D-optimal neural exploration of nonlinear physical systems

Matthieu Blanke
matthieu.blanke@inria.fr
INRIA, DI ENS

Marc Lelarge
marc.lelarge@inria.fr
INRIA, DI ENS

Abstract

Exploring an unknown physical environment in a sample-efficient and computationally fast manner is a challenging task. In this work, we introduce an exploration policy based on neural networks and experimental design. Our policy maximizes the one-step-ahead information gain on the model, which is computed using automatic differentiation, and leads us to an online exploration algorithm requiring small computing resources. We test our method on a number of nonlinear physical systems covering different settings.

1 Introduction

Building a faithful model of a physical system is essential for designing an efficient control law in a model-based approach. In this regard, active system identification — or exploration — aims at exciting the system to collect informative data, in order to estimate the system in a sample-efficient manner [1, 2]. In many cases, an exact model is known only for part of the dynamics, either because the system is too complex to derive an analytical model, or simply because some effects that one wishes to take into account are unknown. Since little is known about those effects a priori, the unknown component is assumed to be a nonlinear function of the state. An example would be air friction on a drone, for which finding a model from physical principles is notoriously difficult [3]. We focus on dynamical systems, where the state x and the input u are governed by an equation of the form:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x, u). \tag{1}$$

What one observes in practice are discrete noisy observations of the dynamics (1):

$$x_{t+1} = x_t + \mathrm{d}t f(x_t, u_t) + w_t, \quad 0 \leq t \leq T-1, \tag{2}$$

where $\mathrm{d}t$ is a known time step, $x_t \in \mathbb{R}^d$ is the state vector, $w_t \sim \mathcal{N}(0, \mathrm{d}t\, \sigma^2 I_d)$ is a normally distributed isotropic noise with known variance σ^2, and the control variables $u_t \in \mathbb{R}^m$ are chosen by the controller with the constraint $\|u_t\|_2 \leq \gamma$. Our aim is to learn the flow f with a parametric function f_θ, whose parameters are gathered in a vector $\theta \in \mathbb{R}^q$. The goal of active system identification is to choose inputs (u_t) that make the trajectory as informative as possible for the estimation of f with f_θ, with the input u_t chosen at time t using the past observations. The computational process of choosing the inputs is called planning. Typically, we want to derive a planning objective $G(u)$ measuring the information gain of the input u, and choose u_t as the input maximizing G under the amplitude constraint. Since we want our algorithm to run online, we attach importance to the computational cost of planning. Typically, gradient-based methods are often too slow for practical uses.

Neural model for dynamics In our active exploration paradigm, a good model for such a nonlinear map should fulfill three conditions. First, it should be expressive enough to learn the target

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

function. Second, it should be able to learn efficiently from the data online. Third, it should have a mathematical measure of uncertainty to guide the choice of inputs, which should be cheap to evaluate and optimize, so that an exploration algorithm could use it to run online. Neural networks are a popular and powerful tool for modelling nonlinear functions, and they meet all of the three conditions. Neural nets are expressive and can be trained online with a constant memory cost [4]. As for measuring uncertainty, the optimal design theory provides an information-theoretic criterion [5, 6], for which we propose an efficient optimization algorithm (see Section 2). A general online neural exploration algorithm is summarized in Algorithm 1. In the case of linear dynamics, an online exploration algorithm based on optimal design input design for classical linear least squares has recently been proposed [7]. Our work can be seen as a generalization to nonlinear dynamics.

Contributions Building on the theory of optimal design for neural networks, we define a D-optimal planning objective for active neural exploration. We derive a tractable approximation of this cost function, which allows us to design an online exploration algorithm. The sample-efficiency of our method is demonstrated on various nonlinear physical systems and its performance is compared to several baselines.

Related work The pure exploration task has recently attracted much interest in the fields of control and reinforcement learning. Several methods have been proposed to learn the dynamics (or transition function) and the uncertainty of the learner, including Gaussian processes [8] (which have the drawback of requiring a quadratic memory cost), Random Fourier Features [9], and an ensemble of neural networks [10]. Experimental design approaches to identification of linear dynamical systems are studied in [7] and [11]. A theoretical study for active nonlinear system identification can be found in [12]. The extension of optimal design of experiments to neural networks is proposed in [5] for static systems, and in [13] for dynamical systems with offline training. More recently, this uncertainty measure for neural nets was used for neural contextual bandits [14].

