ROBUST CONTROL OF DYNAMICAL SYSTEMS VIA NEURAL BAYESIAN LEARNING

Dissertation Proposal

Presenter : Nardos Ayele Ashenafi
Advisor : Dr. Aykut Satici

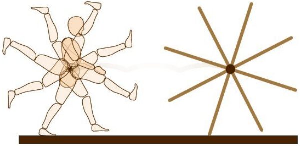
Electrical and Computer Engineering Department
Boise State University, Boise, ID 83725 USA

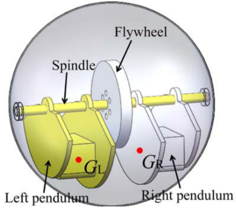
Published Work

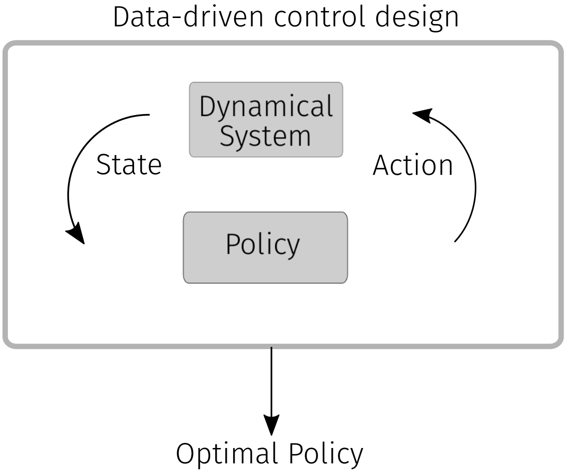
- N. A. Ashenafi, W. Sirichotiyakul, and A. C. Satici, “Robust passivity-based control of underactuated systems via neural approximators and Bayesian inference,” in 2022 IEEE Control Systems Letters (under review), 2022.
- N. A. Ashenafi and A. C. Satici, “Nonholonomic cooperative manipulation in the plane using linear complementarity formulation,” in 2021 IEEE Conference on Control Technology and Applications (CCTA). IEEE, 2021, pp. 634–639.
- W. Sirichotiyakul, N. A. Ashenafi, and A. C. Satici, “Robust data-driven passivity-based control of underactuated systems via neural approximators and Bayesian inference,” in 2022 American Control Conference, 2022.
- W. Sirichotiyakul, N. A. Ashenafi, and A. C. Satici, “Robust interconnection and damping assignment passivity-based control via neural Bayesian inference,” in 2022 IEEE Transactions on Automatic Control (under review), 2022.

Motivation

Data-Driven Control of Underactuated Systems

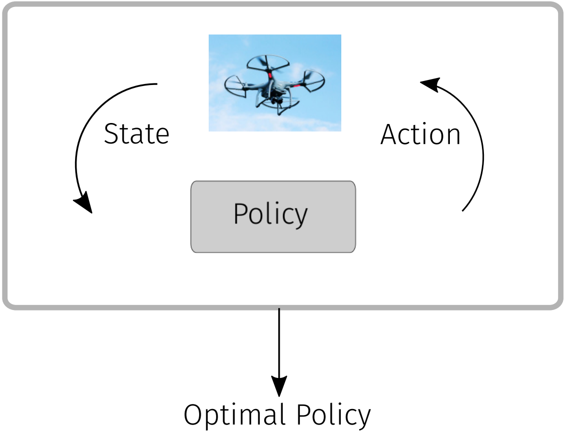






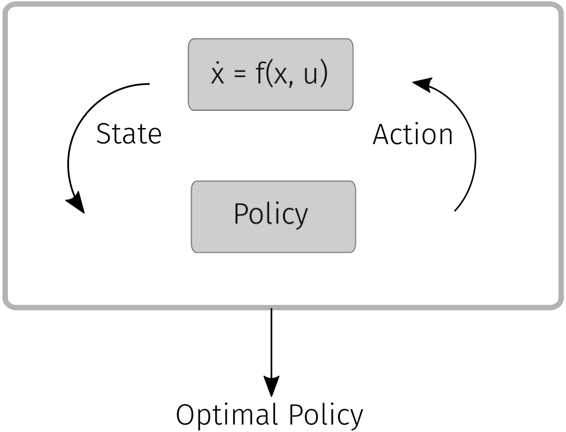
[1] Hu, Yibo, Yanding Wei, and Mengnan Liu. "Design and performance evaluation of a spherical robot assisted by high-speed rotating flywheels for self-stabilization and obstacle surmounting." Journal of Mechanisms and Robotics 13.6 (2021)

