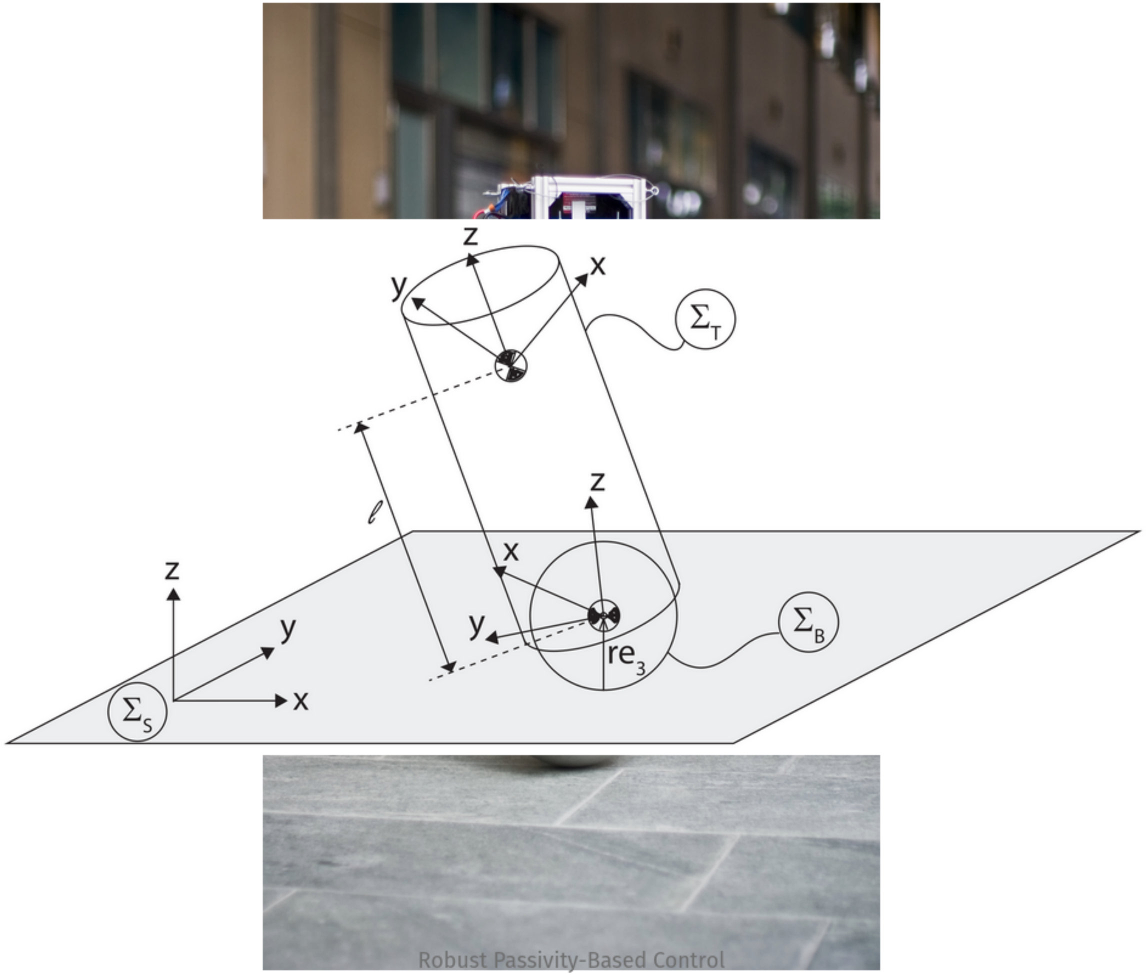
High-dimensional Problem: Ballbot



High-dimensional Problem: Ballbot

$$
\begin{aligned}
\dot{H}_d = (k_1 y_1 + k_2 y_2) \Bigg[& \left(k_e + k_1 k_k + k_2 k_k \frac{(\alpha + \beta \cos(x))^2}{\alpha + \gamma + 2\beta \cos(x)} \right) u \\
& + k_2 k_k \Bigg(- (\alpha + \beta c_x) \Bigg(\frac{\beta s_x}{\alpha + \gamma + 2\beta c_x} \dot{x}^2 \\
& + \frac{\mu s_x}{\alpha + \gamma + 2\beta c_x} \Bigg) + \beta s_x \dot{x}^2 \Bigg) \\
& + k_I \left(k_1 \theta - k_2 (\alpha x + \beta s_x) \right) \Bigg].
\end{aligned}
$$

