Differentiable planning for system identification

Matthieu Blanke **Marc Lelarge**

Abstract

Recent active learning algorithms for the identification of dynamical systems consist in alternating estimation and planning. The latter phase requires solving a complex control problem. In this study, we propose to address this control problem by gradient descent in an automatic differentiation framework. We provide experiments illustrating the effectiveness of our method. Adopting an experimental design perspective, we demonstrate the relevance of E-optimality for planning, hence confirming prior theoretical results.

1 Introduction

System identification is a problem of great interest in many fields such as econometrics, reinforcement learning, robotics or mechanical engineering [1, 2, 3]. The task consists in estimating the dynamics of an unknown system by sampling trajectories from it. We focus here on linear systems: let $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times m}$ be two matrices. Let us consider the linear, discrete-time, controlled dynamical system

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w_t, \qquad 0 \leq t \leq T-1, \\ x_0 &= 0, \end{aligned} \tag{1}$$

where $x_t \in \mathbb{R}^d$ is the state, $w_t \sim \mathcal{N}(0, \sigma^2)$ is a normally distributed noise and the control variables $u_t \in \mathbb{R}^m$ are chosen by the operator with the following energy constraint

$$\frac{1}{T}\sum_{t=0}^{T-1} \|u_t\|^2 \leq \gamma^2. \tag{2}$$

We further assume the condition $\rho(A) < 1$ where $\rho(A)$ is the spectral radius of A, which ensures stability and controllability of the system [4].

We denote by A_* the unknown matrix underlying the dynamics and by $\hat{A}$ its estimation from the available observations. The goal of system identification is to choose the best input sequence (u_t) to estimate A_*. [5]. In this work, we will assume that B is known, which reflects the fact that the operator knows how their action affects the system. The results for estimating A_* can be extended to the case where B is also unknown.

Adaptive learning The fundamental difficulty with this problem is that we cannot directly optimize the trajectory with respect to the control in an exact way, since A_* is unknown. A natural way to proceed consists in alternating estimation and optimization: we can iteratively update an estimate $\hat{A}$ of A_*, then optimize over the trajectories as if this estimate was the ground truth. This last operation is called planning, and the whole process is called adaptive system identification; it is summarized in Algorithm 1.

Optimal design Classically, A_* is estimated by ordinary least squares (OLS) over the observed trajectories of system (1), which has been shown to be nearly optimal for this task, in the minimax sense [5]. The objective of planning is to drive the system towards the states that are the

Preprint. Under review.

most informative in the sense of OLS. According to the theory of experimental design for a linear relation $Y = X\beta$ [6], the informativeness of a point cloud X for estimating β is captured by the eigenvalues of the information matrix $M = X^\top X$, or equivalently by the singular values $s_1 \leq \cdots \leq s_d$ of the design matrix X. In a Bayesian framework for example, the posterior distribution of β has covariance matrix M^{-1}, and the singular values of X define a so-called confidence ellipsoid. From there, the optimal design is the one that maximizes this confidence region according to some criterion: different functionals of these eigenvalues yield different criteria for optimal design [7]. We list some of these criteria in Table 1.

Contribution Planning appears to be a key problem in adaptive experimental design. We propose a novel approach to solve the planning control problem based on automatic differentiation, which inherits the flexibility of differentiable programming and is hence applicable to a wide range of settings (*e.g.* by changing the loss functional, see section 3). We provide experiments illustrating the effectiveness of our method.

2 Related work

The identification of dynamical systems is a primary field of research in control theory, and has thus been widely studied in the last decades [2, 8, 9]. Recent research streams have focused on deriving theoretical error rates for linear system estimation, most often based on least squares regression. One popular framework is that of the linear-quadratic regulator with unknown dynamics [10, 11], where one seeks to asymptotically minimize both the control cost and the estimation error. In parallel, other works aim at deriving finite-time analysis of linear system identification by bounding the estimation error produced by a regression on the trajectories. These results typically produce high probability bounds guaranteeing that the estimation is smaller than ε with probability greater than $1 - \delta$ after a certain number of samples [5, 12, 13, 14]. However, the bounds established in these works are only valid in the passive case (*i.e.* without control), where the system is excited by pure noise.

In contrast, active system identification aims to find the best control input sequence to estimate the dynamics. Recently, Wagenmaker and Jamieson [15] provided the first rigorous study of active input design for linear system identification and proposed an asymptotically optimal algorithm alternating planning and estimation. The planning phase consists in solving a complex control problem for which they mention an alternating minimization approach but do not provide implementation details. A similar theoretical analysis is conducted for a certain class of nonlinear dynamical systems in [16], together with a planning-based algorithm, but no experiments are provided.