Learning from hardware



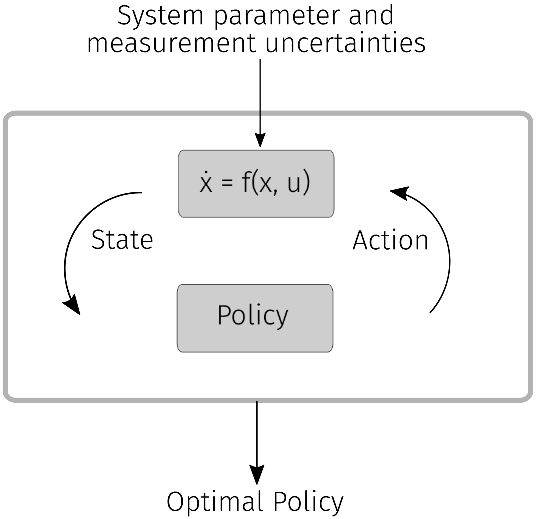
- Accurate system response
- Minimal insight to stability

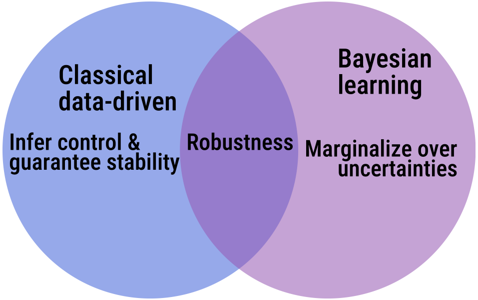
Learning from simulation



- Provides stability analysis
- Inaccurate system response

Our Method



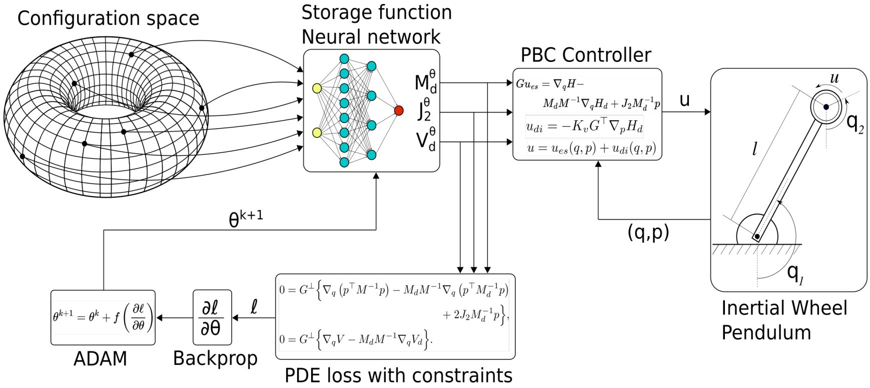


- Provides stability analysis
- Desired performance for wide range of system parameters

Interconnection and Damping Assignment Passivity-Based Control (IDAPBC)

- In the control of mechanical systems, IDAPBC is used to find a policy that overrides the overall energy of the system with a fictitious one with desirable characteristics.

$$H(q,p) = \frac{1}{2} p^\top M^{-1}(q) p + V(q) \xRightarrow{u_{es}+u_{di}} H_d(q,p) = \frac{1}{2} p^\top M_d^{-1}(q) p + V_d(q)$$



Main Learning Problem

$$
\begin{aligned}
\underset{\theta}{\text{minimize}} \qquad & J = \left\| G^{\perp} \left(\nabla_q H - M_d M^{-1} \nabla_q H_d + J_2 M_d^{-1} p \right) \right\|^2 \\
\text{subject to} \qquad & M_d^{\theta} = (M_d^{\theta})^{\top} \succ 0, \\
& J_2^{\theta} = -(J_2^{\theta})^{\top}, \\
& q^{\star} = \underset{q}{\text{argmin}}\; V_d^{\theta}.
\end{aligned}
$$

$V_d^{\theta}, M_d^{\theta}, J_2^{\theta}$ are parametrized by neural networks.

Theoretical Justification of the Robustness Properties of Bayesian Learning

Proof of concept: Compare the performance of deterministic and stochastic optimal control

Deterministic

$$\begin{aligned} \min_{\theta} \quad & \mathcal{J} = \int_0^T \left(\frac{1}{2} q x(t)^2 + \frac{1}{2} r u(t)^2 \right) dt \\ \text{subject to} \quad & \dot{x} = \hat{p} x + u, \\ & u(x) = \theta x. \end{aligned}$$

Stochastic

$$\begin{aligned} \min_{f(\theta)} \quad & \mathcal{J} = \int_0^T \left(\frac{1}{2} q x(t)^2 + \frac{1}{2} r u(t)^2 \right) dt \\ \text{subject to} \quad & \dot{x} = p x + u, \\ & u(x) = \theta x, \\ & \theta \sim f(\theta), \\ & p \sim \mathcal{N}(\hat{p}, \sigma_p^2). \end{aligned}$$