High-dimensional Problem: Ballbot

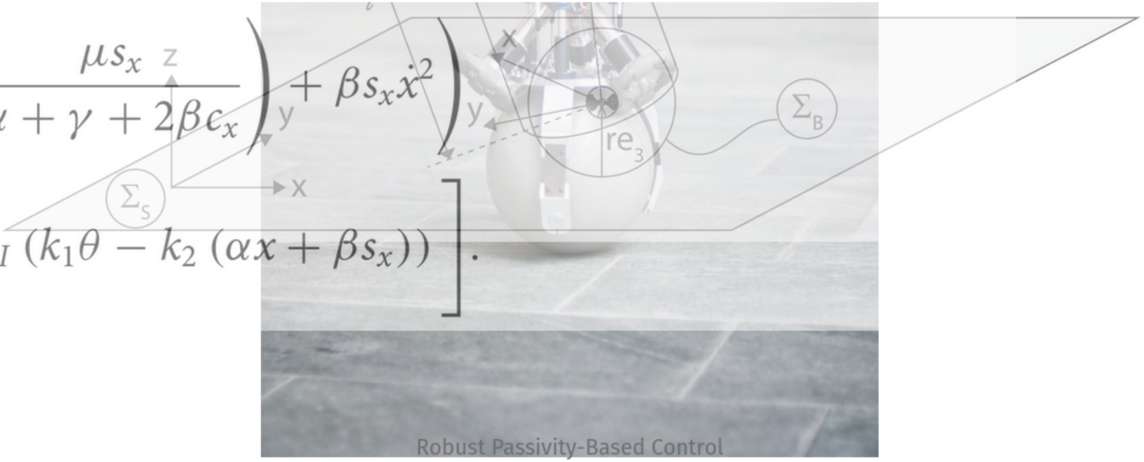
$$u = -\frac{1}{k}\Bigg[k_2 k_k\Bigg(-(\alpha+\beta c_x)\Bigg(\frac{\beta s_x}{\alpha+\gamma+2\beta c_x}\dot{x}^2 + \frac{\mu s_x}{\alpha+\gamma+2\beta c_x}\Bigg) + \beta s_x \dot{x}^2\Bigg) + k_I\left(k_1\theta - k_2\left(\alpha x + \beta s_x\right)\right) + k_p\left(k_1 y_1 + k_2 y_2\right)\Bigg]$$

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

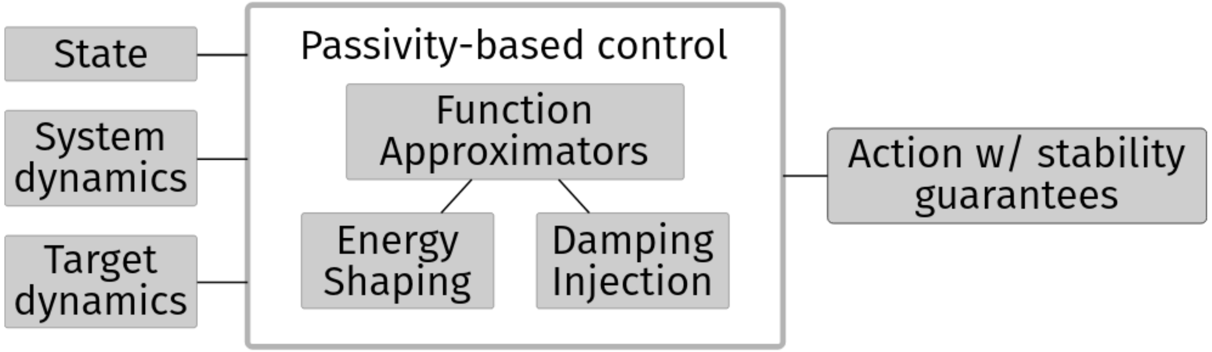
$$+ k_2 k_k \Big(-$$

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

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

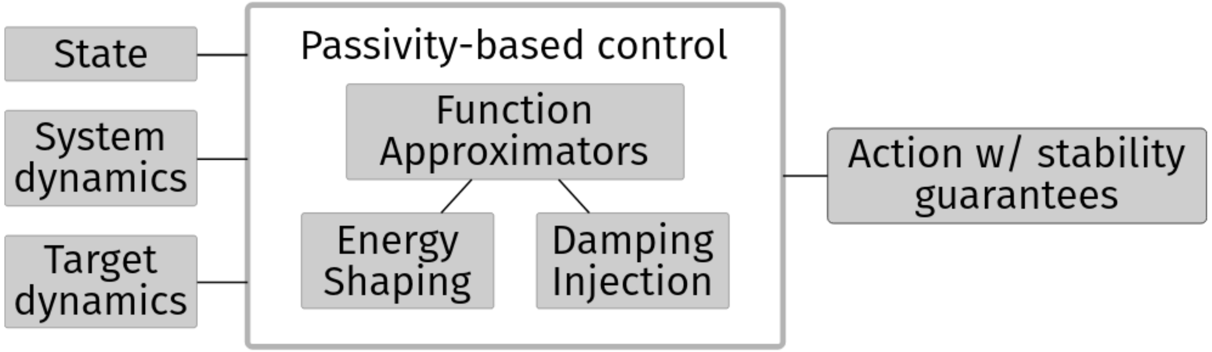


Our Methods



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

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

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

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

- Sampling the state space efficiently

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
- Injecting control task into loss function design

