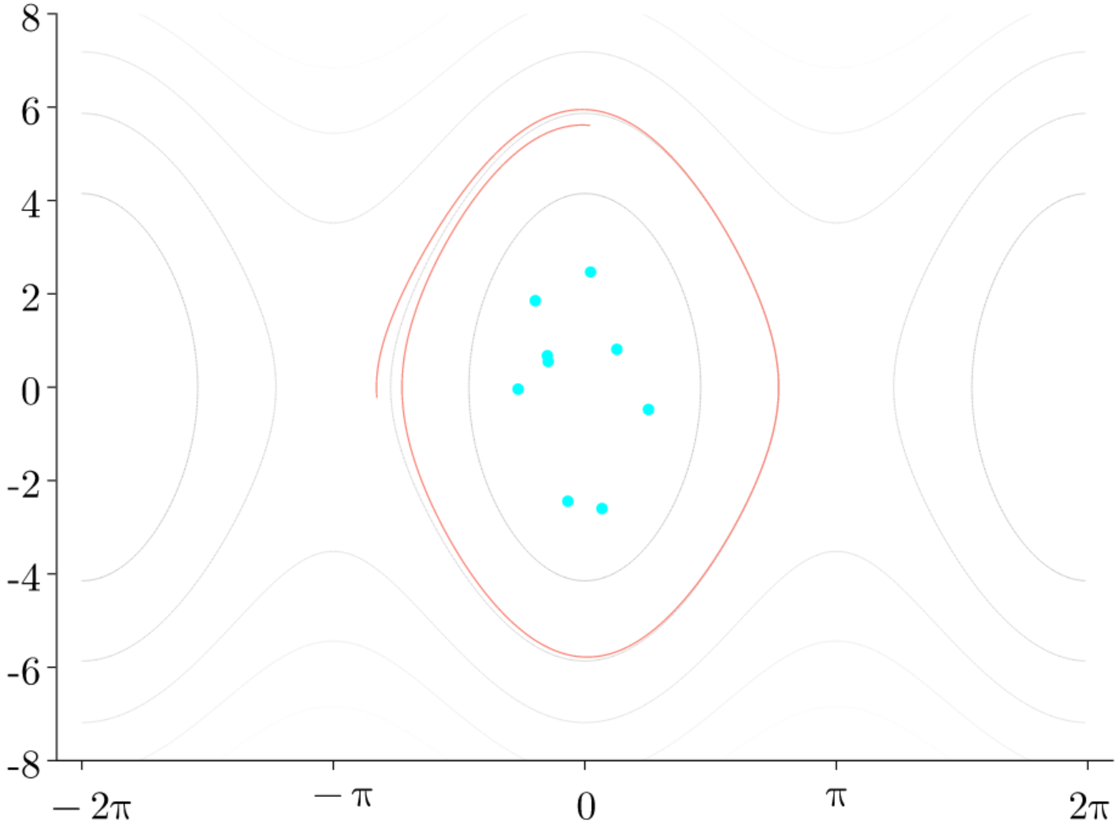
NeuralPbc Sampling State Space 📉



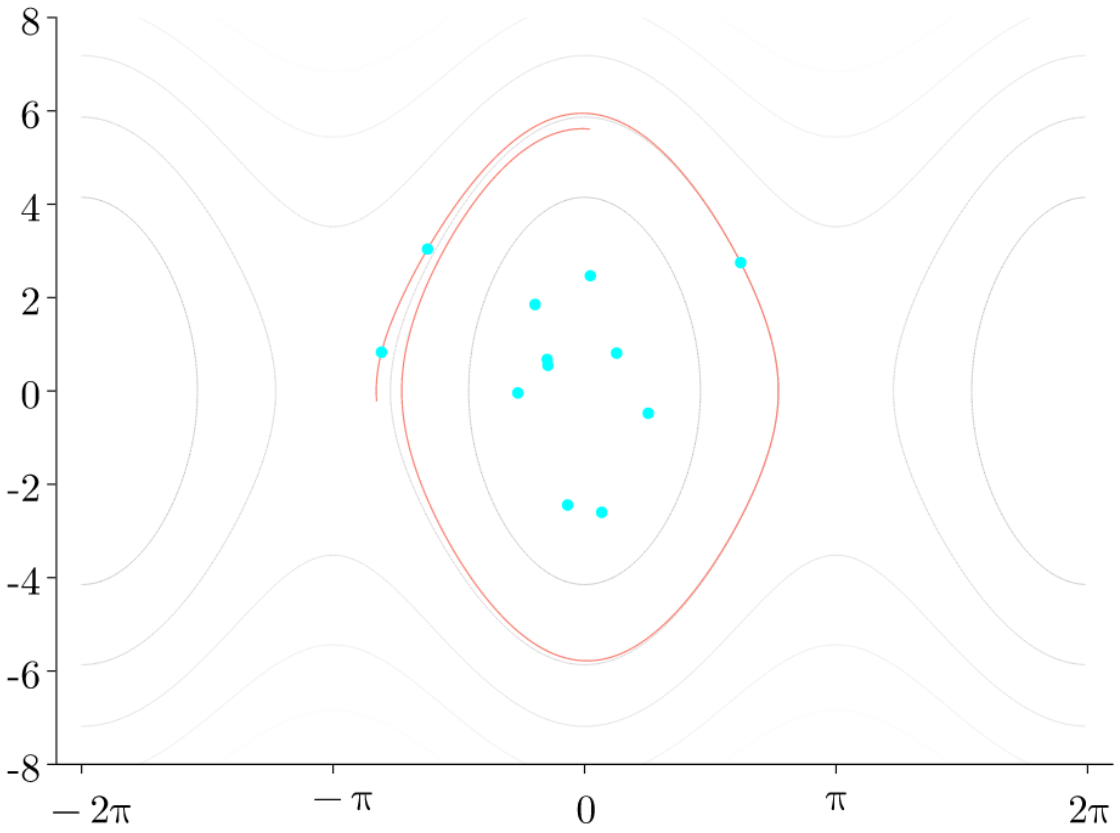
NeuralPbc Sampling State Space 📉



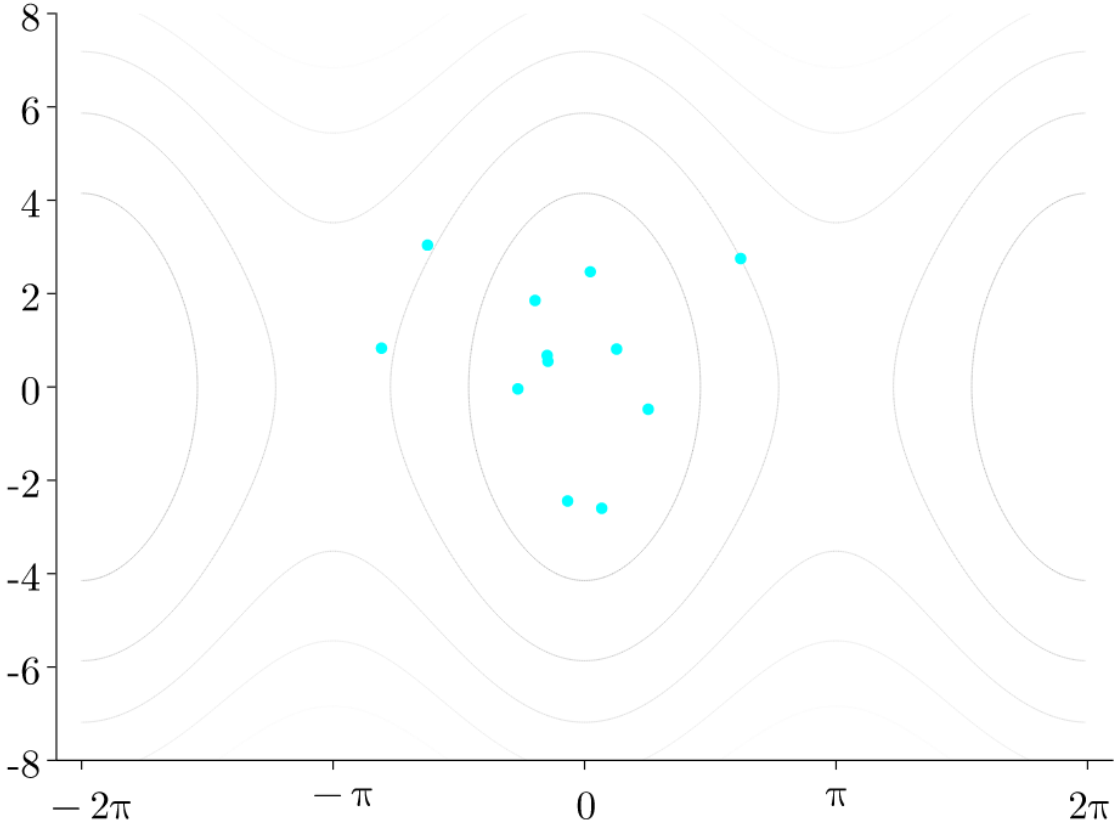
NeuralPbc Sampling State Space 📉



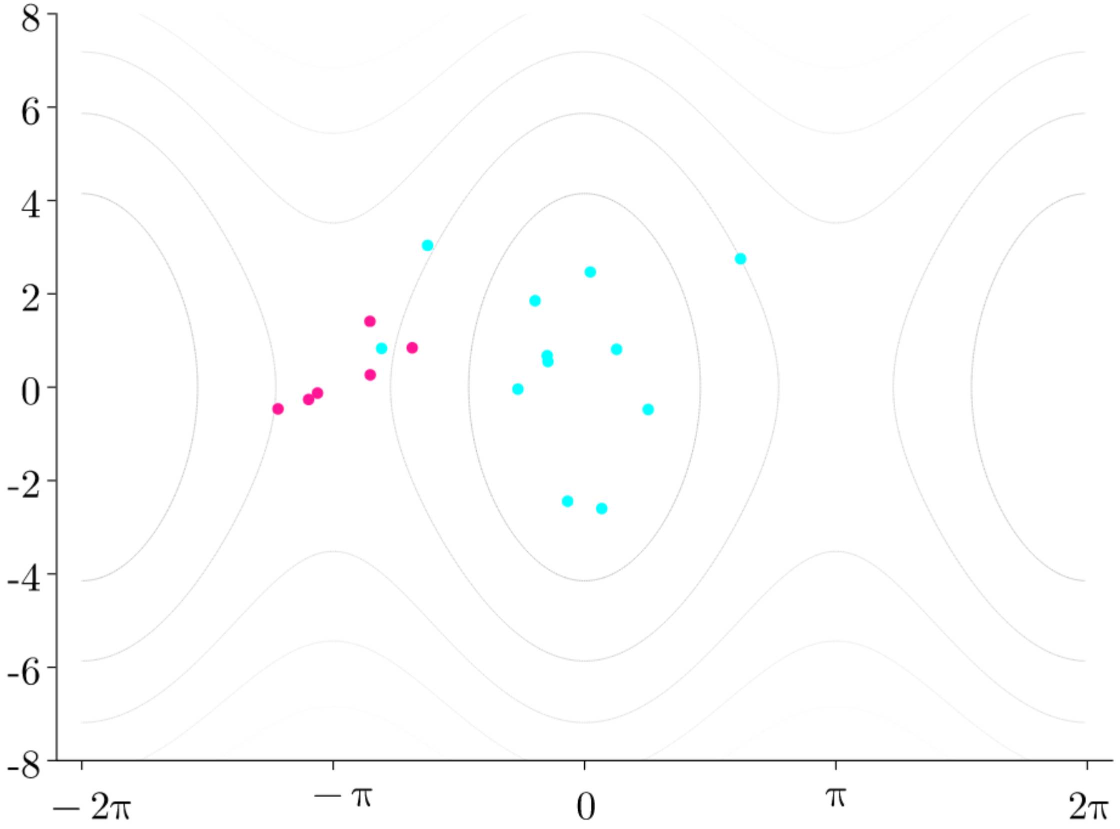
NeuralPbc Sampling State Space 📉



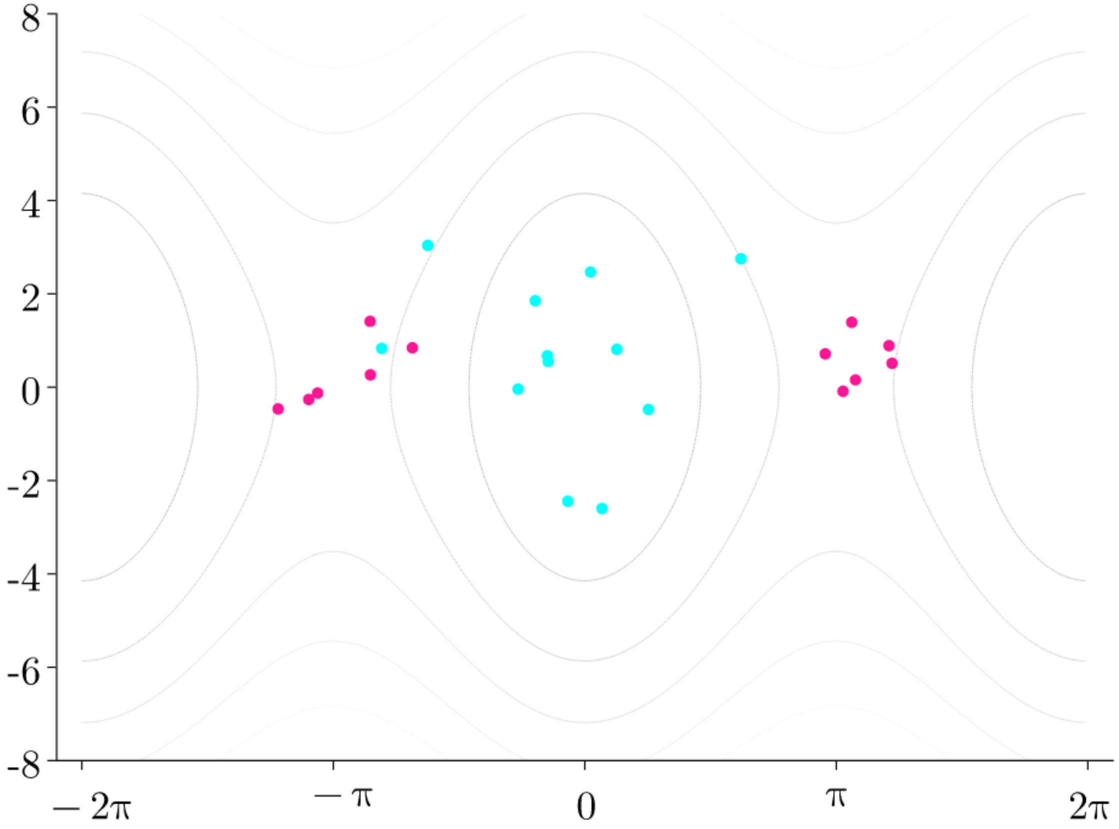
NEURALPBC Sampling State Space 📉



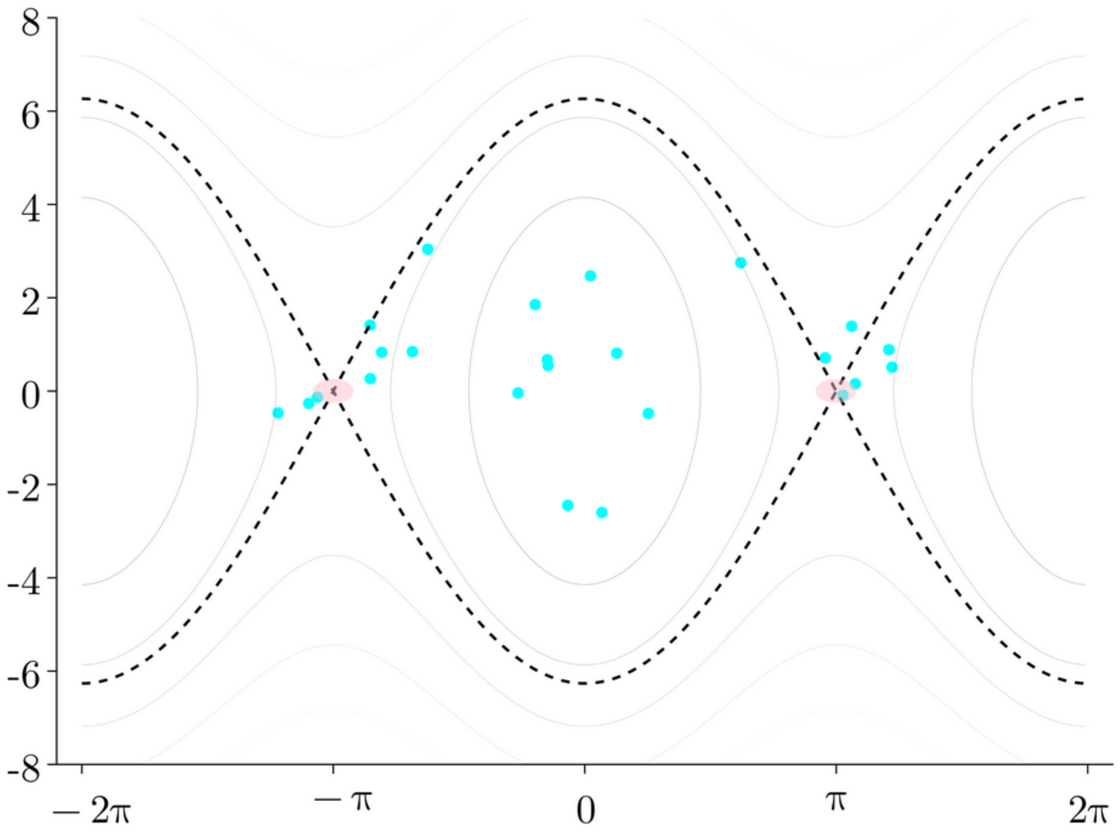
NEURALPBC Sampling State Space 📉



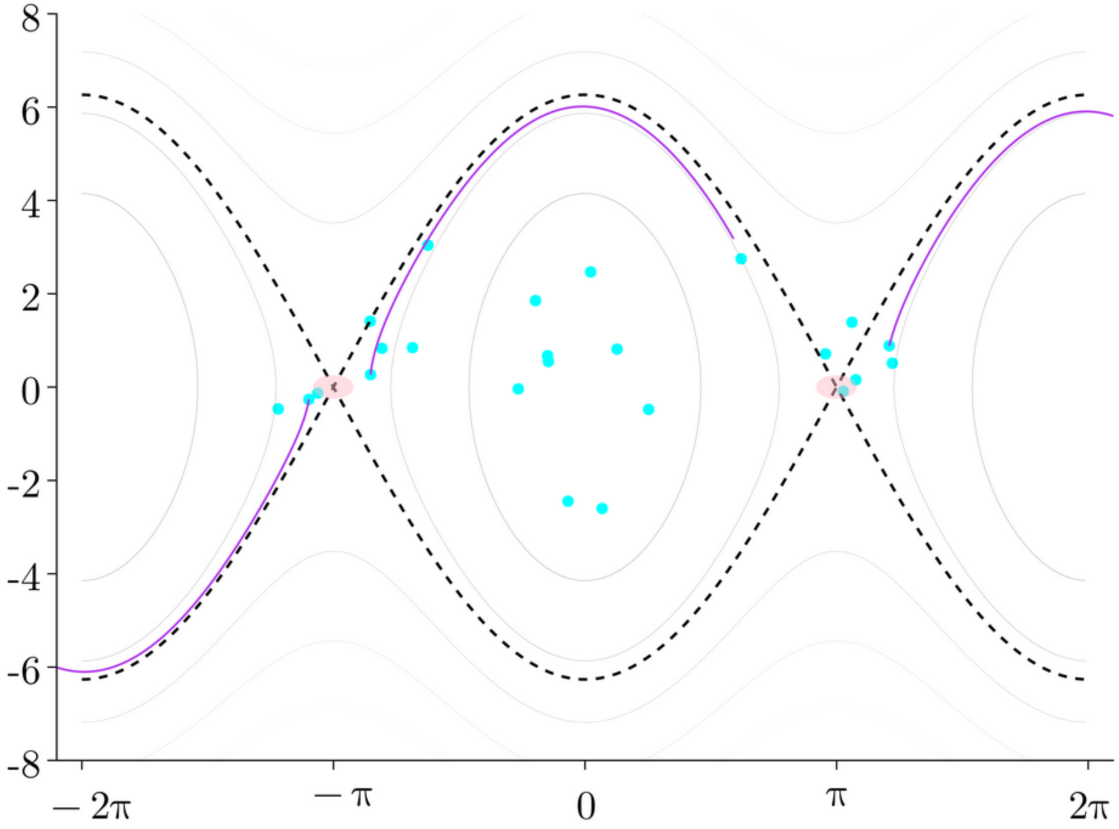
NEURALPBC Sampling State Space 📉