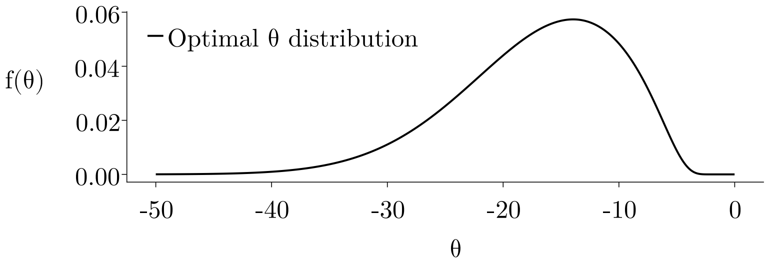
$$x(t) = e^{(p+\theta)t}, \quad x(0) = 1,$$
$$u(x) = \theta x.$$

Deterministic policy

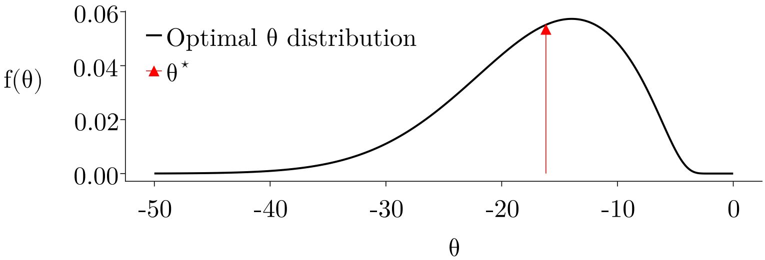
$$\mathcal{J} = \int_0^T \left(\frac{1}{2} q x^2 + \frac{1}{2} r u^2 \right) dt,$$
$$\mathcal{J}_\infty = -\frac{1}{4} \frac{q + r\theta^2}{\hat{p} + \theta},$$
$$\nabla_\theta \mathcal{J}_\infty = -\frac{r}{4} \frac{(\hat{p}+\theta)^2 - \left(\hat{p}^2 + q/r\right)}{(\hat{p}+\theta)^2} = 0,$$
$$\theta^\star = -\hat{p} - \sqrt{\hat{p}^2 + q/r}.$$

Stochastic policy

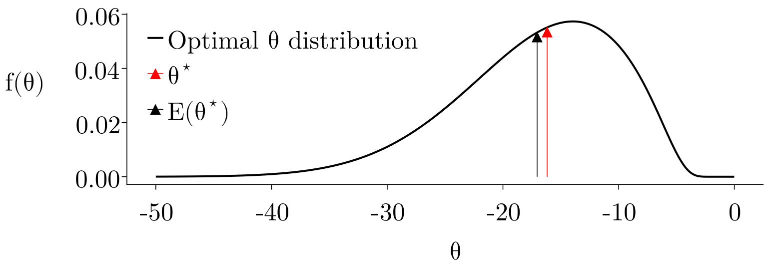
$$g(p) := -p - \sqrt{p^2 + q/r},$$
$$f_{\theta^\star}(\theta^\star) = f_p\left(g^{-1}(\theta^\star)\right) \left| \frac{d}{d\theta} g^{-1}(\theta^\star) \right|.$$



- $\hat{p} = 5, \sigma_p = 5$
- The stochastic solution gives an exponential distribution over optimal θ.



- $\hat{p} = 5, \sigma_p = 5$
- The stochastic solution gives an exponential distribution over optimal θ.
- Red shows deterministic solution.



- $\hat{p} = 5, \sigma_p = 5$
- The deterministic solution is completely unaware of the effects of uncertainties. For e.g., the true dynamics may be unstable under $\theta^\star$. The stochastic controller reasons about stability for various parameters.

Stochastic optimal control

$$
\begin{aligned}
\min_{f(\theta)} \quad & \mathcal{J} = \int_0^T \left(\frac{1}{2} q x(t)^2 + \frac{1}{2} r u(t)^2 \right) dt \\
\text{subject to} \quad & \mathrm{d}x = (p + \theta) x(t)\,\mathrm{d}t + \theta\sigma\,\mathrm{d}W_t, \\
& \theta \sim f(\theta), \\
& p \sim \mathcal{N}(\hat{p}, \sigma_p^2).
\end{aligned}
$$

where W_t is the Wiener process, σ is the standard deviation of the noise.