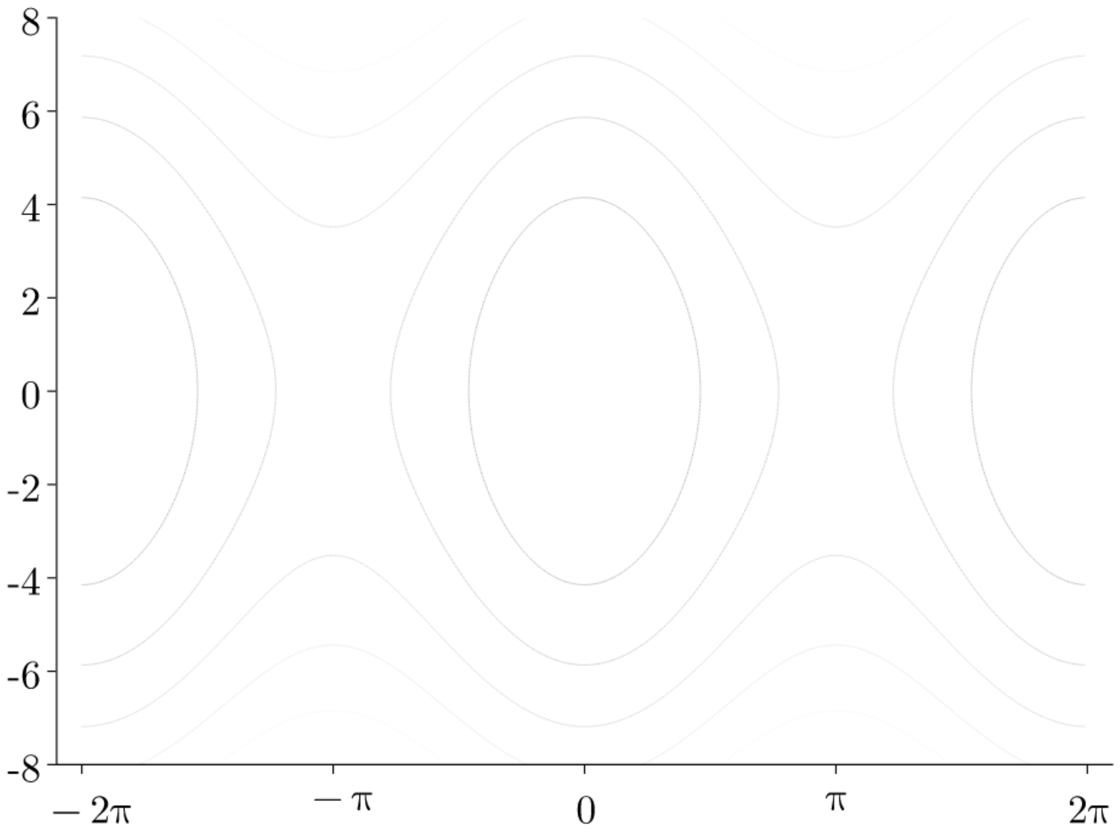
NeuralPbc Problem Statement

Data-Driven Passivity-based control

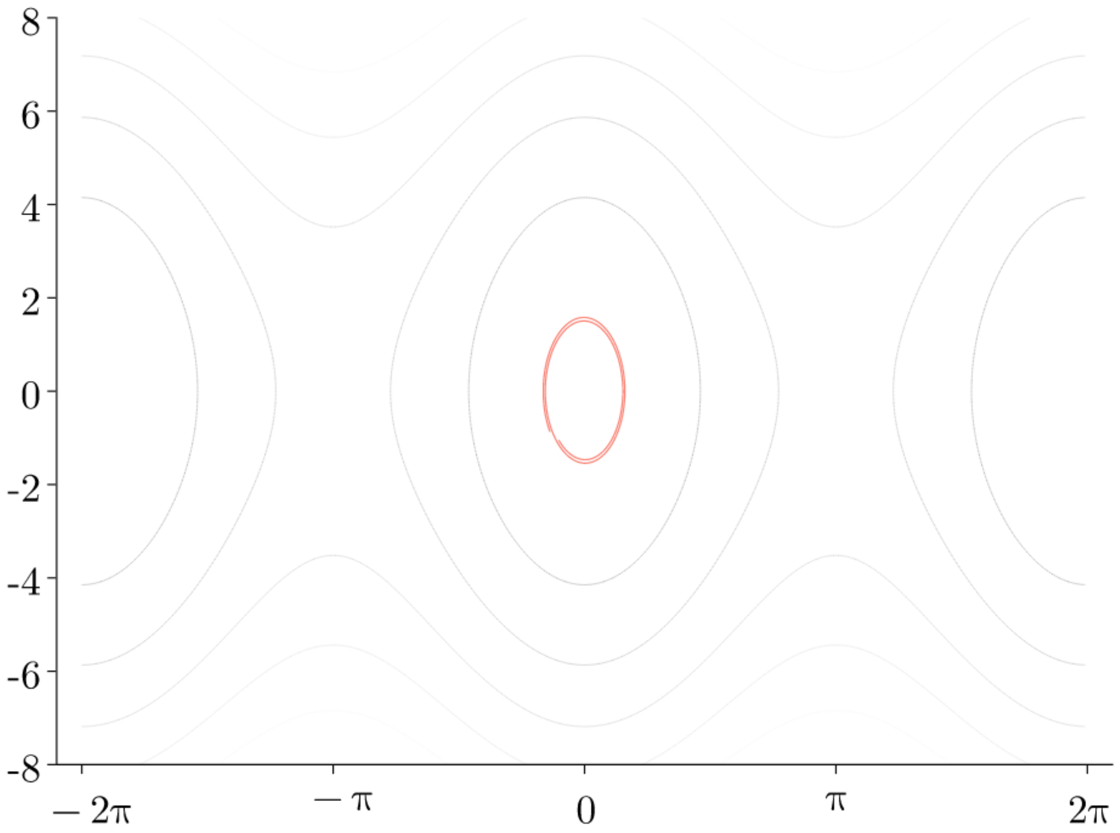
$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \quad & J(\theta, x_0) && = \int_0^T \ell\left(\phi, u^\theta, \theta\right) \mathrm{d}t \\
\text{subject to} \quad & \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} && = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\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

