Proceedings of Machine Learning Research vol 120:1–11, 2020 2nd Annual Conference on Learning for Dynamics and Control

Actively Learning Gaussian Process Dynamics

Mona Buisson-Fenet[12] BUISSONFENET@IS.MPG.DE
Friedrich Solowjow[1] SOLOWJOW@IS.MPG.DE
Sebastian Trimpe[1] TRIMPE@IS.MPG.DE
[1] *Intelligent Control Systems Group, Max Planck Institute for Intelligent Systems, Stuttgart, Germany*
[2] *Centre Automatique et Systèmes, MINES ParisTech, PSL University, Paris, France*

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

Abstract

Despite the availability of ever more data enabled through modern sensor and computer technology, it still remains an open problem to learn dynamical systems in a sample-efficient way. We propose active learning strategies that leverage information-theoretical properties arising naturally during Gaussian process regression while respecting constraints on the sampling process imposed by the system dynamics. Sample points are selected in regions with high uncertainty, leading to exploratory behavior and data-efficient training of the model. All results are validated in an extensive numerical benchmark.

Keywords: Active learning, Gaussian process regression, nonlinear system identification.

1. Introduction

Learning dynamical systems has received considerable attention over the last decades and is widely recognized as an important and difficult problem (Schön et al., 2011). Indeed, in the case of physical systems, sampling data often requires practically involved and time-consuming experiments. Further, sampling at informative locations of the state space is challenging, since the system is constrained by the underlying dynamics. This is one key difference to many machine learning tasks, where data can be collected anywhere. Hence, it is essential to excite the system in such a way that the generated data enables sample-efficient learning. In the case of linear time-invariant (LTI) systems, there exists a rich body of theoretical results for this problem (Ljung, 2001). Nonetheless, it is still an active field of research, e.g., see (Simchowitz et al., 2018) and references therein. Convergence results are usually tightly connected to the well-established theory of persistence of excitation (Green and Moore, 1986), which ensures that control inputs are significant enough to sufficiently excite the system. However, these control inputs are not necessarily optimal and targeted exploration can accelerate learning (Umenberger et al., 2019). Learning nonlinear systems is even more complex, although there have been many advances and progress over the years (see e.g., (Schön et al., 2011; Schoukens and Ljung, 2019)). We consider Gaussian process (GP) regression, which has been proven to be an efficient framework in many related applications, including model learning (Nguyen-Tuong and Peters, 2011) or reinforcement learning (Deisenroth and Rasmussen, 2011; Doerr et al., 2017). These probabilistic models have many advantageous properties for learning dynamical systems (Jain et al., 2018; Eleftheriadis et al., 2017; Doerr et al., 2018), such as taking uncertainty into account, coping with small datasets and incorporating prior knowledge.

Active learning, i.e., sequentially choosing where to sample in order to build an informative dataset, has been investigated in many domains (see (Aggarwal, 2014) for an overview). A critical

© 2020 M. Buisson-Fenet, F. Solowjow & S. Trimpe.

difference that sets the active learning problem for dynamical systems apart is the fact that it is not possible to arbitrarily sample the state-action space. Indeed, the system is constrained by the dynamics, and has to be excited appropriately by control inputs. Existing approaches for actively learning static maps thrive by incorporating information-theoretical criteria that guide the sampling procedure. In particular, the combination with GPs yields powerful theoretical and practical results (Krause et al., 2008; Krause and Guestrin, 2007). For dynamical systems, however, there is only little related work. Recent attempts have been made, proposing a greedy exploration scheme (Jain et al., 2018), focusing on exploration under safety constraints (Koller et al., 2018; Fisac et al., 2019; Heim et al., 2019), or on active exploration for reinforcement learning using linear Bayesian inference rather than GPs (Schultheis et al., 2019). Approaches relying on parametrization of the trajectory have also been presented, including for learning time series with GPs (Binney et al., 2010; Zimmer et al., 2018). The proposed algorithms are related to this work, however, the analysis differs in several important points, which we will further discuss in Section 4.

Contributions We investigate the active learning problem for dynamical systems, which are modeled by a GP. In particular, we take the learned dynamics explicitly into account to guide the exploration.The following contributions are made:

- Proposal of a method that searches for informative points to visit, then separately drives the system to reach them (*separated search and control*, short `sep`). While we can provide some theoretical guarantees on the suboptimality of the sequence of locations to visit, we find the method to have limitations in practice.
- Novel method for deriving input trajectories by maximizing an information criterion, while taking the dynamics into account as constraints. We propose two variants, *receding horizon* and *plan and apply* (short `rec` and `p&a`).
- Benchmark on a set of numerical experiments, including robotic systems from reinforcement learning benchmarks, showing the superiority of approaches based on joint optimization for actively exploring the state-action space.