- With $x(0) = 1$, the solution to the SDE is

$$x(t) = e^{(p+\theta)t} + \theta\sigma \int_0^t e^{(p+\theta)(t-s)} dW_s.$$

- We can compute the expected cost numerically as

$$\mathbb{E}[\mathcal{J}] = \mathbb{E}_p[\mathbb{E}_W[\mathcal{J} \mid p]].$$

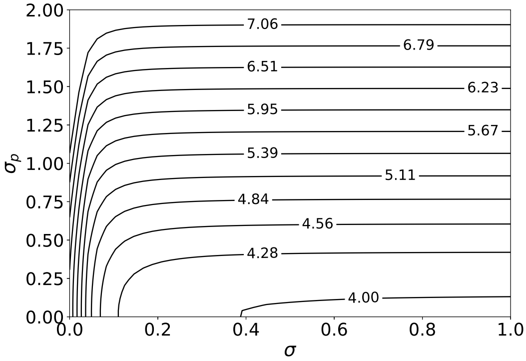


Figure 1: The optimal controller parameter magnitude $|\theta^\star|$ that minimizes $\mathbb{E}[\mathcal{J}]$

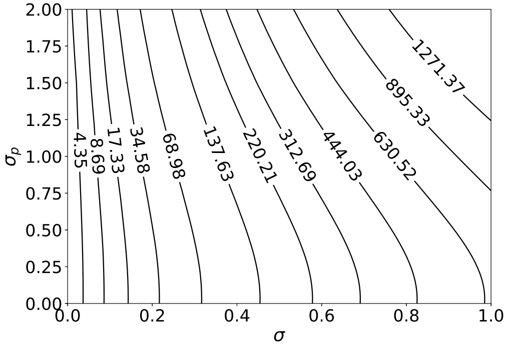


Figure 2: The minimal expected cost $\mathbb{E}[\mathcal{J}]$

Discussion

- It is more robust to develop a stochastic controller given system parameter uncertainties and measurement noise.
- We employ Bayesian learning to design a stochastic policy from data-driven techniques.

Background: Bayesian Learning

- Objective : Learn a target function $F(x;\theta) : \mathcal{X} \rightarrow \mathbb{R}^t$ that best parameterizes the source of a dataset $\mathcal{D}$ with inherent noise.
- The target function is parameterized by the random variable θ.
- The task is to learn the distribution of θ given prior belief.

$$p(\theta|\mathcal{D}) = \underbrace{\frac{\overbrace{p(\mathcal{D} \mid \theta)}^{\textit{likelihood}}\overbrace{p(\theta)}^{\textit{prior}}}{\int_\theta p(\mathcal{D} \mid \theta')p(\theta')d\theta'}}_{\textit{evidence}}$$

- The evidence is intractable in most cases.
- There are techniques used to find the exact or an approximate posterior.

Posterior Distribution: Variational Inference

- Variational inference: select a distribution $q(\theta; z)$ and adjust the distribution parameters z to best fit to $p(\theta \mid \mathcal{D})$.
- The objective is to learn the distribution parameters z, such that the Kullback-Leibler (KL) divergence given by

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

  is minimized.
- This is similar to maximizing the *evidence lower bound* (ELBO):

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

Neural Bayesian Learning (NBL):

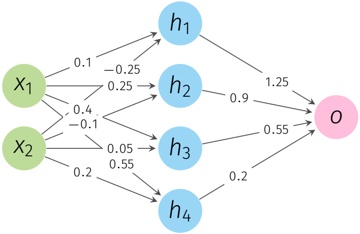
Robust Data-Driven Stochastic Control Design

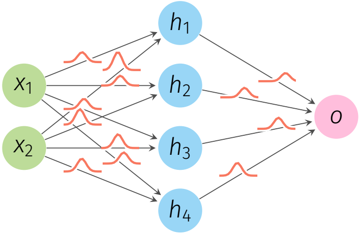
Completed Work

Neural Bayesian Learning

$$
\begin{aligned}
\min_{q(\theta)} \quad & J(\phi(t; x_0, u), u) \\
\text{subject to} \quad & dx = f(x, u; \zeta)\, dt + \nabla_x u(x)\, dW_t, \\
& u(x; \theta) = \mathcal{D}\{F(x; \theta)\}, \\
& \zeta \sim \mathcal{N}(\zeta_0, \Sigma_\zeta), \\
& \theta \sim q(\theta).
\end{aligned}
$$

- $F(x; \theta)$ is a *Bayesian neural network*.
- $\phi(t; x_0, u)$ is a trajectory generated from policy u starting at initial state x_0.
- ζ is a vector of system parameters with uncertainties Σ_ζ.





Algorithm 1 Neural Bayesian Learning

Select a prior distribution $p(\theta)$
$f \leftarrow f(x, u^\theta; \zeta)$ dynamics given by SDE using current policy u^θ
$\mathcal{D} \leftarrow \{x_0\}_{(N_\mathcal{D})}$ ▷ $N_\mathcal{D}$ samples of x_0
for $i = 1 : \texttt{maxiters}$ **do**
  for each $d \subset \mathcal{D}$ **do** ▷ Select batch d from dataset $\mathcal{D}$
    Initialize $\mathcal{J} = 0$
    $\theta \sim q(\theta; z)$ ▷ Sample parameters of F from posterior
    for each $x_0 \in d$ **do**
      $\zeta \sim \mathcal{N}(\zeta_0, \Sigma_\zeta)$ ▷ Sample system parameters
      $\phi(t; x_0, u^\theta) \leftarrow$ integrate dynamics from x_0
      $\mathcal{J} \leftarrow \mathcal{J} + J(\phi(t; x_0, u^\theta), u^\theta)$
    Compose likelihood $p(d|\theta)$
    Compute ELBO $\mathcal{L}(J, z) = \mathbb{E}_{\theta \sim q}\left[\log(p(J \mid \theta; z)p(\theta; z)) - \log(q(\theta; z))\right]$
    $z \leftarrow z + \alpha\, \partial\mathcal{L}/\partial z$
  $\mathcal{D} \leftarrow \{\mathcal{D}\}_{(1,\ldots,N_\mathcal{D}-N_\mathrm{R})} \cup \{x_0\}_{(N_\mathrm{R})}$ ▷ Replay buffer
return $q(\theta; z)$