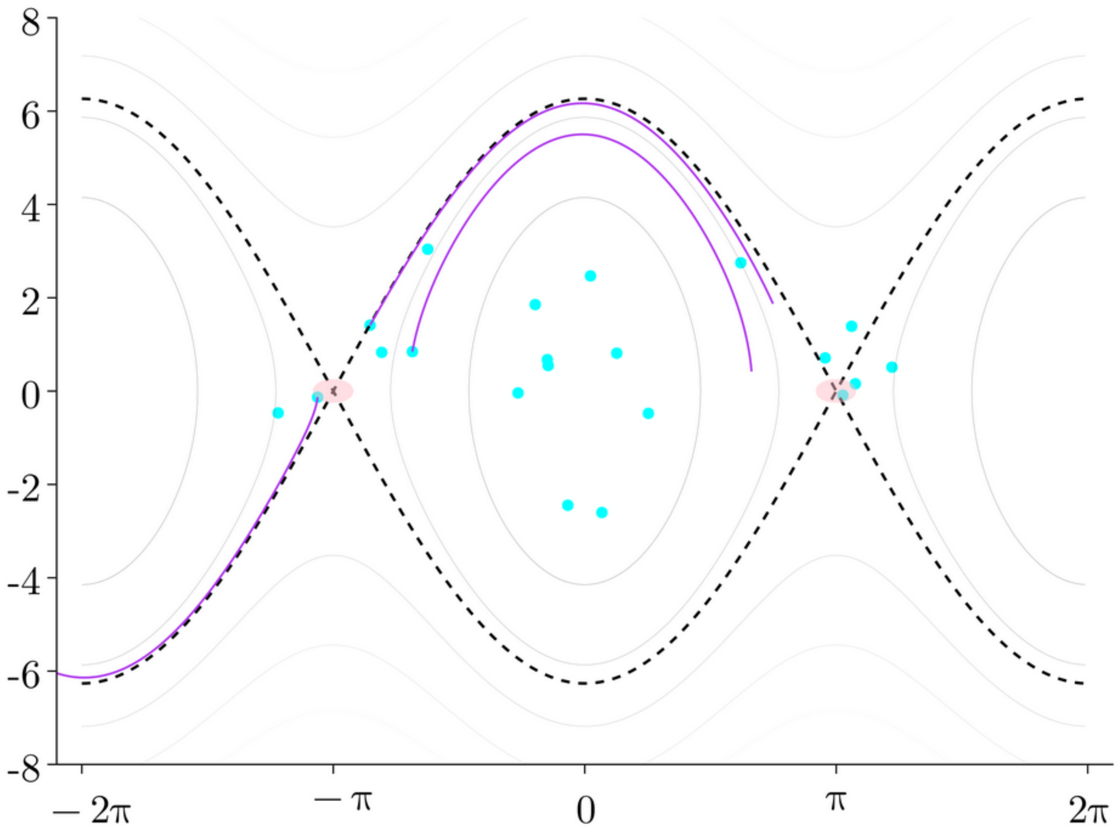
NEURALPBC Sampling State Space 📉



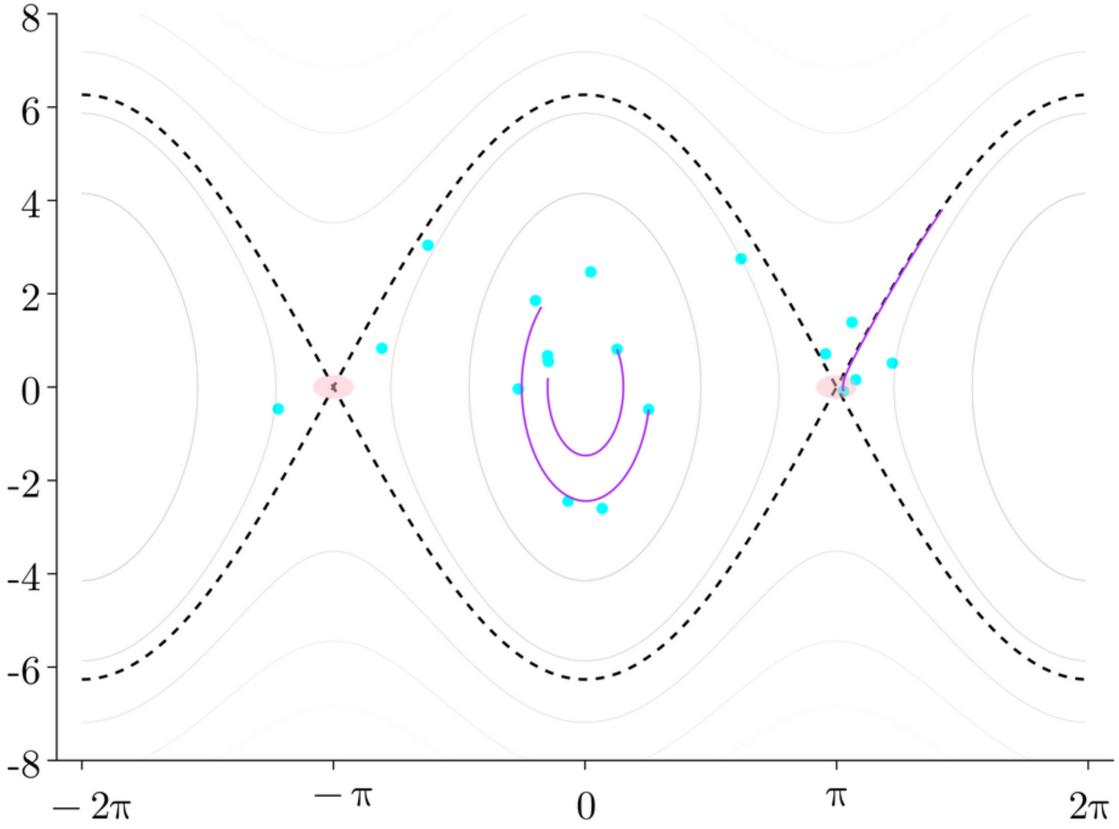
NEURALPBC Sampling State Space 📉



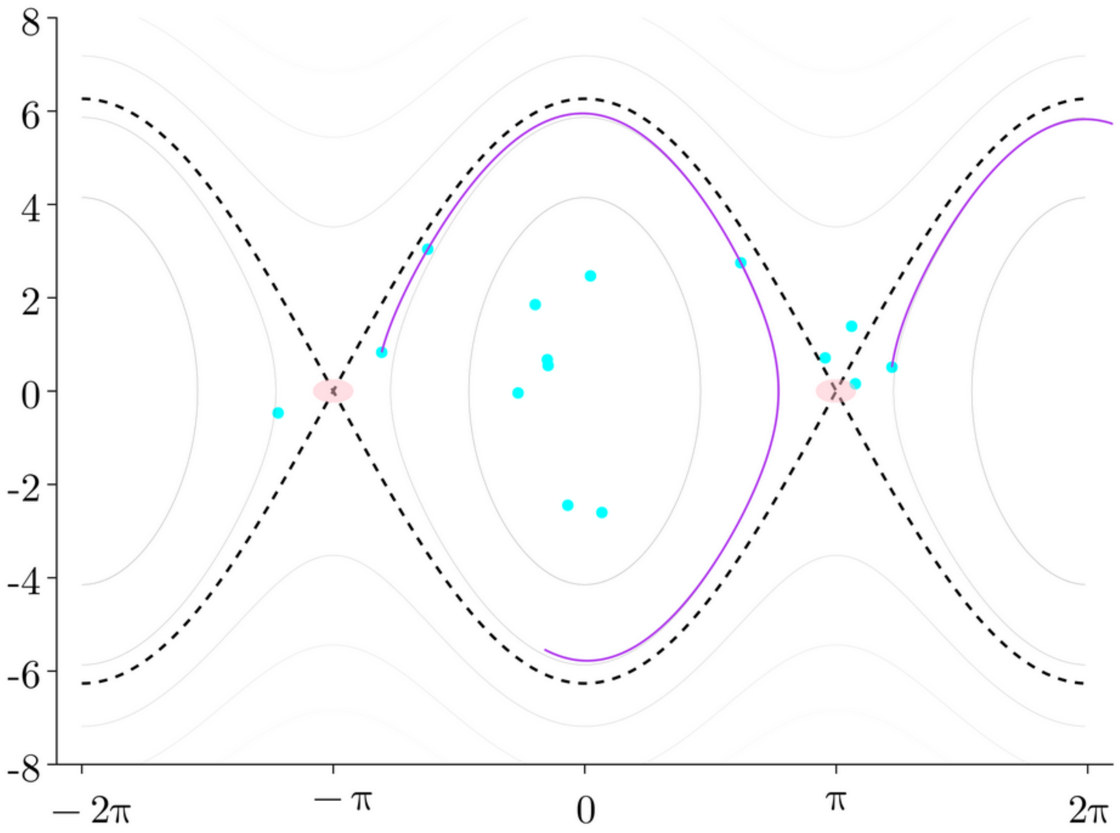
NeuralPbc Sampling State Space 📉



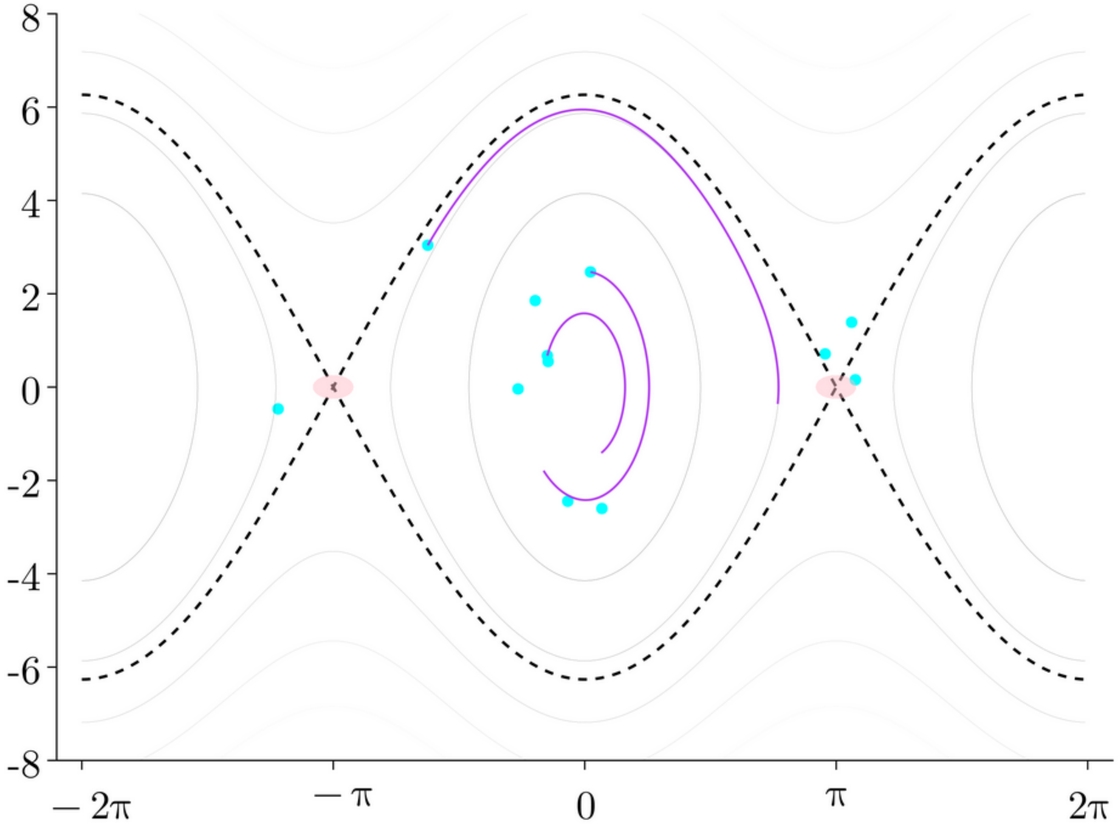
NEURALPBC Sampling State Space 📉



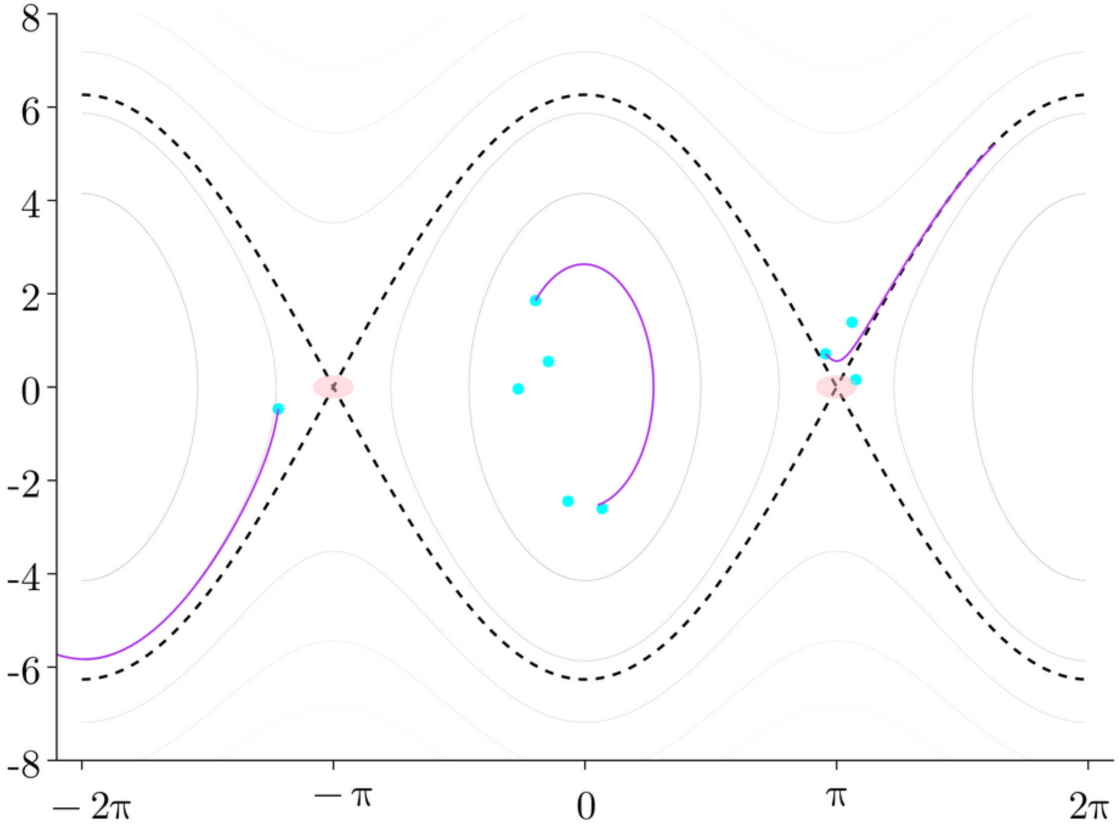
NeuralPbc Sampling State Space 📉



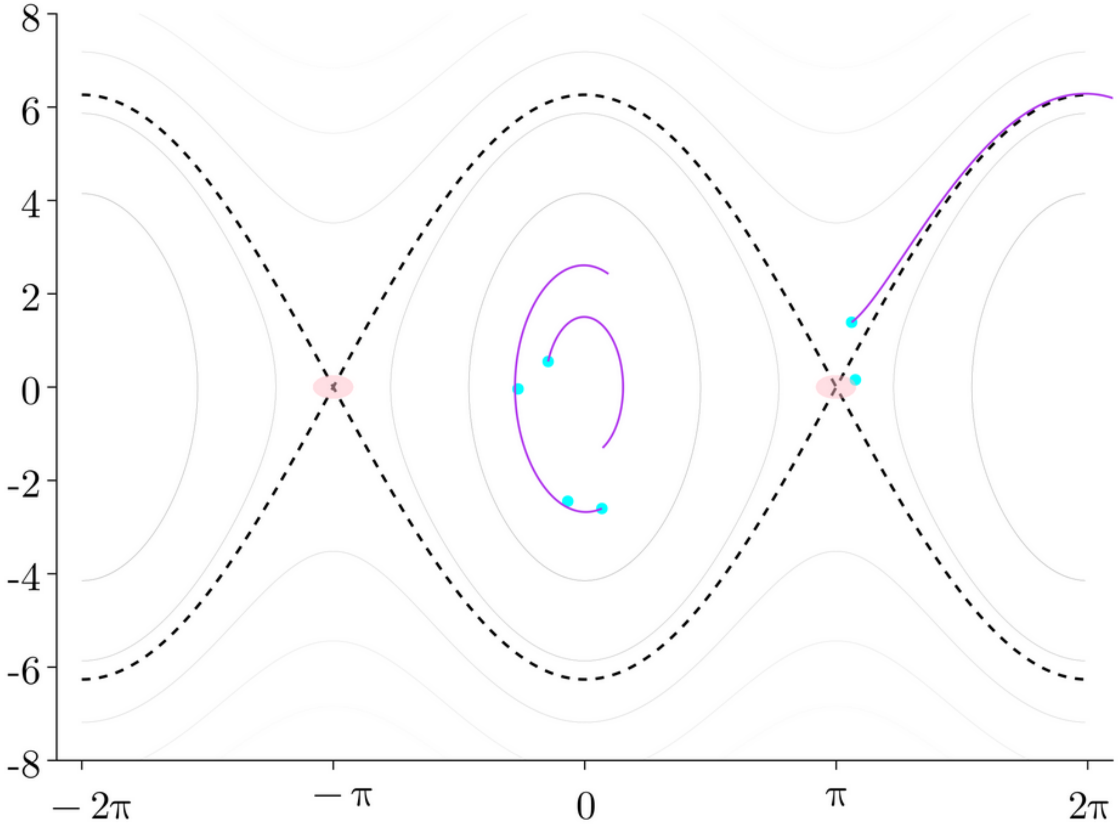
NeuralPbc Sampling State Space 📉



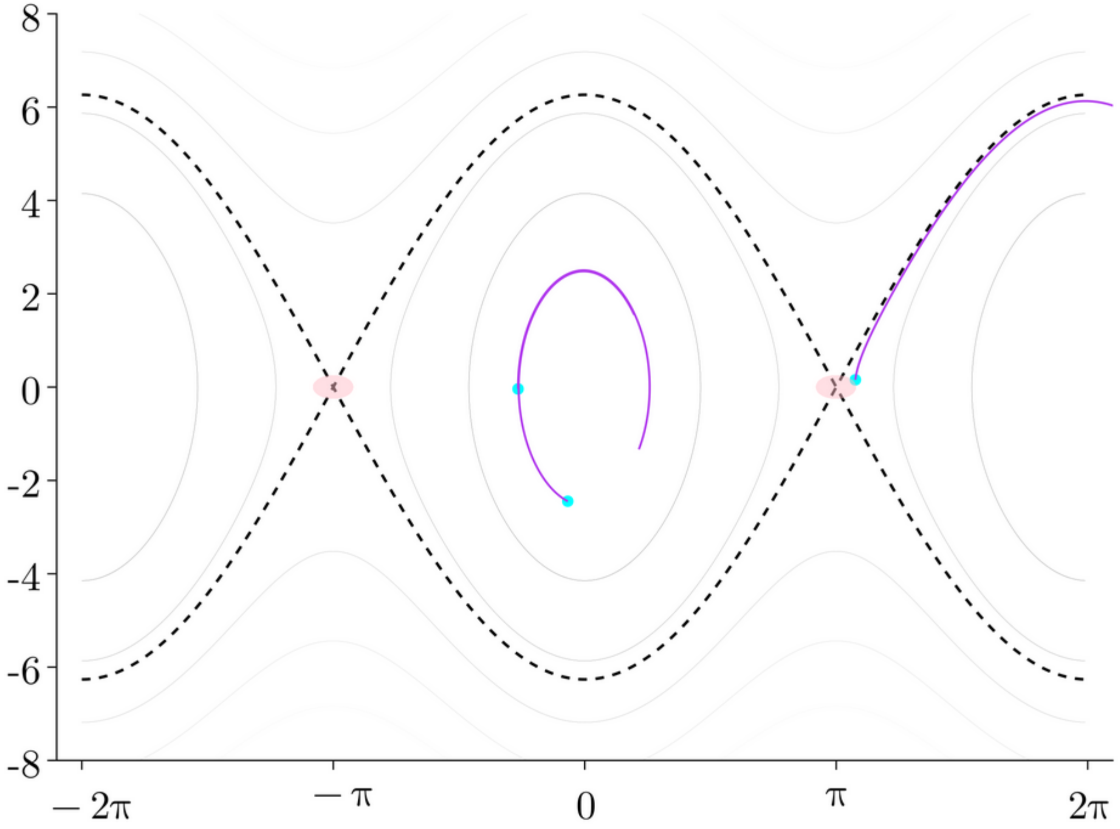
NeuralPbc Sampling State Space 📉



NeuralPbc Sampling State Space 📉



