Greedy online identification of linear dynamical systems

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

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

Abstract

This work addresses the problem of exploration in an unknown environment. For linear dynamical systems, we use an experimental design framework and introduce an online greedy policy where the control maximizes the information of the next step. In a setting with a limited number of observations, our algorithm has low complexity and shows experimentally competitive performances compared to more elaborate gradient-based methods. [1]

1. Introduction

System identification is a problem of great interest in many fields such as econometrics, robotics, aeronautics, or reinforcement learning (Ljung (1998); Natke (1992); Goodwin and Payne (1977); Gupta et al. (1976); Moerland et al. (2021)). The task consists in estimating the parameters of an unknown system by sampling trajectories from it as fast as possible. To this end, inputs must be chosen so as to yield maximally informative trajectories. We focus on linear time-invariant (LTI) systems. Let $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times m}$ be two matrices; we consider the following discrete-time dynamics:

$$x_{t+1} = Ax_t + Bu_t + w_t, \quad 0 \leq t \leq T-1; \qquad x_0 = 0 \tag{1}$$

where $x_t \in \mathbb{R}^d$ is the state, $w_t \sim \mathcal{N}(0, \sigma^2 I_d)$ 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 following power constraint:

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

The system parameters $(A\, B) := \theta \in \mathbb{R}^{d\times q}$ $(q = d+m)$ are unknown initially and are to be estimated from observed trajectories (x_t). The goal of system identification is to choose the inputs u_t so as to drive the system towards the most informative states for the estimation of θ. An example of such a system could be the motion of an aircraft (see Section 4). It may happen that B is known by the controller, in which case $\theta = A$, $q = m$ and our results can be readily adapted.

Motivation In practice, systems have complex dynamics and can only be approximated by linear systems on short time scales, to ensure that the dynamics remains linear and time-invariant. Moreover, long experiments can be costly (think for instance of an aircraft test flight). In order to be practical, our identification algorithm needs to interact as little as

[1] Our code is available at `https://github.com/MB-29/greedy-identification`

possible with the system and to take decisions as fast as possible so that it can run online. Hence we are interested in the regime where T is small, and attach great importance to the computational time of the identification.

Related work System identification has been widely studied in the field of optimal design of experiments (Fedorov et al. (1972); Pukelsheim (2006)). For LTI dynamic systems, classical optimal design approaches provided results for single-input single-output (SISO) systems (Goodwin and Payne (1977); Keviczky (1975); Walter et al. (1997)) or for multi-input multi-output (MIMO) systems in the frequency domain or with randomized time-domain inputs (Mehra (1976)). More recently, system identification received considerable attention in the machine learning community, with the aim of obtaining finite-time bounds on the estimation error (Jedra and Proutiere (2020a,b); Simchowitz et al. (2018)). In Wagenmaker and Jamieson (2020) and Wagenmaker et al. (2021), the inputs are optimized in the frequency domain to maximize an optimal design objective, with theoretical estimation rate guarantees in the large T limit. In our approach, we directly optimize deterministic inputs in the time domain for MIMO LTI systems.

Contributions In this work, we explore a setting for linear system identification with hard constraints on the number of interactions with the system and on the computing resources used for planning and estimation. To the best of our knowledge, finite-time system identification guarantees are only available in the large T limit which makes the hypothesis of linear time-invariant dynamics quite unlikely. Using a framework based on experimental design, we propose a greedy online algorithm requiring minimal computing resources. The resulting policy gives a control that maximizes the amount of information collected at the next step. We show empirically that for short interactions with the system, this simple approach is sample-efficient and can actually outperform more sophisticated gradient-based methods. We compare the algorithms against each other and against an oracle, both on average and on real-life dynamic systems.

2. Background

Notations In the rest of this work, we note $\theta_\star = (A_\star \, B_\star)$ the unknown parameter underlying the dynamics. We define a policy $\pi : (x_{1:t}, u_{0:t-1}) \to u_t$ as a mapping from the past trajectory to the future input. The set of policies meeting the power constraint (2) is noted Π_γ. We note $\tau = (x_{1:T}, u_{0:T-1})$ a trajectory, and we extend this notation to $\tau(\pi, T)$ when the trajectory is obtained using a policy π up to time T. We denote by $\mathbb{E}_\theta$ the average for a dynamical system given by (1), where the randomness comes from the noise w_t and possibly from the policy inducing the control u_t.