2. Problem statement

We consider a system subject to the following discrete-time dynamics:

$$x_{k+1} = f(x_k, u_k) \qquad y_k = x_k + \epsilon_k, \tag{1}$$

where f is an unknown Lipschitz-continuous function, $x_k \in \mathcal{X}$ is the system state at time step $k \in \mathbb{N}$, with $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ the space of possible states, $u_k \in \mathcal{U}$ is the control input, with $\mathcal{U} \subseteq \mathbb{R}^{d_u}$ the space of bounded control inputs, and $\epsilon_k \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is i.i.d. Gaussian measurement noise. We assume the system has sufficient controllability and stability properties in order to exclude notoriously difficult learning problems, where systematic exploration would not be meaningful. For inputs $Z = (z_0, \cdots, z_n)^\intercal$, where $z_k = (x_k, u_k)$, we observe the noisy measurements $Y = (y_1, \cdots, y_{n+1})^\intercal$. We are thus in the standard GP regression setting with noise-free input and noisy target. While there are extensions to more realistic settings such as learning dynamics with noisy inputs (McHutchon and Rasmussen, 2011) and latent states (Doerr et al., 2017, 2018), we do not consider them here for simplicity, as they are orthogonal to the problem of excitation. The true system dynamics f are

approximated by a GP denoted $\hat{f}$ (see (Rasmussen and Williams, 2006) for an overview). It is fully characterized by its mean function $\mu(\cdot)$ and its covariance function $k(\cdot,\cdot)$. The prediction $\hat{f}(z_*)$ at an unobserved point z_* is normally distributed with posterior mean and variance

$$\mu(z_*) = k_*^\intercal (K + \sigma_\epsilon^2 I)^{-1} Y \qquad \sigma^2(z_*) = k_{**} - k_*^\intercal (K + \sigma_\epsilon^2 I)^{-1} k_*, \tag{2}$$

where $K = (k(z_i, z_j))_{z_i, z_j \in Z}$ is the covariance matrix of Z, $k_* = (k(z_i, z_*))_{z_i \in Z}$, and $k_{**} = k(z_*, z_*)$. The prior mean is assumed to be zero without loss of generality. The differential entropy of $\hat{f}$ at z_*, which quantifies the uncertainty of the prediction (MacKay, 2003), is defined as $H(z_*) = \frac{1}{2}\log(2\pi e \sigma^2(z_*))$. The kernel k usually depends on some hyperparameters, which are optimized during learning, often by maximizing the data marginal log likelihood.

We address the following question: how should one excite (1) to generate samples (Z, Y) for learning f in a sample-efficient way? We derive control inputs that optimize information criteria such as differential entropy, while taking the autoregressive structure of (1) explicitly into account. Ultimately, this reduces the prediction error given a fixed number of samples, by choosing informative control inputs that lead to exploratory system behavior.

3. From static to dynamic – a fundamentally different problem

Powerful active learning strategies have been developed for learning static maps (Krause et al., 2008). Here, it is possible to immediately query any point in the input space. Thus, the problem is amenable to a clean information-theoretical treatment, which is lost for dynamical systems. The insights from the well-studied static problem are a natural starting point for this work, and we present an extension to the dynamic setting herein. At the same time, we shall underline the fundamentally different nature of the dynamical problem.

3.1. The static problem: sensor placement

A canonical example for actively learning static maps is the sensor placement problem (Krause et al., 2008). The objective is to find the best locations of N sensors $X = (x_1, ..., x_N)$ out of a finite subset of possible locations, in order to approximate a static map f with a GP $\hat{f}$, using noisy measurements $y_i = f(x_i) + \epsilon_i$. A possible solution is to select $X^{\mathrm{OPT}} = \mathrm{argmax}_X\, I(X, f)$, where $I(X, f) = H(Y_X) - H(Y_X|f)$ is the mutual information between the observations Y at X and the underlying function f, and H is the differential entropy of the GP $\hat{f}$. However, finding such an optimal set of placements is NP-hard (Krause et al., 2008). Therefore, there is a need for tractable approximations. In particular, the optimal sequence of placements can be approximated by the greedy rule