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

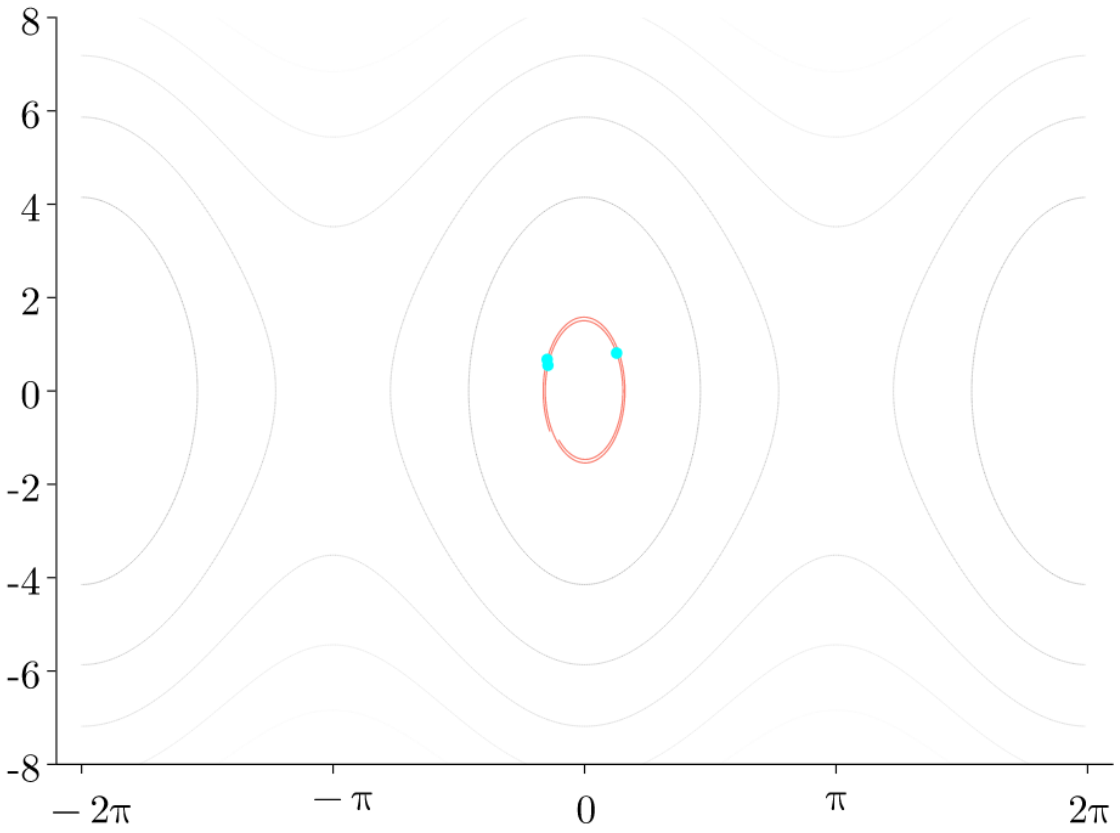
NEURALPBC Sampling State Space 📉



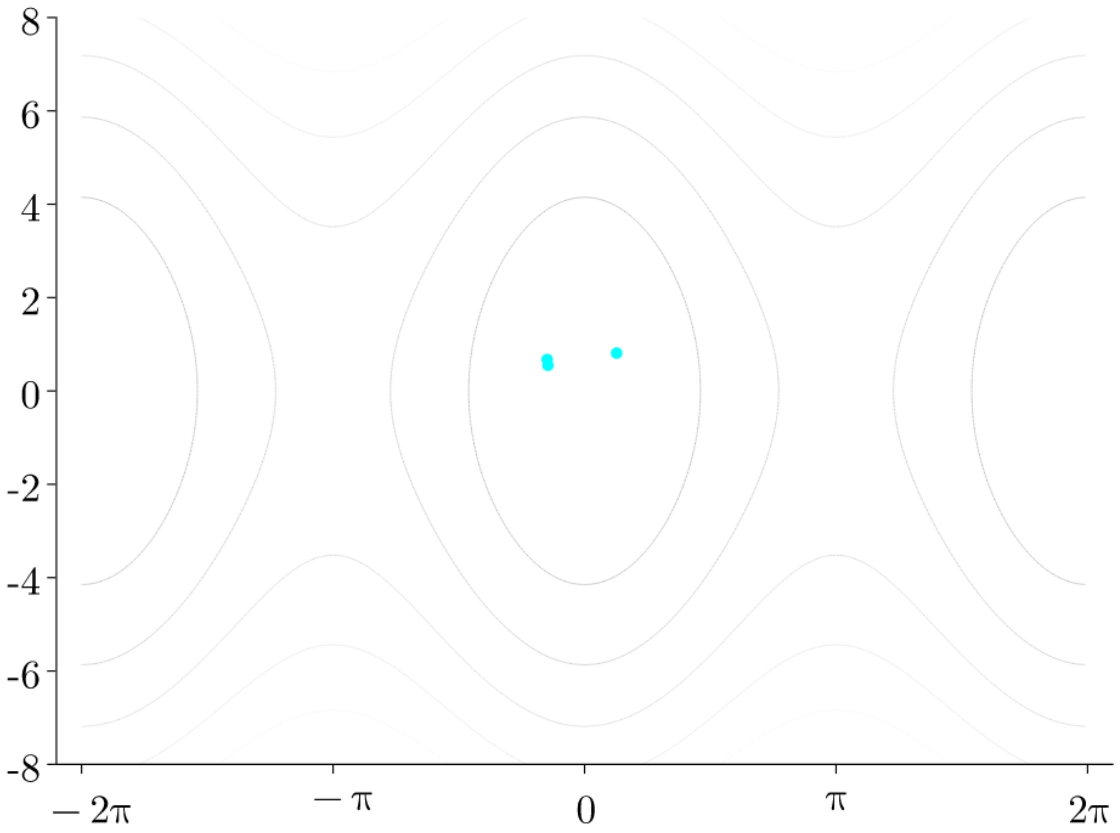