2.1 Adaptive identification

Fix an estimator $\hat{\theta} : \tau \mapsto \hat{\theta}(\tau) \in \mathbb{R}^{d\times q}$, yielding an estimate of the parameters from a given trajectory. Our objective is to find a policy $\pi \in \Pi_\gamma$ yielding trajectories for which the estimate $\hat{\theta}(\tau)$ is close to $\theta_\star$. We measure this performance by the mean squared error:

$$\mathrm{MSE}(\pi) = \frac{1}{2}\mathbb{E}_{\theta_\star}\left[\left\|\hat{\theta}\big(\tau(\pi, T)\big) - \theta_\star\right\|_{\mathrm{F}}^2\right]. \tag{3}$$

Of course, this quantity depends on $\theta_\star$ the true parameter of the system, which is unknown. A natural way of identifying $\theta_\star$ is to proceed sequentially, as follows.

Definition 1 (Adaptive system identification) *Given an estimate $\hat{\theta}_i$ of $\theta_\star$, the policy for the next sequence of inputs can be chosen so as to minimize a cost function F approximating the MSE (3), using $\hat{\theta}_i$ as an approximation of $\theta_\star$. Then, these inputs are played and $\theta_\star$ is re-estimated with the resulting trajectory, and so on. We call planning the process of minimizing F.*

This approach is summarized in Algorithm 1, which takes as inputs a first guess for the parameters to estimate θ_0 and a policy π_0, the problem parameters σ and γ, a schedule $\{t_0, = 0, t_1, \dots, t_{n-1}, t_n = T\}$, a cost functional F and an estimator $\hat{\theta}$. An adaptive identification algorithm is hence determined by a triplet $(\hat{\theta}, F, \{t_i\})$. A natural estimator in this linear problem is the least squares estimator which we define in Section 2.2 and which we adopt in the rest of this work.

Example 1 (Random policy) *A naive strategy for system identification consists in playing random inputs with maximal energy at each time step. This corresponds to the choice $t_i = i$ and π_i returning $u_t \sim \mathcal{N}(0, \frac{\gamma^2}{m} I_m)$.*

Example 2 (Oracle) *An oracle is a controller who is assumed to choose its policy with the knowledge of the true parameter $\theta_\star$. It can hence optimize $F(\pi; \theta, T) = \mathrm{MSE}(\pi)$ offline over $\{t_i\} = \{0, T\}$. By definition, the inputs played by the oracle are the optimal inputs for our problem of mean squared error system identification.*

The two previous examples correspond to the extreme cases where the controller has either no or full knowledge of the system and can be seen as the starting point and the objective of our problem respectively. Adaptive identification aims to gradually use the knowledge from collected data to control the system towards optimally informative states.

2.2 Optimal design of experiments

By the linear structure of our problem, we have both a natural estimator for the matrix $\theta_\star$ and a cost functional for planning from the theory of optimal experiment design.

Definition 2 (Ordinary least squares estimator) *The ordinary least squares (OLS) estimator associated to the trajectory $\tau = (x_{1:T}, u_{0:T-1})$ is given by*

$$\hat{\theta}(\tau)^\top = M_{T-1}^{-1} \sum_{t=0}^{T-1} z_t x_{t+1}^\top, \quad \text{with} \quad z_t = \begin{pmatrix} x_t \\ u_t \end{pmatrix} \quad \text{and} \quad M_t = \sum_{s=0}^{t} z_s z_s^\top. \tag{4}$$

We call the z_t the covariates and M_t the Gram matrix at time t.

The theory of optimal design provides an information-theoretical criterion on the covariates for the accuracy of least squares which takes the form of a concave function of the Gram matrix (Pukelsheim (2006)). We can work out a derivation for our dynamic setting (see Appendix C, Goodwin and Payne (1977) and Theorem 2.1 of Wagenmaker et al. (2021)).

Definition 3 (Optimal design functional) *Let $\Phi(M) = -\mathrm{tr}(M^{-1})$ (A-optimality) or $\Phi(M) = \log\det M$ (D-optimality). Then the associated optimal design cost functional is defined as*

$$F_\Phi(\pi; \theta, t) = -\Phi\left(\mathbb{E}\left[M_t\right]\right) \tag{5}$$

We note that M_t is quadratic in the u_s so maximizing (5) efficiently is challenging even with concavity assumptions on Φ. Recent approaches proposed gradient-based optimization of (5) in the frequency domain, over exponentially large epochs (Wagenmaker et al. (2021)). In the next section, we propose an algorithm where planning is performed online.