Algorithm 1 Neural Bayesian Learning

Select a prior distribution $p(\theta)$
$f \leftarrow f(x, u^\theta; \zeta)$ dynamics given by SDE using current policy u^θ
$\mathcal{D} \leftarrow \{x_0\}_{(N_\mathcal{D})}$ ▷ $N_\mathcal{D}$ samples of x_0
for $i = 1 : \texttt{maxiters}$ **do**
  for each $d \subset \mathcal{D}$ **do** ▷ Select batch d from dataset $\mathcal{D}$
    Initialize $\mathcal{J} = 0$
    $\theta \sim q(\theta; z)$ ▷ Sample parameters of F from posterior
    for each $x_0 \in d$ **do**
      $\zeta \sim \mathcal{N}(\zeta_0, \Sigma_\zeta)$ ▷ Sample system parameters
      $\phi(t; x_0, u^\theta) \leftarrow$ integrate dynamics from x_0
      $\mathcal{J} \leftarrow \mathcal{J} + J(\phi(t; x_0, u^\theta), u^\theta)$
    Compose likelihood $p(d|\theta)$
    Compute ELBO $\mathcal{L}(J, z) = \mathbb{E}_{\theta\sim q}\left[\log(p(J \mid \theta; z)p(\theta; z)) - \log(q(\theta; z))\right]$
    $z \leftarrow z + \alpha\, \partial\mathcal{L}/\partial z$
  $\mathcal{D} \leftarrow \{\mathcal{D}\}_{(1,\ldots,N_\mathcal{D}-N_\mathrm{R})} \cup \{x_0\}_{(N_\mathrm{R})}$ ▷ Replay buffer
return $q(\theta; z)$

Algorithm 1 Neural Bayesian Learning

Select a prior distribution $p(\theta)$
$f \leftarrow f(x, u^\theta; \zeta)$ dynamics given by SDE using current policy u^θ
$\mathcal{D} \leftarrow \{x_0\}_{(N_\mathcal{D})}$ ▷ $N_\mathcal{D}$ samples of x_0
for $i = 1 : \texttt{maxiters}$ **do**
  for each $d \subset \mathcal{D}$ **do** ▷ Select batch d from dataset $\mathcal{D}$
    Initialize $\mathcal{J} = 0$
    $\theta \sim q(\theta; z)$ ▷ Sample parameters of F from posterior
    for each $x_0 \in d$ **do**
      $\zeta \sim \mathcal{N}(\zeta_0, \Sigma_\zeta)$ ▷ Sample system parameters
      $\phi(t; x_0, u^\theta) \leftarrow$ integrate dynamics from x_0
      $\mathcal{J} \leftarrow \mathcal{J} + J(\phi(t; x_0, u^\theta), u^\theta)$
    Compose likelihood $p(d|\theta)$
    Compute ELBO $\mathcal{L}(J, z) = \mathbb{E}_{\theta\sim q}\left[\log(p(J \mid \theta; z)p(\theta; z)) - \log(q(\theta; z))\right]$
    $z \leftarrow z + \alpha\, \partial\mathcal{L}/\partial z$
  $\mathcal{D} \leftarrow \{\mathcal{D}\}_{(1,\ldots,N_\mathcal{D}-N_\mathrm{R})} \cup \{x_0\}_{(N_\mathrm{R})}$ ▷ Replay buffer
return $q(\theta; z)$