NeuralPbc Sampling State Space 📉



NeuralPbc Cost Function

$$J(\theta, x_0) = \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t$$

$\ell \triangleq \ell_{\text{set}}(\gamma) + \ell_{\perp}(\gamma, u)$ where

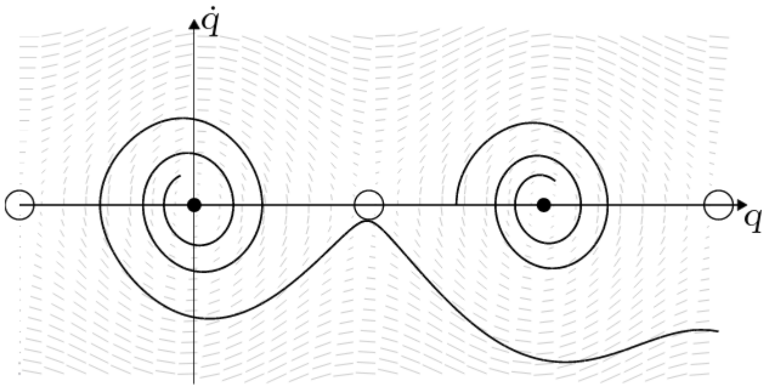
- ϕ is the flow of the equation of motion
- γ is the closed-loop trajectory starting from x_0
- T is the time horizon (hyperparameter)

NeuralPbc Cost Function

$$\ell \triangleq \ell_{\text{set}}(\gamma) + \ell_{\perp}(\gamma, u)$$

Set Distance Loss ℓ_{set}