3 Differentiable planning

Let $s_1 \leq \cdots \leq s_d$ be the ordered singular values of the design matrix $X = (x_1 | \dots | x_T)^\top$, and let $L : \mathrm{R}^d \to \mathrm{R}$ be a differentiable criterion function of the s_i. Formally, planning can be cast into the following optimal control problem :

$$\begin{aligned} &\min_{u_1,\dots,u_T} && L(s_1, \dots, s_d) \\ &\text{such that} && x_{t+1} = \hat{A}x_t + Bu_t + w_t \quad \text{and} \quad \frac{1}{T}\sum_{t=0}^{T-1} \|u_t\|^2 \leq \gamma^2. \end{aligned} \tag{3}$$

In control problem (3), we optimize a functional over a trajectory that is obtained from u_t by integrating the dynamics (1). Besides, this functional is applied after an implicit, nonlinear algebra operation of the trajectory, namely singular value decomposition. These difficulties can be tackled by the use of automatic differentiation.

Indeed, the integration of the trajectory (1) is a sequence of T elementary operations that can be differentiated through. Noise can be considered as an external quantity and does not intervene in the backpropagation step. Remarkably, the singular value decomposition is also a differentiable operation for which one can derive a closed-form adjoint equation [17], hence allowing for backpropagation through SVD. Therefore, the whole minimization program (3) can be solved in an automatic differentiation framework by the means of gradient descent. Our approach is illustrated in Figure 1.

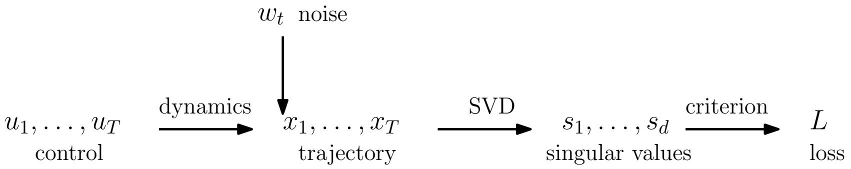


Figure 1: The training architecture for differentiable planning. All the horizontal arrows can be differentiated through automatically.

Experimental design criteria There are several options for the choice of the loss function L (see Table 1). It emerges from the theoretical results of [5], [12] and [15] that the key quantity to maximize in this setting is the minimal eigenvalue s_1, which suggests a E-optimal design.

Towards a continuous time setting Our approach can be extended to the continuous-time limit of a stochastic linear evolution $\mathrm{d}x = Ax\mathrm{d}t + Bu\mathrm{d}t + \mathrm{d}w$ in the following way. We can parametrize $u(t) = u_\theta(x_t, t)$, where u_θ is a neural network, and backpropagate the loss gradient through the differential equation. The last step can be done *e.g.* with an adjoint method, which allows for memory savings [18]. We test the validity of this neural control in the discrete time framework by differentiating through the discrete time step operations. We change the total energy constraint $\sum \|u_t\|^2 \leq T\gamma^2$ by $\|u_t\|^2 \leq \gamma^2 \; \forall t$. This allows us to readily implement the energy constraint by normalizing the output of the neural control.

```
Algorithm 1: Adaptive active system identification
Input: $n_{\text{epochs}}$, energy $\gamma^2$, horizon $T_0$, epoch ratio $r$
Result: control $u_t$
initialize $u_t$ as white noise with mean energy $\gamma^2$;
observe a first trajectory $x_t$ and append it to $X$ ;
for $1 \leq i \leq n_{\text{epochs}}$ do
    estimation : $\hat{A}$ = OLS$(X)$;
    planning : $u_t = \mathrm{Planning}(\hat{A}, T)$ ;
    play the dynamics with $u_t$, observe $x_t$ ;
    append $x_t$ to $X$;
    $T = r * T$;
end
```

Table 1: Alphabetical design criteria

Criterion	$L(s_1, \dots, s_d)$
A-optimality	$1/{s_1}^2 + \cdots + 1/{s_d}^2$
E-optimality	$-s_1$
D-optimality	$-s_1 \times \cdots \times s_d$
T-optimality	$-({s_1}^2 + \cdots + {s_d}^2)$