$$x_i = \underset{x}{\mathrm{argmax}}\, I(x \cup X_{i-1}, f) = \underset{x}{\mathrm{argmax}}\, H_{i-1}(x) \quad \forall\, i \in \{1, ..., N\}, \tag{3}$$

with H_i the differential entropy of the GP at iteration i. Thanks to the submodularity and monotonicity of the function $I(X, f)$ (see (Krause and Guestrin, 2005; Srinivas et al., 2012) for details) and Proposition 4.3 in (Nemhauser et al., 1978), it can be shown that the sequence X^{G} of greedy placements selected by (3) is close to the true optimal sequence X^{OPT}:

$$I(X^{\mathrm{G}}, f) \geq (1 - 1/e) I(X^{\mathrm{OPT}}, f). \tag{4}$$

In this paper, we focus on differential entropy since we are considering a continuous space, for which the mutual information criterion proposed in (Krause et al., 2008) cannot easily be computed.

3.2. Extension to dynamical systems

The dynamical problem is fundamentally different: we cannot sample at an arbitrary state x, we need to steer the system to x through the unknown dynamics f with a sequence of bounded actions u. Therefore, there is also an information gain along the trajectory, which is not considered in the previously introduced static framework.

At first, we ignore this fact: we separate the search for informative states from obtaining the control inputs that drive the system to these states. This method is denoted *separated search and control* (`sep`). Starting from an initial point $z_0 := (x_0, u_0)$, at each iteration the next location to visit is determined by the greedy rule

$$z_i^{\mathrm{G}} = \underset{z \in \mathcal{X} \times \mathcal{U}}{\operatorname{argmax}} I(z \cup Z_{i-1}, f) = \underset{z \in \mathcal{X} \times \mathcal{U}}{\operatorname{argmax}} H_{i-1}(z) \quad \forall\, i \in \{1, ..., N\}, N \in \mathbb{N}. \tag{5}$$

After solving (5), we get a state-action pair $(x_i^{\mathrm{G}}, u_i^{\mathrm{G}})$. We steer the system to x_i^G using a control trajectory $(u_k, ..., u_{k+M-1}) \in \mathcal{U}^M$, then apply u_i^{G}, and update the GP and its hyperparameters with the data collected along the way. Here, M is the control horizon, and k is the time step since the beginning of the experiment, while i indexes the iterations of the greedy procedure. Due to the controllability assumption, the existence of such control trajectories is ensured for sufficiently large M and there exist methods to obtain them, e.g., iLQR (Tassa et al., 2014, 2012). However, there are severe issues in the concrete implementation:

- In general, it is difficult to choose M a priori such that each z_i^{G} is attainable in M steps. However, limiting the search space in (5) to the points attainable in M steps yields a time-varying set from which to choose z. In this case, Proposition 4.3 in (Nemhauser et al., 1978) is not applicable anymore. Hence, suboptimality guarantees of type (4) typically can only be derived for (5) if z is chosen directly from $\mathcal{X} \times \mathcal{U}$.
- With only the estimated dynamics $\hat{f}$ available for controller design, the location chosen by (5) might not actually be reached and the intended data point not be obtained.
- The considered search space $\mathcal{X} \times \mathcal{U}$ is continuous, and not a finite set of possible locations as for the sensor placement problem. This has implications on the property of submodularity, which is originally defined for set functions.
- Solving (5), a non-convex problem in a continuous state-action space, is nontrivial, and might not return the true optimum at each iteration.

One can obtain suboptimality guarantees of type (4) for (5) under restrictive assumptions, namely: $\mathcal{X} \times \mathcal{U}$ is a finite set, z_i^{G} is the true optimum at iteration i, and is actually visited by the control procedure. However, these do not hold in practice. Information gain along the trajectory is also not included in this theoretical framework. Next, we show that more efficient strategies can be designed by optimizing over the whole control trajectory.