Algorithm 1 Neural Bayesian Learning

Select a prior distribution $p(\theta)$
$f \leftarrow f(x, u^\theta; \zeta)$ dynamics given by SDE using current policy u^θ
$\mathcal{D} \leftarrow \{x_0\}_{(N_\mathcal{D})}$ ▷ $N_\mathcal{D}$ samples of x_0
for $i = 1 : \texttt{maxiters}$ **do**
    for each $d \subset \mathcal{D}$ **do** ▷ Select batch d from dataset $\mathcal{D}$
        Initialize $\mathcal{J} = 0$
        $\theta \sim q(\theta; z)$ ▷ Sample parameters of F from posterior
        for each $x_0 \in d$ **do**
            $\zeta \sim \mathcal{N}(\zeta_0, \Sigma_\zeta)$ ▷ Sample system parameters
            $\phi(t; x_0, u^\theta) \leftarrow$ integrate dynamics from x_0
            $\mathcal{J} \leftarrow \mathcal{J} + J(\phi(t; x_0, u^\theta), u^\theta)$
        Compose likelihood $p(d|\theta)$
        Compute ELBO $\mathcal{L}(J, z) = \mathbb{E}_{\theta \sim q}\left[\log(p(J \mid \theta; z) p(\theta; z)) - \log(q(\theta; z))\right]$
        $z \leftarrow z + \alpha\, \partial\mathcal{L}/\partial z$
    $\mathcal{D} \leftarrow \{\mathcal{D}\}_{(1,\ldots,N_\mathcal{D} - N_\mathrm{R})} \cup \{x_0\}_{(N_\mathrm{R})}$ ▷ Replay buffer
return $q(\theta; z)$

Preliminary Results

Tracking Trajectory

- Let $\gamma : t \to \phi(t; x_0, u^\theta)$ represent the flow of the dynamical system.
- The objective is to learn a closed-loop controller that tracks an expert trajectory $\gamma^\star$ provided by a path planner.
- The running cost that achieves this is given by

$$J_{track}(\gamma, \gamma^\star) = \sum_i ||\gamma_i - \gamma_i^\star||^2 .$$

- The performance of NBL is tested on the swing-up task of the simple pendulum.

- The stochastic policy is marginalized over the posterior:

$$u(x) = \frac{1}{N} \sum_{\theta \sim q(\theta; z)} \mathcal{D}\{F(x; \theta)\}.$$

- Deterministic and stochastic policies are compared against system parameter and measurement uncertainties.
- Measurement noise on position $\epsilon_q = 0.0005$ radians, and velocity $\epsilon_{\dot{q}} = 0.05$ radians/s.

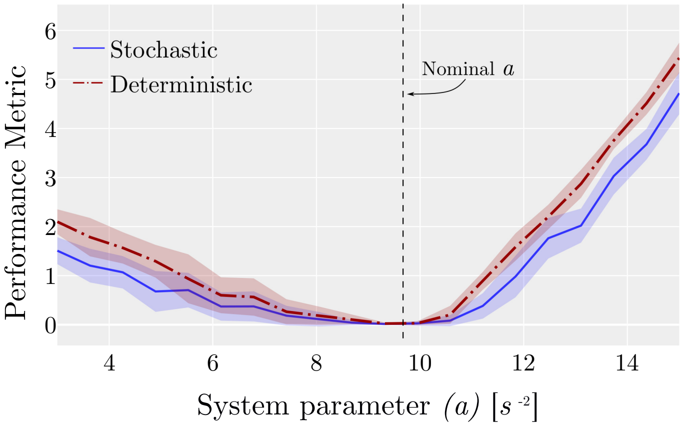


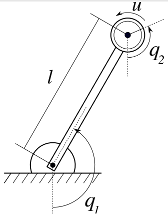
Figure 3: Performance metric is the minimum loss to the expert trajectory over the last 2 seconds of a 10-second-long trajectory

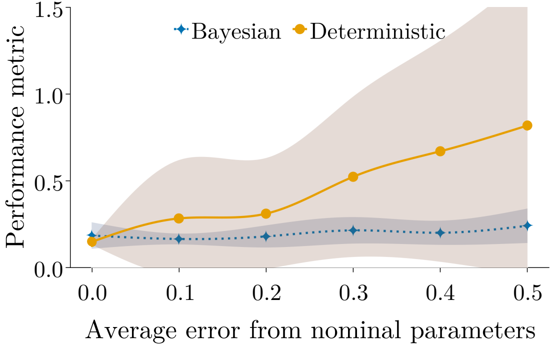
IdaPbc on the Inertia Wheel Pendulum: Simulation Tests

- The objective is to use the momentum from the wheel to swing up the pendulum.
- The dynamics of the system under measurement noise is

$$\mathrm{d}x = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dfrac{mgl\sin(q_1) - u^\theta - b_1\dot{q}_1}{I_1} \\ \dfrac{u^\theta - b_2\dot{q}_2}{I_2} \end{bmatrix} \mathrm{d}t + \nabla_x u^\theta(x)\sigma\,\mathrm{d}W_t$$

- Performance metric is $\int_0^T \left(\frac{1}{2}qx^2 + \frac{1}{2}ru^2\right) dt$.





IdaPbc on the Inertia Wheel Pendulum: Experiments

- The mass of the wheel is varied according to Table 1.
- The same controller is tested on various system parameters.