3. Online greedy identification

A simple, natural approach for system identification consists in updating the policy at each time-step t in a greedy fashion: the input u_t is chosen with energy γ^2 so as to maximize a one-step-ahead objective. Then, a new observation x_t is collected and the process repeats. In the formalism of Section 2.1 corresponds to the schedule $t_i = i$.

3.1 One-step-ahead objective

Following Section 2.2, we adopt the optimal design functional $F = F_\Phi$. At time t, the u_t-dependent Gram matrix is $M_{t-1} + \mathbb{E}_{\theta_t}[z_t {z_t}^\top]$. Therefore, one-step-ahead planning yields the following optimization problem:

$$\begin{aligned} \max_{u \in \mathbb{R}^m} \quad & \Phi\left(M_{t-1} + z(u)z(u)^\top\right) \\ \text{such that} \quad & z(u) = \begin{pmatrix} x_t \\ u \end{pmatrix} \quad \text{and} \quad \|u\|^2 = \gamma^2. \end{aligned} \tag{6}$$

The corresponding online identification algorithm is detailed in Algorithm 2. As we will see in Section 3.2, problem (6) can be solved accurately and at a cheap cost. Moreover, Algorithm 2 offers the advantage of improving the knowledge of $\theta_\star$ at each time-step, so that the planning objective is constantly corrected. This way, the bias introduced by the uncertainty about $\theta_\star$ is minimized, whereas a large bias could impair the identification of the system when planning is performed over larger time sequences.

Algorithm 1 Adaptive identification

```
inputs initial guess $\theta_0$, $\pi_0$, noise variance $\sigma^2$, power $\gamma^2$, cost functional $F$, estimator $\hat{\theta}$
output final estimate $\theta_T$
for $0 \leq i \leq n-1$ do
    run the true system $t_{i+1} - t_i$ steps
      with inputs $u_t = \pi_i(x_{1:t}, u_{1:t-1})$
    $\theta_i = \hat{\theta}(x_{1:t_i}, u_{1:t_i - 1})$        ▷ estimation
    $\pi_i$ solves $\min_{\pi \in \Pi_\gamma} F(\pi; \theta_i, t_{i+1})$ ▷ planning
end for
```

Algorithm 2 Greedy identification

```
inputs initial guess $\theta_0$, noise variance $\sigma^2$, power $\gamma^2$, time horizon $T$
output final estimate $\theta_T$
for $0 \leq t \leq T-1$ do
    $u_t \in \operatorname{argmax}_{\|u\|_2^2 = \gamma^2} \Phi\big(M_{t-1} + z(u)z(u)^\top\big)$
    play $u_t$, observe $x_{t+1}$
    $M_{t+1} = M_t + z_{t+1}{z_{t+1}}^\top$
    ${\theta_{t+1}}^\top = M_{t+1}^{-1}\big(M_t \theta_t + z_t {z_t}^\top\big)$
end for
```

3.2 Solving the one-step D-optimal design problem

We show that the one-step-ahead planning for online system identification is equivalent to an optimization program which can be solved efficiently.

Proposition 4 *For D-optimality, there exists a symmetric matrix $Q \in \mathbb{R}^{m \times m}$ and $b \in \mathbb{R}^m$ such that the problem* (6) *is equivalent to*

$$\min_{\|u\|_2^2 = \gamma^2} \quad u^\top Q u - 2b^\top u \tag{7}$$

The coefficients Q and b can easily be obtained in terms of M_{t-1} and θ_t (see Appendix B). Furthermore, the minimizers of Problem (7) can be characterized in the following way.

Proposition 5 *Note $\{\alpha_i\}$ the eigenvalues of Q, and u_i and b_i the coordinates of u_* and b in a corresponding orthonormal basis. Then a minimizer u_* of norm γ satisfies the following equations for some nonzero scalar μ:*

$$u_i = b_i/(\alpha_i + \mu) \quad \textit{and} \quad \sum_i \frac{{b_i}^2}{(\alpha_i + \mu)^2} = \gamma^2. \tag{8}$$

Our greedy planning problem (7) can hence be solved efficiently at the cost of a scalar root-finding search and an eigenvalue decomposition. Hager (2001) provides bounds that allow for an efficient initialization of the root-finding search.