Penalizes when closed-loop trajectory γ under the current control law is far away from a neighborhood $\mathcal{S}$ of $x^\star$



$$\ell_{\text{set}}(x) = \inf_t \{\|a - b\| : a \in \gamma(t), b \in \mathcal{S}\}$$

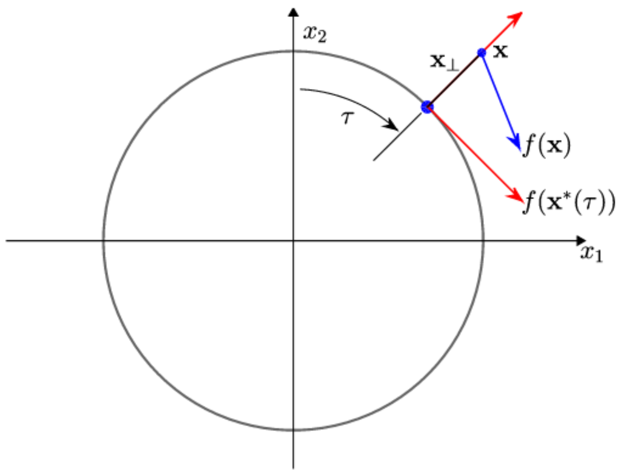
- The set $\mathcal{S}$ may be chosen as
  - A ball around $x^\star$
  - Estimated region of attraction
- No additional loss if any point in γ is in $\mathcal{S}$

NeuralPbc Cost Function

$$\ell \triangleq \ell_{\text{set}}(\gamma) + \ell_{\perp}(\gamma, u)$$

Transversal Distance Loss $\ell_{\perp}$