NeuralPbc Sampling State Space 📉



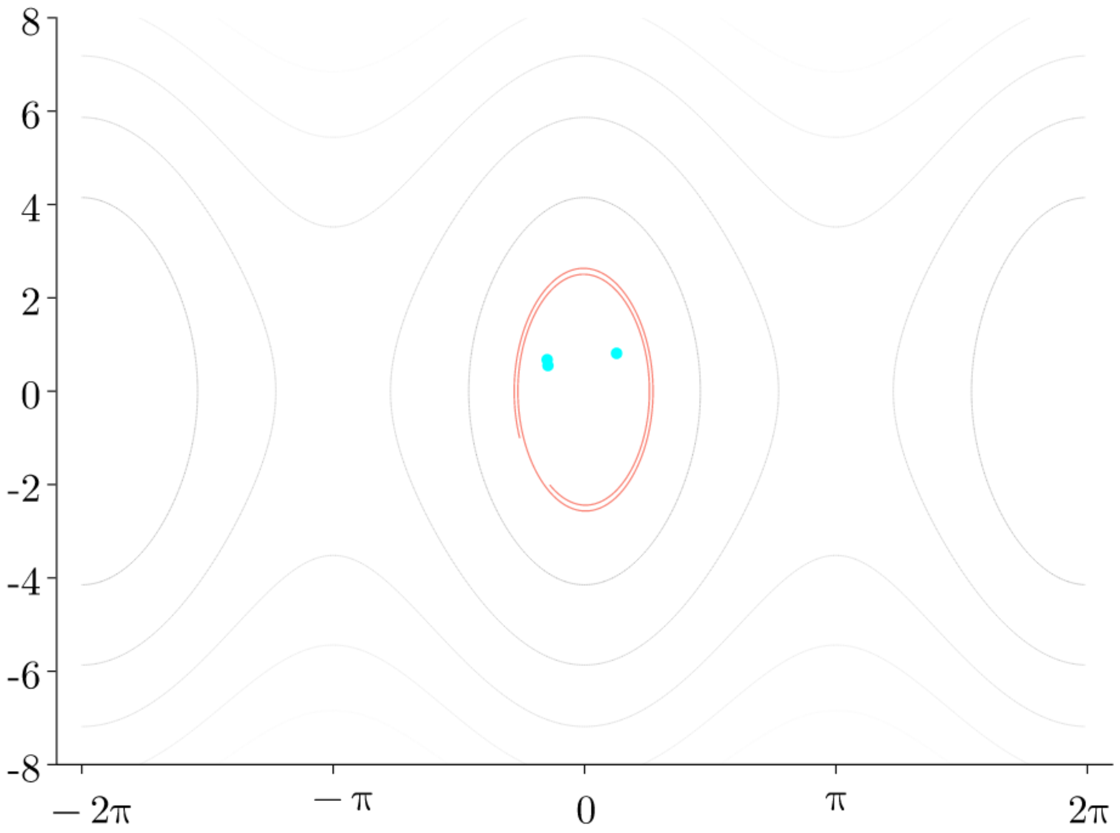
NeuralPbc Sampling State Space 📉



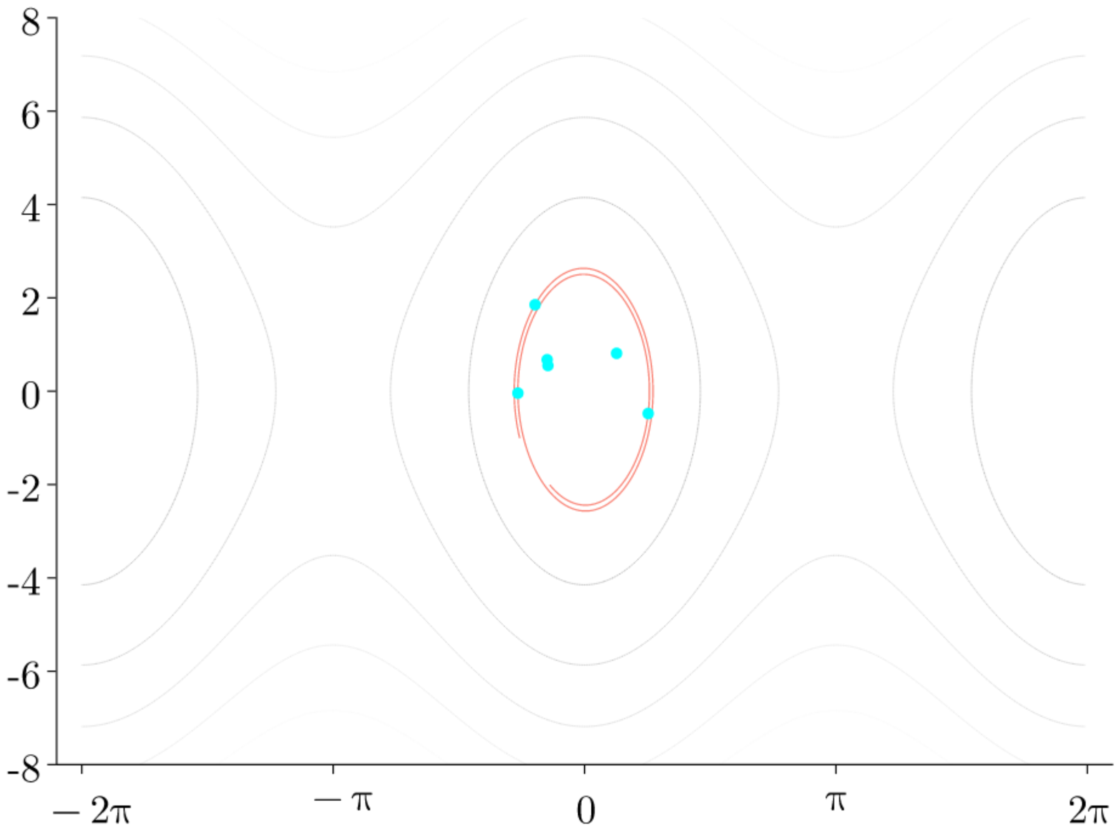