4. Performance study

We compare our greedy algorithm to the TOPLE algorithm of Wagenmaker et al. (2021). We also implement a gradient-based planning algorithm where we directly optimize the inputs in the time domain, and refer to it as "gradient" (see Appendix D.2 for details). As the gradient-based approaches trade speed for accuracy when the number of gradient steps increases, an important performance factor is the computational time.

Performance The performance of a policy π is measured by the average estimation error over the experiments: $\varepsilon = \mathrm{MSE}(\pi)$. The two resources that may be limited are the number of observations T and the computational cost C. Since all algorithms have linear time complexity, we also introduce the computational rate $c = C/T$ so that $\varepsilon = \varepsilon(c, T)$. In practice, we find that $c_{\mathrm{greedy}} \ll c_{\mathrm{gradient}}$, where c_{gradient} is the computational rate needed for the gradient descent to converge.

4.1 Computational complexity

In order to compare the performances of gradient and greedy as a function of the computation resource, we analyze them in a (T, C) diagram.

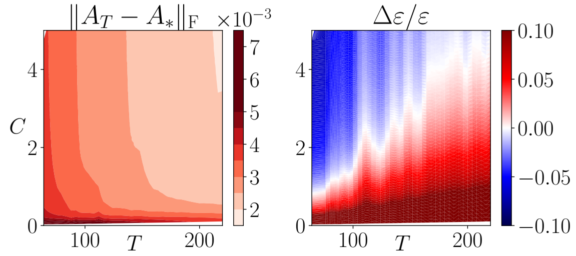
Experiment We build an experimental diagram where we plot the average estimation error for $\theta_\star = A_\star$ with known $B_\star$, as a function of T and C for the gradient algorithm. Increasing C allows for more gradient steps. We run trials with random matrices $A_\star$ of size $d = 4$, with $B = I_d$. We set $\gamma = 1$, $\sigma = 10^{-2}$, $T \in [60, 220]$. The gradient algorithm optimizes the A-optimality functional (5) with $\{t_i\} = \{0, 10, T/2, T\}$. The obtained performances are compared with those of the greedy algorithm, which has a fixed, small computational rate c_{greedy}. Our diagrams are plotted on Fig. 1a.

Results Our diagrams show that the greedy algorithm is preferable in a phase of low computational rate: $C < c_0 \times T$. The phase-separating rate c_0 corresponds to a relatively high number of gradient steps: the iso-performance along this line are almost vertical, meaning that the gradient descent has almost converged. Furthermore, the maximum performance relative gain of the gradient algorithm is of 10%. This suggests that the greedy approach is competitive and that a gradient approach comes with a large cost for a moderate improvement.

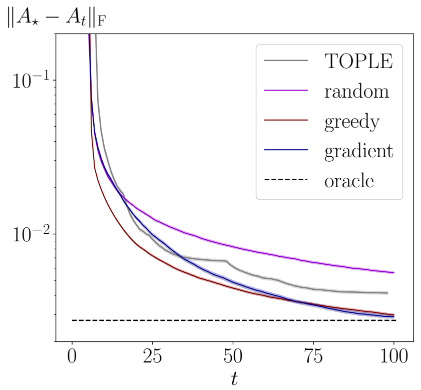
4.2 Average performance

We now test the performances of our algorithms on random matrices, with the same settings as in the previous experiment.

Experiment For each matrix $A_\star$, we test our greedy approach against the gradient-based algorithm. We also run an oracle optimal control (see Appendix D.2), and a random input



(a) Experimental (T, C) diagram. **Left.** Performance of the gradient algorithm, with varying T and C (varying number of gradient steps). **Right.** Relative performance of the gradient algorithm with respect to the greedy algorithm: positive means that greedy performs better.



(b) Identification error for random $A_\star$ averaged over 1000 samples.

Table 1: Performances of different algorithms.

	Random	**TOPLE**	**Gradient**	**Greedy**
Average c	1	~ 100	~ 50	2.36
Aircraft error	1.1×10^{-1}	8.6×10^{-2}	8.3×10^{-2}	8.2×10^{-2}
Aircraft computation time	1	55.7	25	1.8

baseline (Example 1). We measure the average running time over the trials, from which we deduce c.

Results The performances over time are plotted on Figure 1b. Both the gradient algorithm and the greedy algorithm closely approach the oracle. However, the computational cost of the gradient algorithm is far larger, as shown in Table 1 (first row).