Measures how close γ is to $\gamma^{\star}$ (expert trajectory) using transverse coordinates $x_{\perp}$



- Coordinate transformation
  - $\tau \in \mathbb{R}$ a surrogate for time
  - $x_{\perp} \in \mathbb{R}^{2n-1}$ quantify how far away the current state is from $\gamma^{\star}$
- By construction $x_{\perp} \to 0 \Leftrightarrow \gamma = \gamma^{\star}$

$$\ell_{\perp} = x_{\perp}^{\top} Q x_{\perp} + u^{\top} R u, \; Q \succcurlyeq 0, \; R \succ 0$$

- No preferred orbit? $Q = 0$

Backprop through ODE Solutions

We need $\partial J / \partial \theta$, which depends ODE solutions

Backprop through ODE Solutions

We need $\partial J/\partial\theta$, which depends ODE solutions

😿 Combining `autodiff` with numerical ODE solvers

Backprop through ODE Solutions

We need $\partial J / \partial \theta$, which depends ODE solutions

😿 Combining `autodiff` with numerical ODE solvers

😿 Adjoint sensitivity method: solve the adjoint problem backward in time

$$\frac{\mathrm{d}\lambda}{\mathrm{d}t} = -\lambda \frac{\partial f}{\partial x}, \qquad \frac{\partial J}{\partial \theta} = \lambda(t_0) \frac{\partial f}{\partial x}$$