2 Neural D-optimal exploration

Classical D-optimal design In our pure exploration framework, the choice of inputs for our controlled dynanics should be guided by some measure of uncertainty about our model f_θ. In the case of a scalar linear map $f_\theta(z) = z \cdot \theta$, the classical optimal design theory provides an information-theoretic criterion measuring the volume of the confidence ellipsoid for the parameter vector θ and called D-optimality [15, 16]:

$$\max_{(z_s)} \log\det\left(\mathbb{E}[M_T]\right) \quad \text{with} \quad M_t = \sum_{s=0}^{t-1} z_s {z_s}^\top \in \mathbb{R}^{d\times d}. \tag{3}$$

Linearized optimal design The D-optimality criterion can be extended to non-linear models, as follows [5]. Assuming that the parameter vector is close to the convergence value $\theta_\star$, we can linearize the target function to the first order in θ:

$$f_\theta(z,\theta) \simeq f_\theta(z,\theta_\star) + J_\theta(z,\theta_\star) \times (\theta - \theta_\star) \quad \text{with} \quad J_\theta(z,\theta) := \frac{\partial f_\theta}{\partial \theta}(z,\theta) \in \mathbb{R}^{d\times q}. \tag{4}$$

The evaluations at $\theta_\star$ can be approximated with those at the current estimate θ_t. In this linearization regime, the learning of f reduces to ordinary least squares for θ. Specifically, an observation at a point z corresponds to d distinct covariates of dimension q: $g_\theta^k(z,\theta) = \nabla_\theta f_\theta^k(z,\theta) \quad \in \mathbb{R}^q, \quad 1 \le k \le d$, which are concatenated into the Jacobian $J_\theta(z,\theta)$. Taking D-optimality for our information gain criterion, we should solve

$$\max_{(z_s)} \log\det\left(\mathbb{E}[M_T]\right) \quad \text{with} \quad M_t = \sum_{s=0}^{t-1} J_\theta(z_s,\theta_s)^\top J_\theta(z_s,\theta_s) \in \mathbb{R}^{q\times q}. \tag{5}$$

In our dynamic setting, we want to derive an optimization objective for planning from the information-theoretic criterion (5). Since we want to obtain an online policy, we focus on a one-step-ahead objective, which yields a greedy approximation of (5). Still, greedy planning in our dynamic setting has been found to perform well in practice [6, 7]. We define the next-step prediction of the state as $x(u) = x_t + \mathrm{d}t f_\theta(x_t, u)$, and consider J as a function of x as follows: $J(x) = J_\theta(x, u = 0; \theta)$. We can then define a one-step-ahead D-optimal information gain.

Definition 1 (D-optimality information gain)**.** One-step-ahead D-optimality yields the following objective, which should be maximized under the quadratic constraint $\|u\|_2 \leq \gamma^2$:

$$G_{\mathrm{D}}(u) = \log\det\Big[M_t + J\big(x(u)\big)^\top J\big(x(u)\big)\Big]. \tag{6}$$

The maximization of G_{D} is not straightforward because of the determinant and because J is a nonlinear function of x. To make D-optimality tractable, we linearize $J(x)$, which amounts to linearizing f to second order in θ and x. Intuitively, since we are planning between t and $t + \mathrm{d}t$ the corresponding variation in x is small so it is reasonable to use linear approximations.

Proposition 1. Let $1 \leq k \leq d$, and let us denote $M = M_t \in \mathbb{R}^{q\times q}$, $D = \partial g_\theta^k/\partial x \in \mathbb{R}^{q\times d}$, and $B = \mathrm{d}t\,\partial f_\theta/\partial u \in \mathbb{R}^{d\times m}$, with the derivatives evaluated at $\bar{z} := z(u = 0; x_t, \theta)$. Linearization of our D-optimality objective (6) with respect to x to first order in $\mathrm{d}t$ for the k-th component of the dynamics yields the following optimization problem:

$$\begin{aligned} &\max_{u\in\mathbb{R}^m} && u^\top Q u - 2v^\top u \\ &\text{subject to} && \|u\|_2^2 = \gamma^2, \\ \text{with}\quad Q = B^\top D^\top M^{-1} DB \quad &\in \mathbb{R}^{m\times m}, && v = -B^\top D^\top M^{-1} g \quad \in \mathbb{R}^m. \end{aligned} \tag{7}$$

Problem (7) can be solved efficiently at the cost of a root search and a $m \times m$ eigenvalue decomposition [7]. Algorithm 2 summarizes the computation of the D-optimal input using our method. Note that we recover the D-optimal exploration algorithm of [7] in the case of linear dynamics.