4.3 Identification of an aircraft system

Experiment We now study a more realistic system from the field of aeronautics: the dynamics of the lateral motion of a Lockheed Jet star. We use the numerical values issued in a report from the NASA (Gupta et al. (1976)), which are summarized in Appendix E. In this setting, the number of observations is limited: $T = 150$. We apply the algorithms under study to this LTI system. Our results are summarized in Table 1 (second and third rows).

Results The greedy algorithm outperforms the gradient-based ones, both in accuracy and in compute. With small T, the estimate of $A_\star$ is too inaccurate for long-term planning to be efficient. It is more effective to update the estimate and the policy frequently. We obtain similar results for the longitudinal system of a C-8 Buffalo aircraft (Gupta et al. (1976)).

5. Conclusion

In this work, we explore a setting for linear system identification with hard constraints on the number of interactions with the real system and on the computing resources used. We introduce a fast, greedy online algorithm and show empirically that it can actually outperform more sophisticated gradient-based methods in this setting. Interesting directions of future research include finding a stopping time condition and extending this approach to the optimal control of an unknown LQR system.

References

Stephen Boyd and Lieven Vandenberghe. *Introduction to applied linear algebra: vectors, matrices, and least squares*. Cambridge university press, 2018.

V.V. Fedorov, V.V. Fedorov, W.J. Studden, E.M. Klimko, and Academic Press (Londyn). *Theory of Optimal Experiments*. Cellular Neurobiology. Academic Press, 1972. ISBN 9780122507502. URL `https://books.google.fr/books?id=v6vTAvqGny4C`.

G.C. Goodwin and R.L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Developmental Psychology Series. Academic Press, 1977. ISBN 9780122897504. URL `https://books.google.fr/books?id=tu5QAAAAMAAJ`.

NK Gupta, RK Mehra, and WE Hall Jr. Application of optimal input synthesis to aircraft parameter identification, 1976.

William W Hager. Minimizing a quadratic over a sphere. *SIAM Journal on Optimization*, 12 (1):188–208, 2001.

Yassir Jedra and Alexandre Proutiere. Finite-time identification of stable linear systems optimality of the least-squares estimator. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 996–1001. IEEE, 2020a.

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

L. Keviczky. "design of experiments" for the identification of linear dynamic systems. *Technometrics*, 17(3):303–308, 1975. ISSN 00401706. URL `http://www.jstor.org/stable/1268065`.

Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.

Raman K. Mehra. Synthesis of optimal inputs for multiinput-multioutput (mimo) systems with process noise part i: Frequenc y-domain synthesis part ii: Time-domain synthesis. In Raman K. Mehra and Dimitri G. Lainiotis, editors, *System Identification Advances and Case Studies*, volume 126 of *Mathematics in Science and Engineering*, pages 211–249. Elsevier, 1976. doi: https://doi.org/10.1016/S0076-5392(08)60873-5. URL `https://www.sciencedirect.com/science/article/pii/S0076539208608735`.

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

H.G. Natke. System identification: Torsten söderström and petre stoica. *Automatica*, 28(5): 1069–1071, 1992. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(92)90167-E. URL `https://www.sciencedirect.com/science/article/pii/000510989290167E`.

Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

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

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

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

Eric Walter, Luc Pronzato, and John Norton. *Identification of parametric models from experimental data*, volume 1. Springer, 1997.

Appendix A. Notations

The data-generating distribution knowing the parameter θ can be computed using the probability chain rule with the dynamics (1):

$$p(\tau|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}\sum_{t=0}^{T-1} \|x_{t+1} - Ax_t - Bu_t\|_2^2\right]. \tag{9}$$

We define the log-likelihood (up to a constant):

$$\begin{aligned} \ell(\tau, \theta) &= -\frac{1}{2\sigma^2}\sum_{t=0}^{T-1} \|x_{t+1} - Ax_t - Bu_t\|_2^2 \\ &= -\frac{1}{2\sigma^2}\|Y - Z\theta^\top\|_{\mathrm{F}}^2, \end{aligned} \tag{10}$$

where we have noted $Y = (y_0 \dots y_{T-1})^\top \in \mathbb{R}^{T\times d}$ and $Z = (z_0 \dots z_{T-1}{}^\top) \in \mathbb{R}^{T\times q}$ the observations and the covariates associated to the parameter θ. If $\theta = (A\, B)$, then $y_t = x_{t+1}$, $z_t = \begin{pmatrix} x_t \\ u_t \end{pmatrix}$. If $\theta = A$, then $y_t = x_{t+1} - Bu_t$ and $z_t = x_t$. We also note $U = (u_0 \dots u_{T-1}{}^\top) \in \mathbb{R}^{T\times m}$ the input matrix and $X = (x_0 \dots x_{T-1}{}^\top) \in \mathbb{R}^{T\times d}$ the state matrix. Note that $Z^\top Z = M_T$.