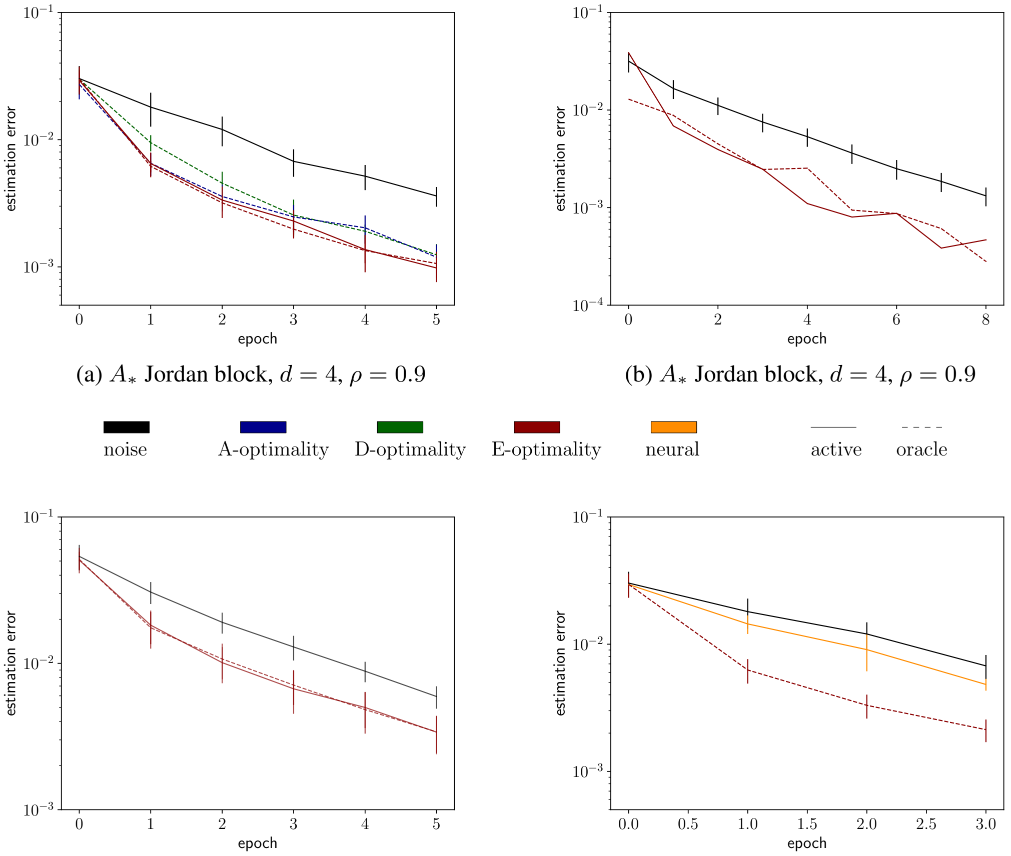
4 Experimental results

We implement differentiable planning by performing gradient descent on objective (3). We optimize a matrix of non-normalized control vectors $U = (u_1 | \dots | u_T)$, which we rescale at each planning step according to (2). We then implement the whole adaptive identification Algorithm 1 in this automatic differentiation framework and compare it to an oracle: the oracle performs planning according to the true dynamics matrix A_*, and solves (3), with $\hat{A} = A_*$. We also confront our results to a baseline consisting in normally distributed controls of mean energy γ^2: $u_t \sim \mathcal{N}(0, \gamma^2 I/d)$, as in [15]. Our code is available at `https://anonymous.4open.science/r/differentiable-planning/`.

Experimental setup We test our method on two types of unknown matrices A_*: a Jordan block and Gaussian matrices with a spectral radius normalization. We restrict ourselves to $B = I$ and $d = 4$. We set $\sigma = 0.1$, $\gamma = 1$, $r = 2$ and $T_0 = 100$. We use Pytorch [19], which implements the backward adjoint function for SVD. We perform gradient descent with the Adam optimizer [20] with a learning rate of 0.005. We set a constant number of gradient steps per epoch equal to 250. Note that we can trade accuracy for speed by changing this number. We also test the performance of the neural control. We take for u_θ a 2-layer fully connected architecture, and train it with the E-optimality criterion.

Results Our results are presented in Figure 2. Overall, the active learning algorithm outperforms the baseline and its performance matches that of the oracle algorithm, in line with the works of

Wagenmaker *et al.* [15]. Figure 2a compares different optimality criteria, and suggests that the most performant is E-optimality. Figure 2b shows single realizations of an E-optimal agent on a longer time horizon $T = T_0 \times 2^8 = 25600$. The reached accuracy is of the same order of magnitude as that of [15] for the same setting. Figure 2c represents the performance of the algorithm for randomly generated matrices at each run. Figure 2d shows the performance of the neural control in this setting. The limited results of the neural control are expected because its structure is not adapted to the discrete time setting. We ran the simulations on a laptop CPU. One run took approximately 5 hours for 8 epochs.