Backprop through ODE Solutions

We need $\partial J / \partial \theta$, which depends ODE solutions

😿 Combining `autodiff` with numerical ODE solvers

😿 Adjoint sensitivity method: solve the adjoint problem backward in time

$$\frac{\mathrm{d}\lambda}{\mathrm{d}t} = -\lambda \frac{\partial f}{\partial x}, \quad \frac{\partial J}{\partial \theta} = \lambda(t_0) \frac{\partial f}{\partial x}$$

😺 Adjoint methods + `autodiff` implemented in `DiffEqFlux.jl`

Robust Control Under Uncertainties

Optimal Control under System Parameter Uncertainties

$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \quad & J(\theta, x_0) &&= \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \quad & \dot{x} &&= f\left(x, u^\theta; p\right) \\
& p &&\sim \mathcal{N}\left(\widehat{p}, \sigma_p\right)
\end{aligned}
$$

Robust Control Under Uncertainties

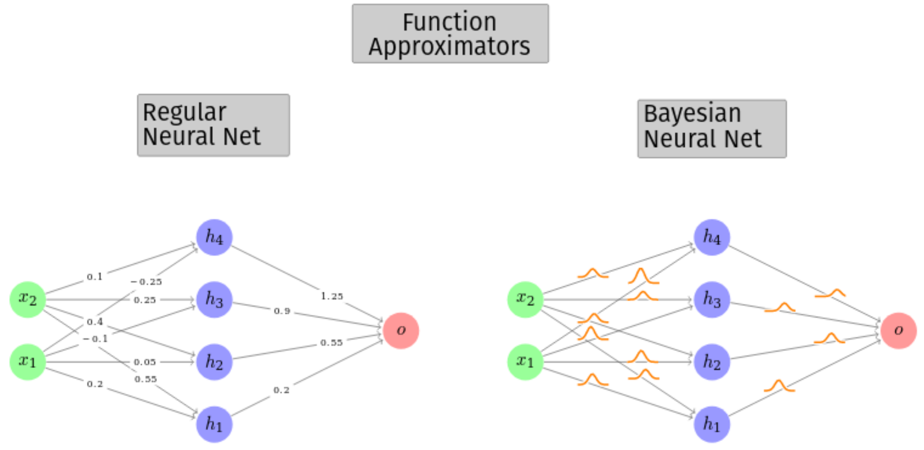
Optimal Control under System Parameter and Measurement Uncertainties

$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \qquad J(\theta, x_0) &= \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \qquad dx &= f\left(x, u^\theta\right) dt + \nabla_x u(x) dW_t \\
p &\sim \mathcal{N}\left(\widehat{p}, \sigma_p\right)
\end{aligned}
$$

Robust Control Under Uncertainties

Optimal Control under System Parameter and Measurement Uncertainties

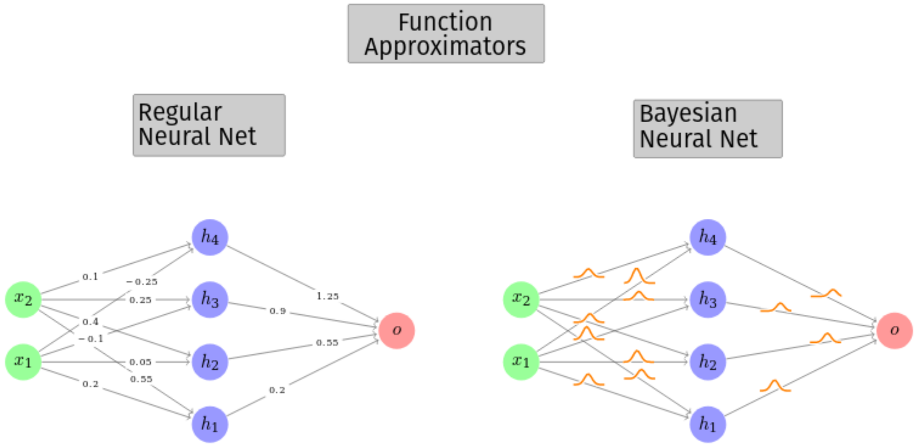
$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \qquad J(\theta, x_0) &= \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \qquad dx &= f\left(x, u^\theta\right) dt + \nabla_x u(x) dW_t \\
p &\sim \mathcal{N}\left(\widehat{p}, \sigma_p\right)
\end{aligned}
$$



Robust Control Under Uncertainties

Optimal Control under System Parameter and Measurement Uncertainties

$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \quad & J(\theta, x_0) &&= \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \quad & dx &&= f(x, u^\theta)dt + \nabla_x u(x) dW_t \\
& p &&\sim \mathcal{N}(\widehat{p}, \sigma_p)
\end{aligned}
$$



- BNNs capture variance in controller
- Akin to learning ensemble of controllers that each minimize J

Bayesian Learning

Bayesian Passivity-based Control

$$
\begin{aligned}
\underset{z}{\text{minimize}} \quad & J(\theta, x_0) &&= \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \quad & dx &&= f\left(x, u^\theta\right)dt + \nabla_x u(x) dW_t \\
& p &&\sim \mathcal{N}\left(\widehat{p}, \sigma_p\right) \\
& u^\theta &&= -G^\dagger \nabla_q H_d^\theta - K_D G^\top \nabla_p H_d^\theta \\
& \theta &&\sim q(\theta; z)
\end{aligned}
$$

Bayesian Learning

Bayesian Passivity-based Control

$$
\begin{aligned}
\underset{z}{\text{minimize}} \quad & J(\theta, x_0) &&= \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \quad & dx &&= f\left(x, u^\theta\right)dt + \nabla_x u(x) dW_t \\
& p &&\sim \mathcal{N}\left(\widehat{p}, \sigma_p\right) \\
& u^\theta &&= -G^\dagger \nabla_q H_d^\theta - K_D G^\top \nabla_p H_d^\theta \\
& \theta &&\sim q(\theta; z)
\end{aligned}
$$

$$
p(\theta \mid \mathcal{D}) = \frac{\overbrace{p(\mathcal{D} \mid \theta)}^{\text{likelihood}} \overbrace{p(\theta)}^{\text{prior}}}{\underbrace{\int_\theta p(\mathcal{D} \mid \theta') p(\theta') d\theta'}_{\text{evidence}}} \approx \underbrace{q(\theta; z)}_{\text{VI}}.
$$

Bayesian Learning

Bayesian Passivity-based Control

$$
\begin{aligned}
\underset{z}{\text{minimize}} \quad & J(\theta, x_0) &&= \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \quad & dx &&= f(x, u^\theta) dt + \nabla_x u(x) dW_t \\
& p &&\sim \mathcal{N}(\widehat{p}, \sigma_p) \\
& u^\theta &&= -G^\dagger \nabla_q H_d^\theta - K_D G^\top \nabla_p H_d^\theta \\
& \theta &&\sim q(\theta; z)
\end{aligned}
$$

$$
p(\theta \mid \mathcal{D}) = \underbrace{\frac{\overbrace{p(\mathcal{D} \mid \theta)}^{\text{likelihood}} \overbrace{p(\theta)}^{\text{prior}}}{\int_\theta p(\mathcal{D} \mid \theta') p(\theta') d\theta'}}_{\text{evidence}} \underbrace{\approx q(\theta; z)}_{\text{VI}}.
$$

KL-divergence and ELBO

$$
\begin{aligned}
D_{\mathrm{KL}} &= \mathbb{E}_{\theta \sim q}\left[\log \frac{q(\theta; z)}{p(\theta \mid \mathcal{D})}\right] \\
&= \log p(\mathcal{D}) - \mathbb{E}_{\theta \sim q}\left[\log \frac{p(\mathcal{D} \mid \theta) p(\theta)}{q(\theta; z)}\right] \\
\mathcal{L}(\mathcal{D}; z) &= \mathbb{E}_{\theta \sim q}[\log p(\mathcal{D} \mid \theta) p(\theta) - \log q(\theta; z)]
\end{aligned}
$$