Appendix B. Proofs

B.1 Proof of Definition 2

The least squares estimator minimizes the quadratic loss

$$\frac{1}{2}\sum_{t=0}^{T-1} \|x_{t+1} - Ax_t - Bu_t\|_2^2, \tag{11}$$

which writes

$$\frac{1}{2}\left\|Y - Z\theta^\top\right\|_{\mathrm{F}}^2 = \frac{1}{2}\sum_{j=1}^{d} \|Y_j - Z\theta_j\|_2^2 \tag{12}$$

with Y_j the j-th column of Y and θ_j the j-th row of θ. The d terms of the sum can be minimized independently, with each θ_j minimizing the least squares of the vectorial relation $Y_j = Z\beta$. The solution for θ_j is equal to $\hat{\theta}_j = (Z^\top Z)^{-1}Z^\top Y_j$ (see *e.g.* Boyd and Vandenberghe (2018)). By concatenating the columns, we obtain that $\hat{\theta}^\top = (Z^\top Z)^{-1}Z^\top Y$, which proves (4). Note here that a controllability assumption on $(A_\star, B_\star)$ is sufficient to ensure that Z can be made full rank, and hence that the moment matrix $Z^\top Z$ is invertible.

B.2 Proof of Proposition 4

By the matrix determinant lemma, we find that

$$\begin{aligned} \log\det\left(M_{t-1} + z(u)z(u)^\top\right) &= \log\det M_{t-1} \\ &\quad + \log\left(1 + z(u)^\top M_{t-1}^{-1} z(u)\right). \end{aligned} \tag{13}$$

Maximizing this quantities with respect to u amounts to maximizing $z(u)^\top M_{t-1}^{-1} z(u)$. The matrix M_{t-1}^{-1} is symmetric because M_{t-1} is symmetric, and so are its diagonal submatrices. Given the affine dependence of z in u and the (possible) block structure of z and M_{t-1}, $z(u)^\top M_{t-1}^{-1} z(u)$ is of the form $u^\top Q u - 2b^\top u$, up to a constant. Matrices Q and b are obtained with the sub-matrices of M_{t-1}.

Partially known dynamics If $B_\star$ is known, the one-step-ahead Gram matrix at time t is M_t instead of M_{t-1}, and $z(u) = A_t x_t + Bu$. This leads to

$$Q = -B^\top {M_t}^{-1} B, \quad b = B^\top {M_t}^{-1} A_t x_t. \tag{14}$$

B.3 Proof of Propostion 5

By the Lagrange multiplier theorem there exists a nonzero scalar μ such that $Qu_* - b = -\mu\, u_*$, where μ can be scaled such that $Q + \mu I_m$ is nonsingular. Inverting the optimal condition and expanding the equality constraint gives the two conditions.

Appendix C. Optimal design of experiments

In the theory of optimal design, the informativeness of an experiment is measured by the size of the expected Fisher information.

Definition 6 (Fisher information matrix) *Let $\ell(\tau, \theta) = \log p(\tau|\theta)$ denote the log-likelihood of the data-generating distribution knowing the parameter θ. The Fisher information matrix is defined as*

$$I(\theta) = -\mathbb{E}_\theta \left[\frac{\partial^2 \ell(\tau, \theta)}{\partial \theta^2} \right] \quad \in \mathbb{R}^{qd \times qd}. \tag{15}$$

Proposition 7 *For the LTI system* (1),

$$I(\theta) = \frac{T}{\sigma^2} \mathrm{diag}(\Gamma_T, \dots, \Gamma_T), \tag{16}$$

the number of blocks being d, and with

$$\Gamma_t = \frac{1}{t} \mathbb{E}_\theta [M_{t-1}]. \tag{17}$$

Proof The log-likelihood (10) can be separated into a sum over the θ_j as in (12). The quadratic term in θ_j is $\|Z\theta_j\|_2^2 = {\theta_j}^\top Z^\top Z \theta_j$ and the other terms are constant or linear. Differentiating twice and taking the expectation gives $\mathbb{E}_\theta[Z^\top Z]$, which yields the desired result after dividing by $-\sigma^2$. ■