Algorithm 1 Online neural exploration

input neural model f_θ, policy π, time horizon T, time-step $\mathrm{d}t$, learning rate η
output dynamics model f_θ
for $0 \leq t \leq T-1$ **do**
  choose $u_t = \pi_t(x_{0:t}, u_{0:t-1}; f_\theta)$
  observe $x_{t+1} = x_t + \mathrm{d}t\, f(x_t, u_t)$
  compute the loss
    $\ell_t = \|f_\theta(x_t, u_t) - (x_{t+1} - x_t)/\mathrm{d}t\|_2^2$
  update $\theta \leftarrow \theta - \eta\nabla\ell_t(\theta)$
end for

Algorithm 2 Neural D-optimal planning

inputs current state x_t, current model f_θ, Gram matrix $M_t \in \mathbb{R}^{q\times q}$, time step $\mathrm{d}t$
output control u_t
define $\bar{x} = x_t + \mathrm{d}t\, f_\theta(x_t, u = 0)$
choose $1 \leq k \leq d$ and define g as the k-th row of J
compute the derivatives $J(\bar{x})$, $g(\bar{x})$, $D(\bar{x})$ and $B(\bar{x})$
compute Q and b
optimize $u_t \in \operatorname*{argmax}_{\|u\|_2^2=\gamma^2} u^\top Q u - 2v^\top u$.
return u_t

3 Exploration of physical systems

We introduce physical systems on which we expriment our exploration methods. Some of them are inspired by classical control environments of the OpenAI Gym [17]. In order to be more realistic, we take into account friction forces, which results in stable nonlinear dynamics. When controlling a physical system, friction can change the behaviour of the system and thus must be taken into account [18]. Moreover, friction forces are typically nonlinear and do not have a theoretical model, which motivates a nonlinear system identification approach.

Physical systems Our nonlinear environments are the pendulum ($d = 2, m = 1$), the cartpole with linear friction ($d = 4, m = 1$) [19], the damped robot arm ($d = 4, m = 2$) [20] and the quadrotor with nonlinear friction ($d = 6, m = 2$) [21]. The different agents learn the dynamics with partial knowledge of f. Typically, the learner knows a priori that the dynamics is of second order, and knows some components of the forces, such the controller action. The other forces, including friction, are unknown and learned online.

Baselines and oracles We confront our D-optimal exploration with several baselines. First, a random policy returns normally distributed inputs with maximal energy $u_t \sim \mathcal{N}\big(0, (\gamma^2/m)I_d\big)$. Second, a naive criterion for our inputs is to impose regular spacing of the trajectory points in the phase space. We hence propose a policy that maximizes $G(u) = \sum_{s\leq t} \|x(u) - x_s\|_2$ by gradient descent through the model prediction $x(u)$, which we call uniform exploration. Third, we define a periodic policy that excites the eigenmode ω_0 of the systems and returns for example $u_t = \gamma\sin(\omega_0 t)$

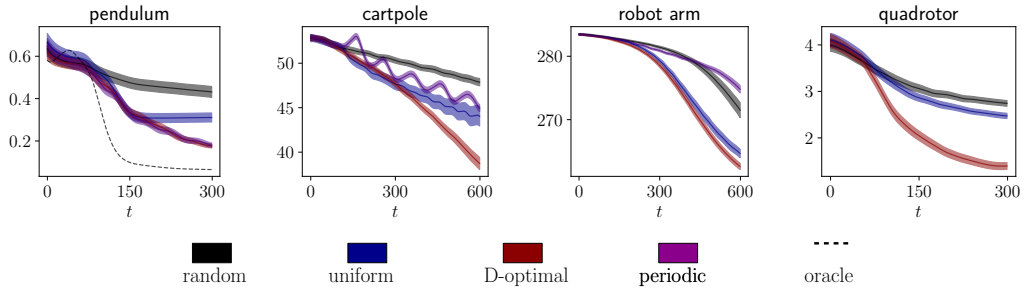

Figure 1: Evaluation loss curves for different environments against time t averaged over 100 trials.

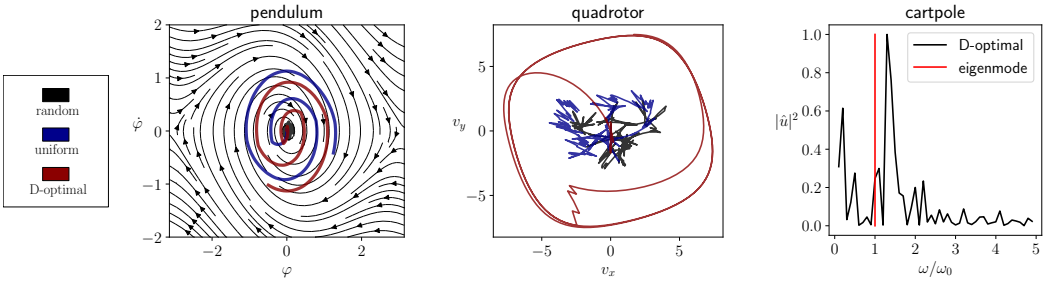

Figure 2: D-optimal sample trajectories and inputs. **Left and middle.** Phase space trajectories, for different agents. **Right.** Spectral density of the D-optimal inputs in the cartpole environment.

when $m = 1$. Near-resonance inputs yield large amplitude states in a periodic system, but require knowledge of the system's eigenmode (and hence of part of its dynamics) in advance. Finally, an oracle for the pendulum system knows the dynamics down to the physical parameters (mass, length, gravity, and friction coefficient), and hence learns them by ordinary least squares, with D-optimal inputs with respect to this parametrization.