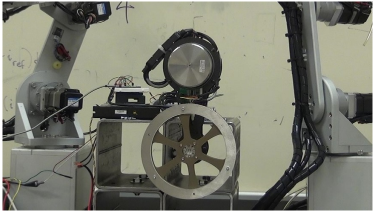


Table 1: System parameters used in real-world experiments. The errors in the last column are $\|\zeta - \zeta_0\| / \|\zeta_0\|$

Parameter set ζ	I_1	I_2	mgl	Error
Nominal	0.0455	0.00425	1.795	0
3 Rings	0.0417	0.00330	1.577	0.122
2 Rings	0.0378	0.00235	1.358	0.243
1 Ring	0.0340	0.00141	1.140	0.365

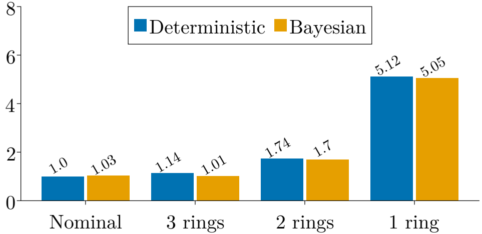
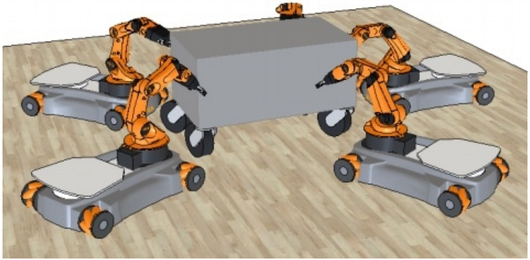


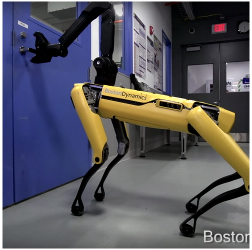
Figure 4: Performance metric is $\int_0^T \left(\frac{1}{2}qx^2 + \frac{1}{2}ru^2\right) dt$

Future Work

NBL control design through contacts

- The NBL technique can be extended to more complicated systems such as contact-rich systems
- The objective is to design robust data-driven controllers that exhibit desired properties and account for uncertainties in the hybrid dynamical model





[1]Antonio Petitti, Antonio Franchi, Donato Di Paola, Alessandro Rizzo. Decentralized Motion Control for Cooperative Manipulation with a Team of Networked Mobile Manipulators. 2016 IEEE Interna- tional Conference on Robotics & Automation (ICRA), May 2016, Stockholm, Sweden. pp. 441-446.

- Contacts, impacts and Coulomb friction are modelled through linear complementarity problems (LCP) given by the following convex optimization problem

Linear Complementarity Problem

$$\begin{aligned} \underset{\xi_n,\xi_t,\Lambda_n,\Lambda_t}{\text{minimize}} \quad & 0 \\ \text{subject to} \quad & 0 \leq \begin{pmatrix} \xi_n \\ \xi_t \\ \Lambda_l \end{pmatrix} \perp \begin{pmatrix} \Lambda_n \\ \Lambda_r \\ \xi_l \end{pmatrix} \geq 0, \\ & \begin{pmatrix} \xi_n \\ \xi_r \\ \Lambda_l \end{pmatrix} = A \begin{pmatrix} \Lambda_n \\ \Lambda_r \\ \xi_l \end{pmatrix} + B \end{aligned}$$

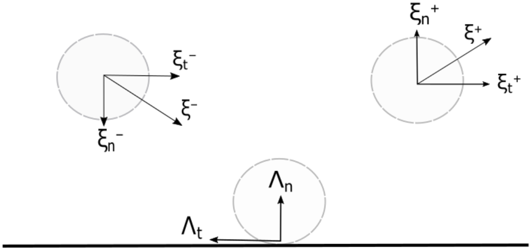


Figure 5: Bouncing ball contact event

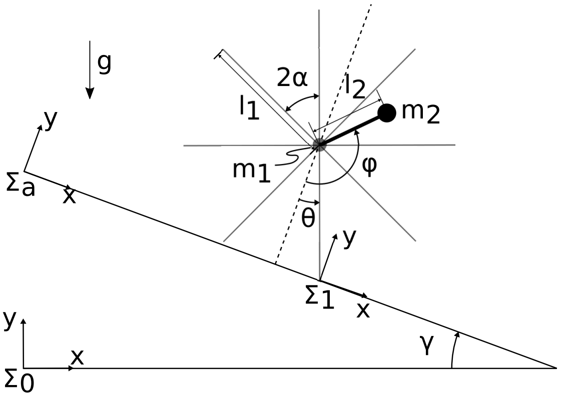
Λ represents contact forces and ξ are the post-contact velocities, in the tangential and normal directions.