Bayesian Solution

Computing ELBO

$$\mathcal{L}(\mathcal{D}; z) = \mathbb{E}_{\theta \sim q}[\log p(\mathcal{D} \mid \theta)p(\theta) - \log q(\theta; z)]$$

requires:

Bayesian Solution

Computing ELBO

$$\mathcal{L}(\mathcal{D}; z) = \mathbb{E}_{\theta \sim q}[\ \log p(\mathcal{D} \mid \theta)p(\theta) - \ \log q(\theta; z)]$$

requires:

- *Likelihood*:

$$p(J(\theta, x_0)) = \mathcal{N}(0, s).$$

Bayesian Solution

Computing ELBO

$$\mathcal{L}(\mathcal{D}; z) = \mathbb{E}_{\theta\sim q}[\,\log p(\mathcal{D} \mid \theta)p(\theta) - \,\log q(\theta; z)]$$

requires:

- *Likelihood*:

$$p(J(\theta, x_0)) = \mathcal{N}(0, s).$$

- *Prior*:
  - Uninformed
  - Deterministic

Bayesian Solution

Computing ELBO

$$\mathcal{L}(\mathcal{D}; z) = \mathbb{E}_{\theta\sim q}[\log p(\mathcal{D} \mid \theta)p(\theta) - \log q(\theta; z)]$$

requires:

- *Likelihood*:

$$p(J(\theta, x_0)) = \mathcal{N}(0, s).$$

- *Prior*:
  - Uninformed
  - Deterministic

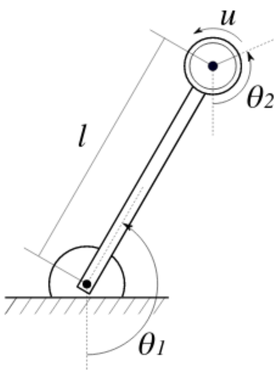
Prediction through maximum aposteriori

$$u(x) = u\left(x, \operatorname*{argmax}_{\theta} p(\theta; z)\right).$$

Prediction through marginalization

$$u(x) = \frac{1}{N}\sum_{\theta\sim q} u(x, \theta).$$

Case Study: Inertia Wheel Pendulum



The control input u is torque applied to the wheel. The equations of motion are

$$\begin{bmatrix} I_1 & 0 \\ 0 & I_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -mgl \sin q_1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} u.$$

NeuralPBC training setup

	Deterministic	**Bayesian**
H_d neural net size	(6, 12, 3, 1)	(6, 5, 3, 1)
Learned parameters	133	128
Optimizer	ADAM	DecayedAdaGrad
Initial learning rate	0.001	0.01
Replay buffer size	400	50

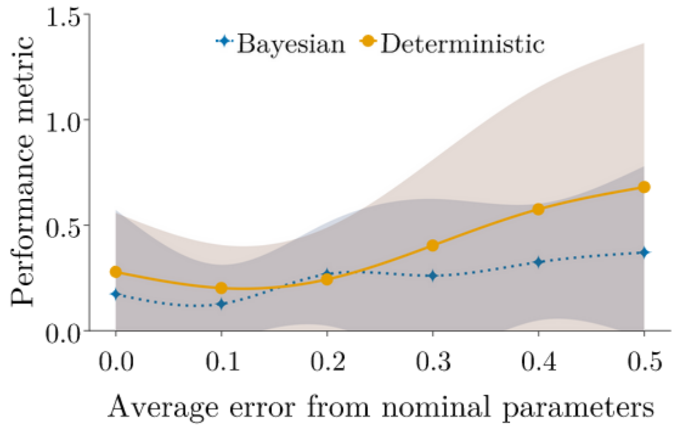
Deterministic vs Bayesian in Simulation

- Tested on swingup-task of the inertia wheel pendulum
- Parameter uncertainities and measurement noise are modelled as

$$dx = \left(\begin{bmatrix} \nabla_p H \\ -\nabla_q H \end{bmatrix} + \begin{bmatrix} 0 \\ G(q) \end{bmatrix} u^\theta(x) \right) dt + \nabla_x u(x) dW_t.$$

- Measurement noise given by Wiener process with state uncertainties of 0.05 rad and 0.0005 rad/s
- Performance metric given by

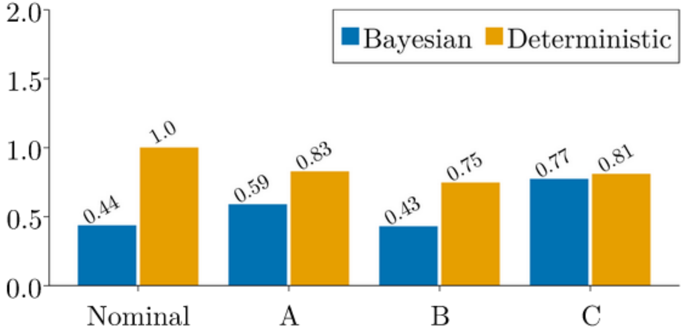
$$J_T = \frac{1}{2} \int_0^T (x^\top Q x + u^\top R u) dt$$

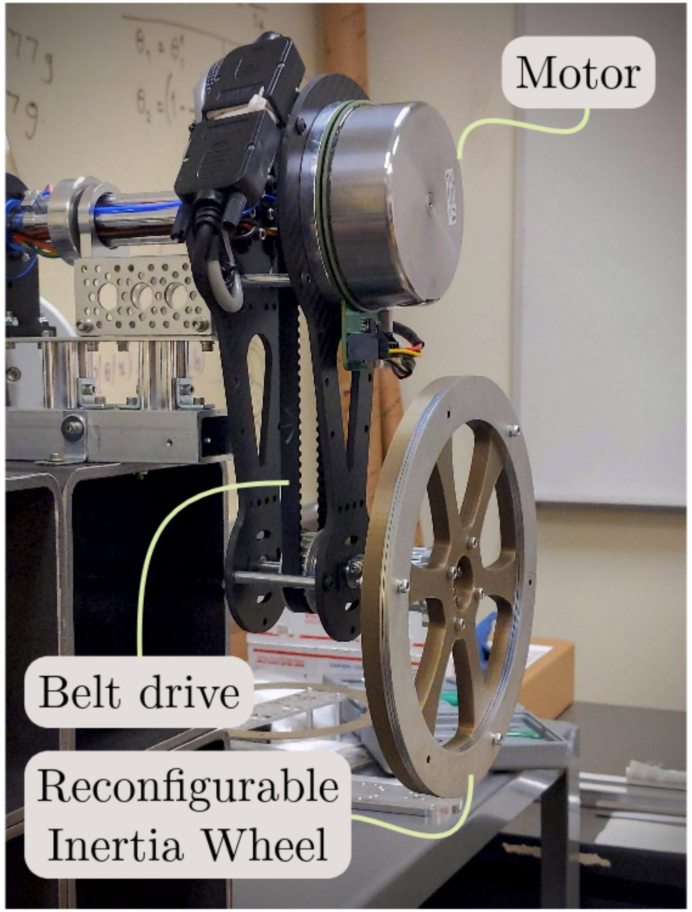


Deterministic vs Bayesian in Experiment

System parameters used in real-world experiments. The errors in the last column are $\|p_s - p_s^{\mathrm{nom}}\| / \|p_s^{\mathrm{nom}}\|$.

Parameter set p_s	I_1	I_2	mgl
Nominal	0.0455	0.00425	1.795
A	0.0417	0.00330	1.577
B	0.0378	0.00235	1.358
C	0.0340	0.00141	1.140





Closing Thoughts and Future Directions

PBC + machine learning techniques ✨

- We uncovered the engineering foundations for combining them
- Extensive experimental results in simulation and on hardware

Future directions—applications in:

- Dynamical models with uncertainity
- Hybrid dynamical systems (walking machines)