Experimental setup For each physical system, we fix a neural architecture f_θ modelling the dynamics and run the exploration Algorithm 1 for the different policies. The performance of an agent in an environment is measured by the L^2 norm $\|f_\theta - f\|_2$ of the learned model evaluated on a grid covering typical z values. The neural nets for f_θ have two hidden layers, randomly initialized weights, a width of 16 and are trained using the Adam optimizer [22]. Our experiments are run on a laptop CPU. Our code is available at `https://github.com/MB-29/exploration`.

Results The results are shown in Figures 1 and 2. The D-optimal agent is more sample-efficient than baselines, and rivals and sometime outperforms oracles. The phase space trajectories show that the D-optimal policy yields wide and spaced states. In the cartpole environment, the model learned by the D-optimal policy results in quasi periodic inputs, with a frequency close to the pendulum's eigenmode. These results show that the D-optimal agent seeks large amplitude trajectories that are informative for the underlying neural model.

4 Conclusion and future works

We studied the problem of identifying unknown physical systems and proposed an exploration policy based on optimal design for neural networks. Our experiments show that this approach is sample-efficient and yields informative trajectories. In the future, it would be interesting to study the computational complexity of the planning algorithm, whose cost is dominated by the computation of the Jacobian and its derivative and might get big in large dimension. Keeping in mind that the purpose of exploration is to serve an exploitation objective, a natural idea would be to test our method on downstream model-based classical control tasks and compare it with other state-of-the-art approaches.

References

[1] G.C. Goodwin and R.L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Developmental Psychology Series. Academic Press, 1977.

[2] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[3] Russ Tedrake. *Underactuated Robotics*. 2022.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[5] David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

[6] David A Cohn. Neural network exploration using optimal experiment design. *Neural networks*, 9(6):1071–1083, 1996.

[7] Matthieu Blanke and Marc Lelarge. Online greedy identification of linear dynamical systems, 2022.

[8] Mona Buisson-Fenet, Friedrich Solowjow, and Sebastian Trimpe. Actively learning gaussian process dynamics. In *Learning for dynamics and control*, pages 5–15. PMLR, 2020.

[9] Matthias Schultheis, Boris Belousov, Hany Abdulsamad, and Jan Peters. Receding horizon curiosity. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 1278–1288. PMLR, 30 Oct–01 Nov 2020.

[10] Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-based active exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5779–5788. PMLR, 09–15 Jun 2019.

[11] Andrew Wagenmaker, Max Simchowitz, and Kevin Jamieson. Task-optimal exploration in linear dynamical systems, 2021.

[12] Horia Mania, Michael I. Jordan, and Benjamin Recht. Active learning for nonlinear system identification with guarantees, 2020.

[13] David Cohn. Neural network exploration using optimal experiment design. *Advances in neural information processing systems*, 6, 1993.

[14] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

[15] Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

[16] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[17] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[18] MARCO C De Simone, SERENA Russo, and ALESSANDRO Ruggiero. Influence of aerodynamics on quadrotor dynamics. *Recent Researches in Mechanical and Transportation Systems Influence*, pages 111–118, 2015.

[19] CD Green. Equations of motion for the cart and pole control task. *Sharpneat*, 2020.

[20] Joe Chen. Chaos from simplicity: an introduction to the double pendulum. 2008.

[21] Dewei Zhang, Hui Qi, Xiande Wu, Yaen Xie, and Jiangtao Xu. The quadrotor dynamic modeling and indoor target tracking control method. *Mathematical problems in engineering*, 2014, 2014.

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The algorithm is derived in Section 2 and the experiments are presented in Section 3
   (b) Did you describe the limitations of your work? [Yes] Computational limitations are discussed in Section 4.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We see no potential negative societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] The only assumption pertaining to Proposition 1 is differentiability of the model and it is verified by neural networks in general.
   (b) Did you include complete proofs of all theoretical results? [No] Proof of Proposition 1 is not provided in this workshop version, it will be provided in the full paper.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the footnote in the abstract.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The architecture we use is described in Section 3. Additional details about the environments and experiments are included in the full paper, not in this extended abstract.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Our plots have error bars.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the experimental setup in Section 3.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]