NEURALPBC Sampling State Space 📉



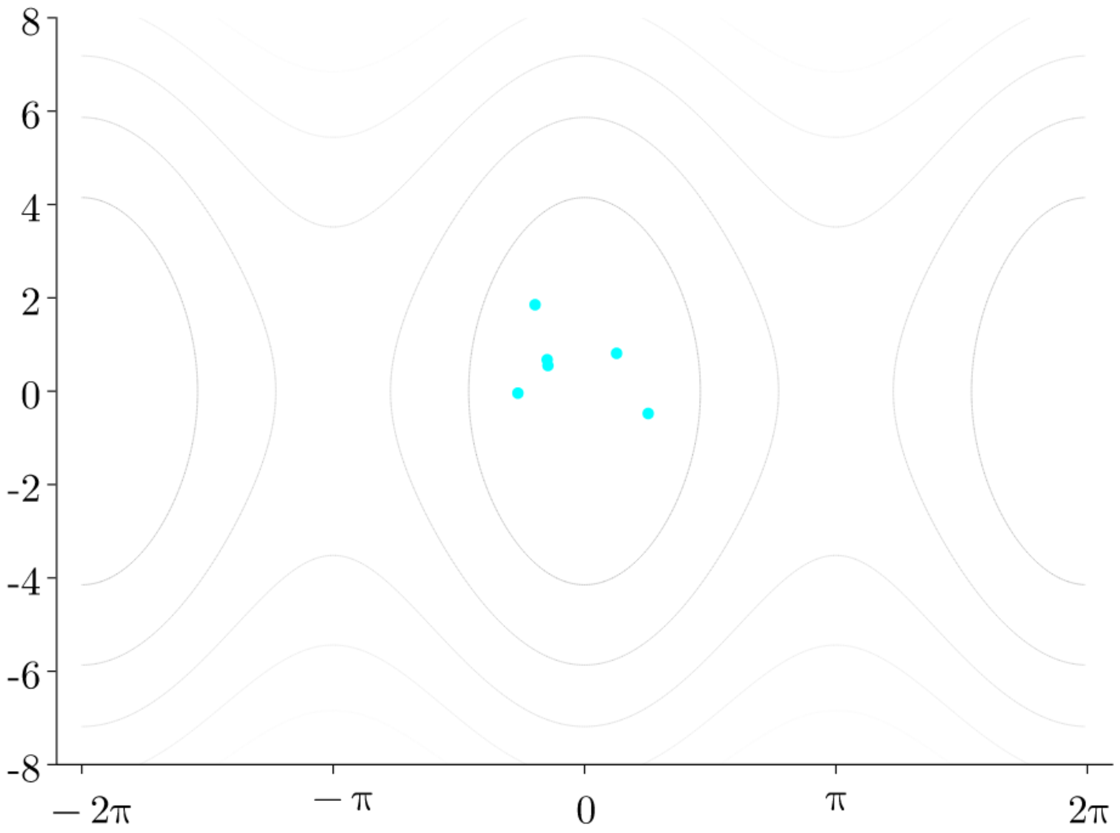
NeuralPbc Sampling State Space 📉



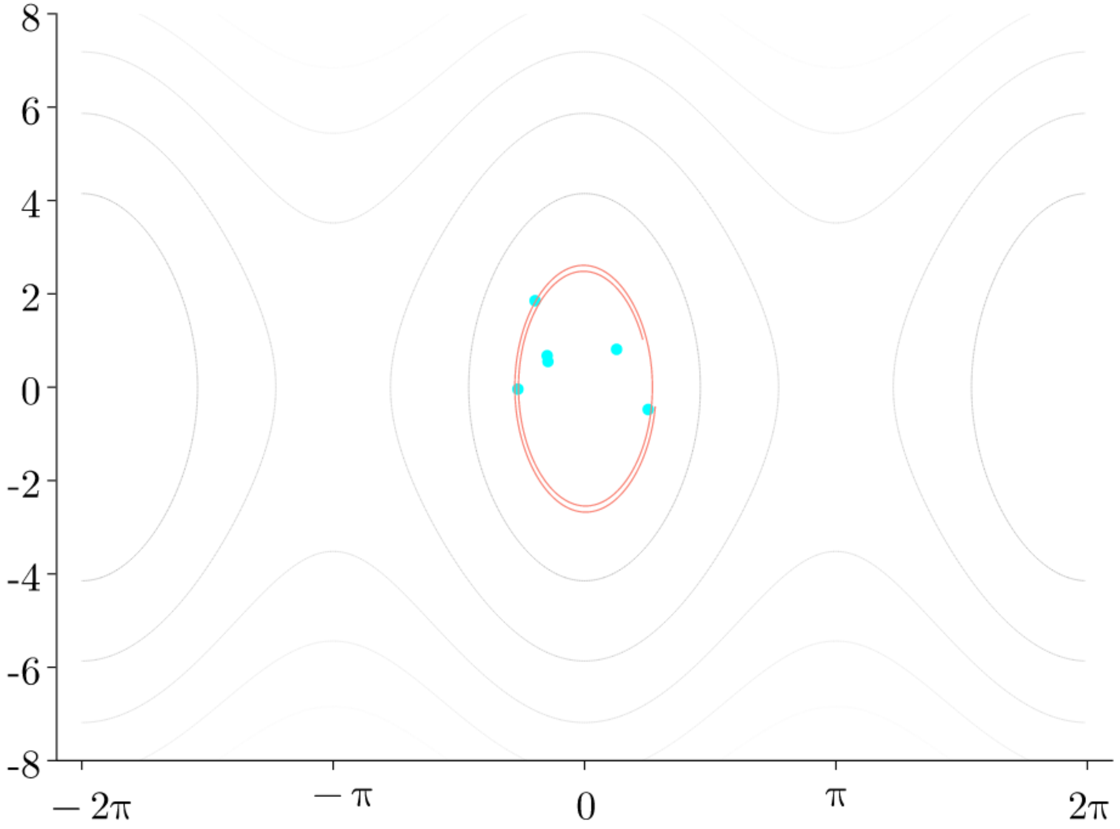
NEURALPBC Sampling State Space 📉