Neural Bayesian Learning with Contacts, Impacts and Coulomb Friction

$$
\begin{aligned}
\min_{q(\theta)} \quad & J(\phi(t; x_0, u), u) \\
\text{subject to} \quad & M(x)\,\mathrm{d}x - h(x, \dot{x}, u)\,\mathrm{d}t - \mathrm{d}\Lambda = 0, \\
& u(x; \theta) = \mathcal{D}\{F(x; \theta)\}, \\
& \zeta \sim \mathcal{N}(\zeta_0, \Sigma_\zeta), \\
& \theta \sim q(\theta),
\end{aligned}
$$

where M is the mass matrix, h holds Coriolis terms and generalized forces and Λ holds the normal and tangential contact forces on the system.

- The rimless wheel will be used as a test bed for the NBL control design.
- We will design a robust controller under system parameter uncertainties such as surface friction.



[1] Sirichotiyakul, Wankun, et al. "Energetically-optimal discrete and continuous stabilization of the rimless wheel with torso." International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Vol. 59230. American Society of Mechanical Engineers, 2019.

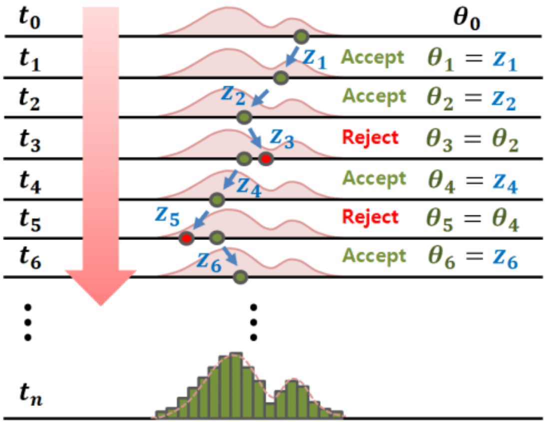
Table 2: Schedule

Timeline	Task
May - August 2022	Design and build the rimless wheel
August - September 2022	Learn deterministic policy of the rimless-wheel
October - November 2022	Learn stochastic policy of the rimless-wheel
November - December 2022	Submit work to a journal publication
December 2022 - February 2023	Write dissertation
March - May 2023	Prepare dissertation presentation and defend

Questions?

Appendix

- Markov Chain Monte Carlo (MCMC) methods: estimate the exact posterior distribution by drawing samples generated from a Markov chain.
- The samples are accepted or rejected based on a criterion.



[1] Jin, Seung-Seop & Ju, Heekun & Jung, Hyung-Jo. (2019). Adaptive Markov chain Monte Carlo algorithms for Bayesian inference: recent advances and comparative study. Structure and Infrastructure Engineering. 10.1080/15732479.2019.1628077.

- The system's equations of motion can then be expressed as

$$\begin{bmatrix}\dot{q}\\ \dot{p}\end{bmatrix} = \begin{bmatrix}0 & I_n\\ -I_n & 0\end{bmatrix}\begin{bmatrix}\nabla_q H\\ \nabla_p H\end{bmatrix} + \begin{bmatrix}0\\ G(q)\end{bmatrix} u, \tag{1}$$

where $G(q) \in \mathbb{R}^{n\times m}$ is the input matrix, I_n is the $n \times n$ identity matrix, and $u \in \mathcal{U} \subset \mathbb{R}^m$ is the control input.

- In IDAPBC, the closed-loop dynamics is chosen as the port-controlled Hamiltonian (PCH) form:

$$\begin{bmatrix}\dot{q}\\ \dot{p}\end{bmatrix} = \begin{bmatrix}0 & M^{-1}M_d\\ -M_d M^{-1} & J_2(q,p) - GK_vG^\top\end{bmatrix}\begin{bmatrix}\nabla_q H_d\\ \nabla_p H_d\end{bmatrix}, \tag{2}$$

where $J_2 = -J_2^\top$

Tracking Trajectory

- Let $\gamma : t \to \phi(t; x_0, u^\theta)$ represent the flow of the dynamical system
- The objective is to track an expert trajectory $\gamma^\star$ provided by a path planner
- The running cost that achieves this is given by

$$J_{track}(\gamma_\perp) = \sum_{x_\perp \in \gamma_\perp,\, x_\perp^\star \in \gamma_\perp^\star} ||x_\perp - x_\perp^\star||^2$$

  where $(\cdot)_\perp$ expresses the state in transverse coordinates along the desired orbit $\gamma^\star$
- The corresponding likelihood is

$$\begin{aligned} p(\mathcal{D} \mid \theta) = p(\gamma_\perp|\theta) &= \prod_{x_\perp \in \gamma_\perp,\, x_\perp^\star \in \gamma_\perp^\star} \mathcal{N}(x_\perp^\star, \Sigma) \\ &= \prod_{x_\perp \in \gamma_\perp,\, x_\perp^\star \in \gamma_\perp^\star} \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}(x_\perp - x_\perp^\star)^\top \boldsymbol{\Sigma}^{-1}(x_\perp - x_\perp^\star)} \end{aligned}$$