(a) A_* Jordan block, $d = 4$, $\rho = 0.9$

(b) A_* Jordan block, $d = 4$, $\rho = 0.9$

(c) Random A_*, $d = 4$, $\rho = 1$

(d) A_* Jordan block, $d = 4$, $\rho = 0.9$

Figure 2: Estimation error for different types of A.

5 Conclusion

We have proposed a gradient-based numerical approach for optimal design in dynamical system identification. Our results prove its efficiency on different types of matrices, and confirm the relevance of E-optimal design in this setting. Combined with an adaptive algorithm, it can therefore be used to solve the system identification control problem. We also discussed the possibility of extending our approach to a continuous-time framework with a neural control, which could be trained with neural differential equations for improved efficiency. Although the algorithm we consider in this study is relatively simple when compared to those of the related literature, we did not study theoretical bounds for the estimation of A_*. This could be the object of future works.

References

[1] H.G. Natke. System identification: Torsten söderström and petre stoica. *Automatica*, 28(5):1069–1071, 1992.

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

[3] Thomas M. Moerland, Joost Broekens, and Catholijn M. Jonker. Model-based reinforcement learning: A survey, 2021.

[4] Donald E Kirk. *Optimal control theory: an introduction*. Springer, 1970.

[5] Max Simchowitz, Horia Mania, Stephen Tu, Michael I. Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification, 2018.

[6] Olaf Wolkenhauer, Peter Wellstead, Kwang-Hyun Cho, Julio R Banga, and Eva Balsa-Canto. Parameter estimation and optimal experimental design. *Essays in biochemistry*, 45:195–210, 2008.

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

[8] Michel Gevers, Xavier Bombois, Roland Hildebrand, and Gabriel Solari. Optimal Experiment Design for Open and Closed-loop System Identification. *Communications in Information and Systems (CIS)*, 11(3):197–224, 2011. Special Issue Dedicated to Brian Anderson on the Occasion of His 70th Birthday: Part II.

[9] H. Jansson and H. Hjalmarsson. Input design via lmis admitting frequency-wise model specifications in confidence regions. *IEEE Transactions on Automatic Control*, 50(10):1534–1549, 2005.

[10] Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In Sham M. Kakade and Ulrike von Luxburg, editors, *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pages 1–26, Budapest, Hungary, 09–11 Jun 2011. PMLR.

[11] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. Regret bounds for robust adaptive control of the linear quadratic regulator, 2018.

[12] Yassir Jedra and Alexandre Proutiere. Finite-time identification of stable linear systems: Optimality of the least-squares estimator, 2020.

[13] Tuhin Sarkar and Alexander Rakhlin. Near optimal finite time identification of arbitrary linear dynamical systems, 2019.

[14] Anastasios Tsiamis and George J. Pappas. Finite sample analysis of stochastic system identification, 2019.

[15] Andrew Wagenmaker and Kevin Jamieson. Active learning for identification of linear dynamical systems, 2020.

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

[17] Mike B. Giles. Collected matrix derivative results for forward and reverse mode algorithmic differentiation. In Christian H. Bischof, H. Martin Bücker, Paul Hovland, Uwe Naumann, and Jean Utke, editors, *Advances in Automatic Differentiation*, pages 35–44, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[18] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019.

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[20] 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 claims in the abstract refer to the method exposed in section 3 and to the experimental results of section 4.
   (b) Did you describe the limitations of your work? [Yes] We specified that our work is mostly experimental and that a theoretical analysis is yet to be conducted.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We did not see any 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? [N/A] We did not include theoretical results.
   (b) Did you include complete proofs of all theoretical results? [N/A] *Idem.*
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] A URL to our GitHub repository is provided.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The hyperparameters are specified in section 4, scripts are provided in the repository.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We provided error bars with respect to random seeds.
   (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] We specified our resources and the computing time in section 4.
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? [Yes] We provided a citation for Pytorch.
   (b) Did you mention the license of the assets? [No] We used the recommended citation format for Pytorch.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We provided a link to a code repository.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We didn't use personal data.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] *Idem.*
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] We neither used crowdsourcing nor conducted research with human subjects
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] *Idem.*
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] *Idem*