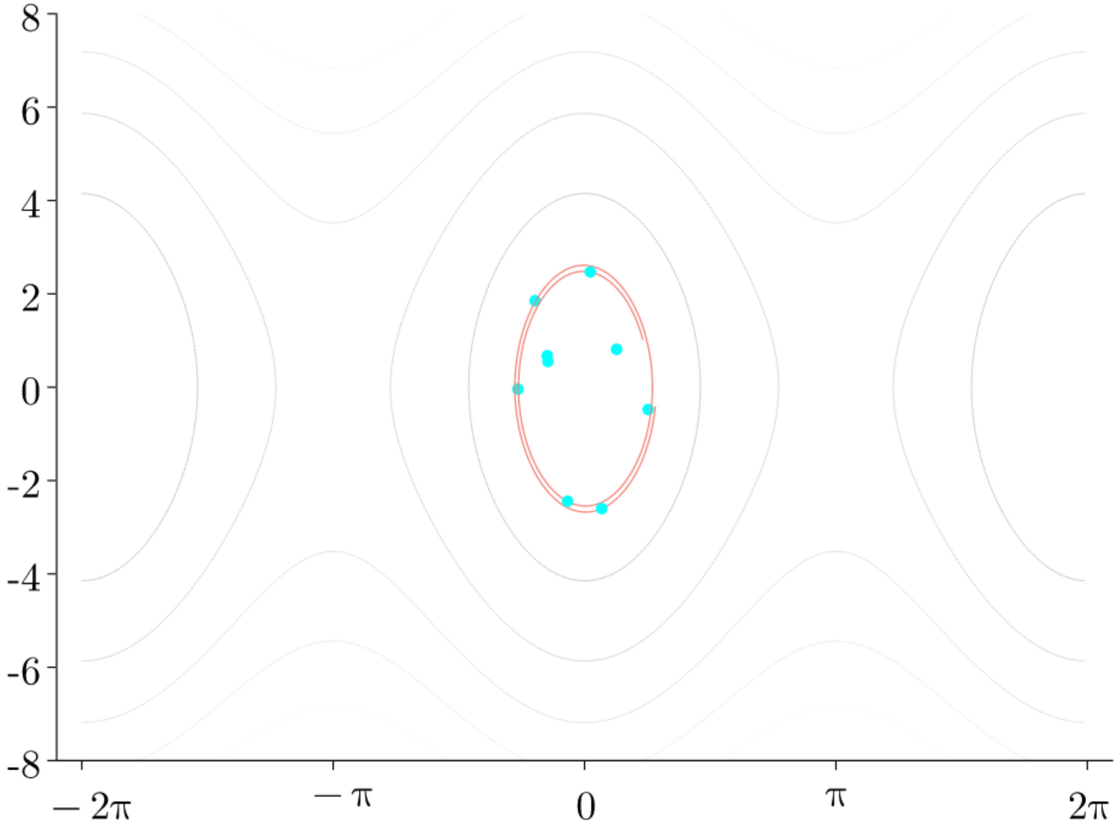
NeuralPbc Sampling State Space 📉



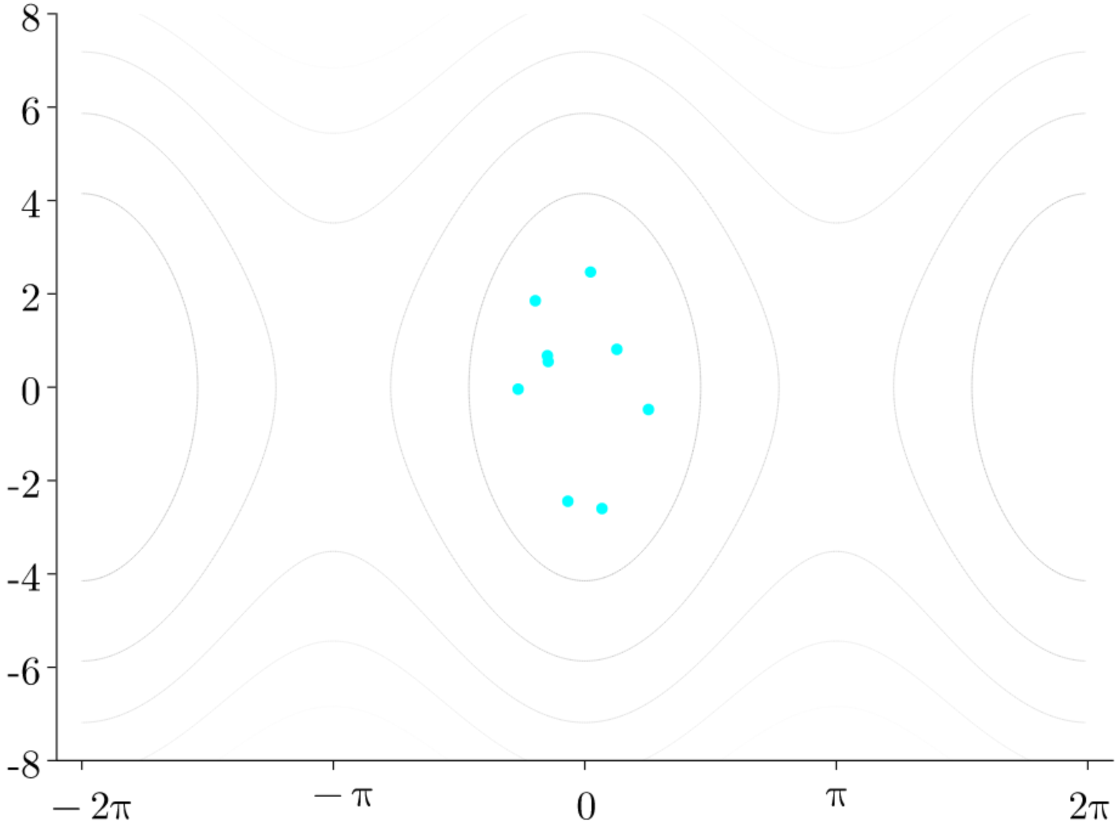
NeuralPbc Sampling State Space 📉