4. Informative control generation

Separating search from control yields an insightful embedding into the sensor placement problem. However, the crucial properties of this problem also become apparent, revealing that many aspects of the current solutions are not transferrable to dynamical systems. The discussed insufficiencies

inspired us to jointly optimize for control inputs and informative states with respect to the approximate dynamics as constraints. We propose the following approach: at time step k, we pick the most informative control trajectory by solving

$$U_k^* = \underset{(u_k,\dots,u_{k+M-1})\in\mathcal{U}^M}{\operatorname{argmax}} \sum_{i=0}^{M-1} H_k(\hat{x}_{k+i}, u_{k+i}) \tag{6}$$
$$\text{s.t. } \hat{x}_{k+i+1} = \hat{f}_k(\hat{x}_{k+i}, u_{k+i}),\ u_{k+i} \in \mathcal{U},\ \forall\, i \in \{0, ..., M-1\}$$

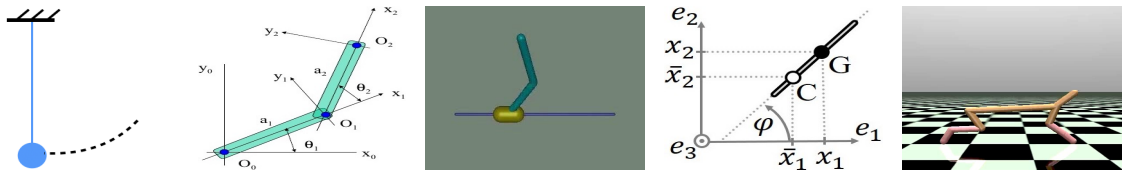
for a fixed time horizon M, $\hat{x}_k = x_k$, and $\hat{x}_{k+i+1}$ the mean of the GP prediction. This method is highly versatile: the cost function can easily be extended and further regularized, e.g., by penalizing a suitable norm of the control signals. Numerically, we use direct multiple shooting in CasADi (Andersson et al., 2018) with Ipopt (Wächter and Biegler, 2005), as in (Koller et al., 2018; Jain et al., 2018). Note that (6) is only an upper bound of the entropy accumulated over the trajectory. This bound could be made sharper, for example by approximating the propagation of uncertainty through moment matching, but this necessitates added approximations and computations.

Receding horizon or plan and apply For the above-described method, we propose two options. In the default setting, we update the GP and optimize its hyperparameters at each time step. Then, we solve (6) in a receding time fashion (variant denoted *receding horizon*, short `rec`). However, this is very costly in terms of computations and may lead to a shortsighted behavior, since the exploration strategy has a chance to "change its mind" every time step, which can lead to a greedy behavior if the optimization landscape is too flat. Thus, we propose a computationally cheaper alternative: we solve (6), roll out the whole control trajectory, batch update the GP with the measurements taken along the way, optimize its hyperparameters, and iterate (variant denoted *plan and apply*, short `p&a`). In this case, M needs to be well-chosen: if it is too large, the GP will not be updated often enough, but if it is too small, the procedure will be too shortsighted.

Related algorithms Recent works propose related algorithms, but with a different focus. For example, in (Zimmer et al., 2018), exploration under safety constraints of dynamical systems modeled by GPs is investigated. However, exploration is achieved by parametrizing and selecting an informative piecewise linear trajectory in state space, which simplifies the optimization problem. An exploration scheme for learning GP dynamics is presented in (Jain et al., 2018), but it is greedy. The model learned during exploration is then used for reference tracking under uncertainty by solving an MPC problem. In (Koller et al., 2018), an MPC scheme for safe exploration of dynamical systems is proposed, where the constraint does not directly lie on the estimated dynamics, but on the propagation of safe ellipsoids through these dynamics. Exploration for reinforcement learning is considered in (Schultheis et al., 2019), where a sequence of discrete actions is optimized. The proposed algorithm is related to (6), but using linear Bayesian inference as the learning framework. In this paper, we focus on exploration for learning dynamical systems with GPs, with a novel viewpoint extending solutions of the static problem. We also provide a comprehensive benchmark of control systems, which follows next.

5. Numerical benchmark

We compare the proposed methods in a numerical benchmark. For each approach, we evaluate the prediction error and quantify how much of the state space has been explored. The results are summarized in Table 1 and Figure 2.



(*a*) Pendulum (*b*) Two-link robot (*c*) DIPC (*d*) Unicycle (*e*) Half-cheetah

Figure 1: Illustration of the benchmark systems.