Definition 8 (Design criteria) *In the field of optimal design of experiments, the size of the information matrix is measured by some criterion $\Phi : \mathbb{S}_n^+(\mathbb{R}) \to \mathbb{R}_+$, which is a functional of its eigenvalues $\lambda_1, \dots, \lambda_d \geq 0$. The quantity $\Phi(I)$ represents the amount of information brought by the experiment and should be maximized.*

The criteria are required to have properties such as homogeneity, monotonicity and concavity in the sense of the Loewner ordering, which can be interpreted in terms of information theory: monotonicity means that a larger information matrix brings a greater amount of information, concavity means that information cannot be increased by interpolation between experiments. We refer to Pukelsheim (2006) for more details.

Table 2: Alphabetical design criteria.

Optimality	$\Phi(\lambda_1, \dots, \lambda_d)$
A-optimality	$-\big(1/\lambda_1 + \dots + 1/\lambda_d\big)$
D-optimality	$\log \lambda_1 + \dots \log \lambda_d$
E-optimality	λ_1

Appendix D. Gradient-based identification

In this section, we propose a gradient-based approach to planning. In a sequential identification scheme of Algorithm 1, the cost functions (3) and (5) can be optimized by projected gradient descent. This builds on the following remark. The gradients with respect to U can either be derived analytically (see Goodwin and Payne (1977), section 6 for the derivation of an adjoint equation) or automatically in an automatic differentiation framework. We rescale U at each step to ensure the power constraint is met. The t_i are chosen arbitrarily. The computational complexity of the algorithm is linear in T: each gradient step backpropagates through the planning time interval.

D.1 Gradient-based optimal design

We propose a gradient-based method to optimize U by performing gradient descent directly on U in functional (5). Note that we optimize the inputs directly in the time domain, whereas other approaches such as Wagenmaker et al. (2021) perform optimization in the frequency domain by restricting the control to periodic inputs.

D.2 Gradient through the oracle MSE

Given the true parameters $\theta_\star = (A_\star \, B_\star)$, the optimal control for the MSE minimizes the MSE cost (3), as explained Example 2. However, the dependency between Z and W makes this functional complicated to evaluate and to minimize with respect to the inputs, even when the true parameters $\theta_\star$ are known. We propose a numerical method to minimize (3) using automatic differentiation an Monte-Carlo sampling. Given one realization of the noise and inputs U, the gradient of the MSE can be computed automatically in an automatic differentiation framework. Then, one can sample a batch of b noise matrices $W_1, \dots, W_b \sim \mathcal{N}(0, \sigma^2 I)$ and approximate the gradient of (3) by

$$\nabla \mathrm{MSE}(U) \simeq \frac{1}{b} \sum_{i=1}^{b} \nabla_U \mathrm{tr} \left[Z(Z^\top Z)^{-2} Z^\top W_i {W_i}^\top \right] . \tag{18}$$

Although we do not have convergence guarantees due to the lack of structure of the objective function, the gradient descent does converge in practice, to a control that outperforms the adaptive controls.

Algorithm 3 Planning by projected gradient descent

```
inputs $A_t$, $\sigma$, $\gamma$, $T$, $\eta$
output control $U \in \mathbb{R}^{(T-t)\times m}$
for $0 \leq j \leq n_{\text{gradient}}$ do
    $U = U - \eta \nabla F(U)$
    $U = (\gamma\sqrt{T}/\|U\|_{\text{F}}) \times U$
end for
```

Appendix E. Experimental details

The linear dynamics for an aircraft flying at 573.7 meters/sec at 6.096 meters are given by the following matrix, obtained after discretization and normalization of the continuous-time system described in Gupta et al. (1976):

$$A_\star = \begin{pmatrix} .955 & -.0113 & 0 & -.0284 \\ 0 & 1 & .0568 & 0 \\ -.25 & 0 & -.963 & .00496 \\ .168 & 0 & -.00476 & -.993 \end{pmatrix}, \quad B_\star = 0.1 \times \begin{pmatrix} 0 & 0.0116 \\ 0 & 0 \\ 1.62 & .789 \\ 0 & -.87 \end{pmatrix}, \tag{19}$$

and $\sigma = 1$, $\gamma \simeq 4$ deg.