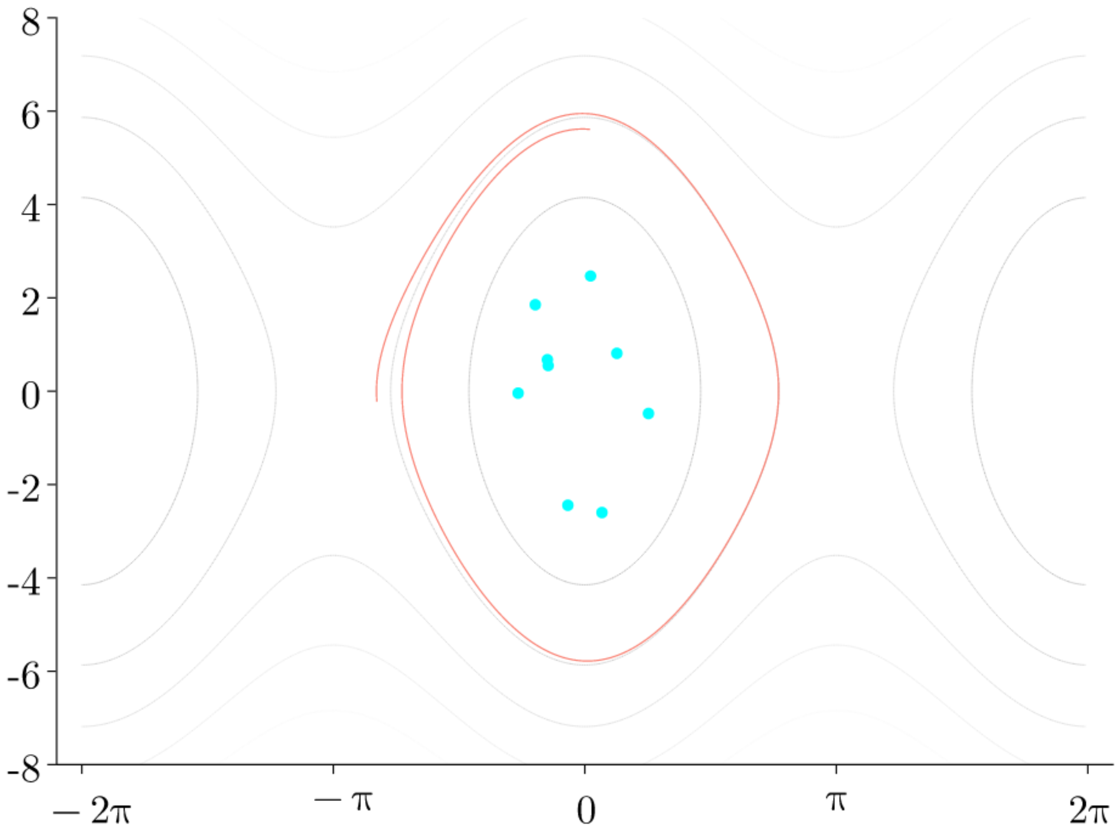
NeuralPbc Sampling State Space 📉



NeuralPbc Sampling State Space 📉



NeuralPbc Sampling State Space 📉



NeuralPbc Sampling State Space 📉