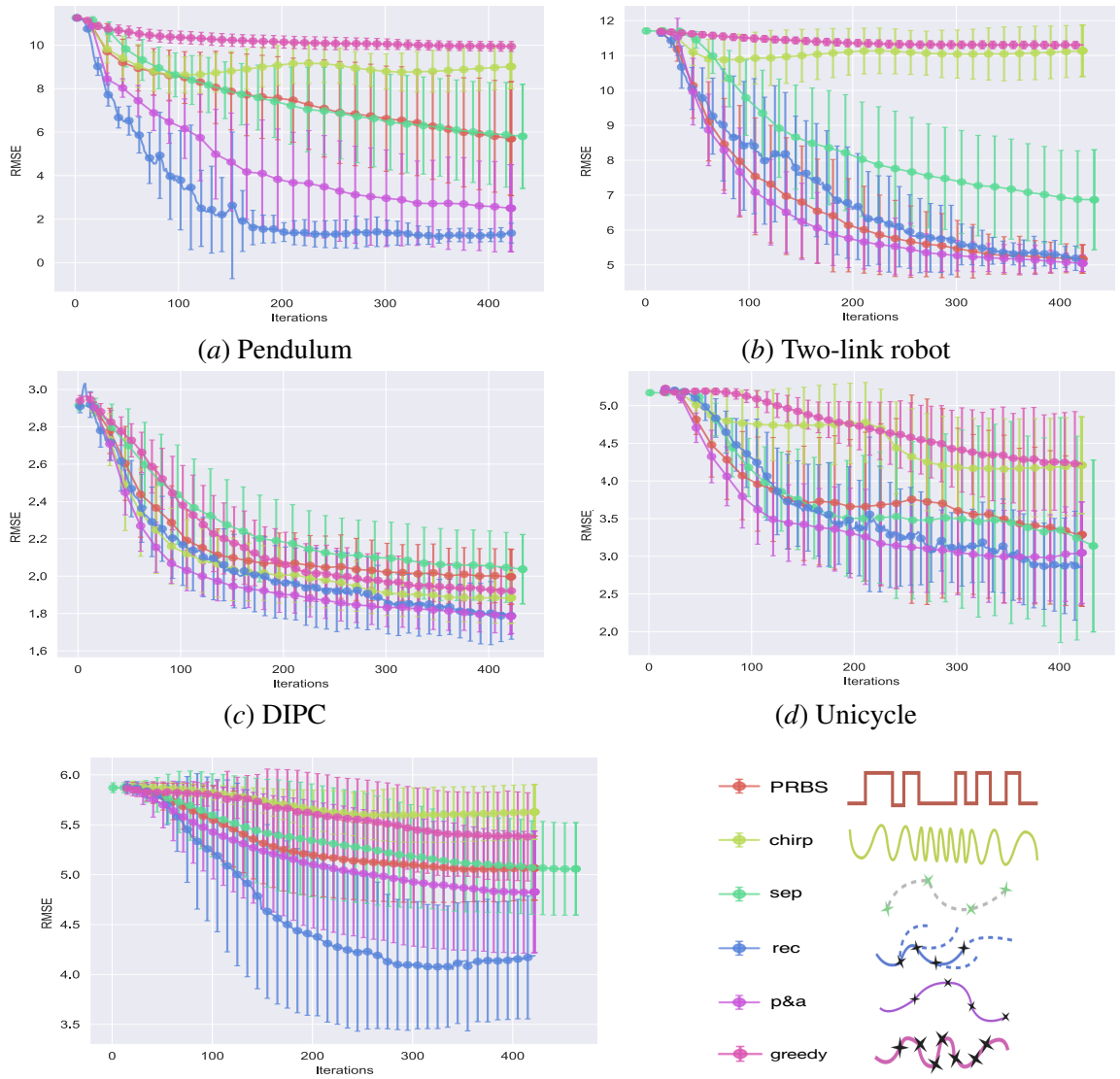
Experimental settings We run the following methods[1]: pseudorandom binary sequences (PRBS) and chirps ((Nelles, 2001), Section 17.7) to compare to standard system identification signals, separated search and control (`sep`), and our method based on optimal control (6) with either receding horizon (`rec`), plan and apply (`p&a`). We also compare with the greedy strategy ($M = 1$). Random control inputs with a holding time of either M or 1 were also investigated, but they did not challenge the other standard signals and are not included in the final plots. We focus on the standard GP setting: squared exponential kernel, independent GPs for each output dimension, hyperparameters that maximize the marginal log likelihood. We evaluate on five nonlinear dynamical systems with continuous state-action space and bounded controls, illustrated in Figure 1:

- Pendulum, with $d_x = 2$, $d_u = 1$, $\sigma_\epsilon^2 = 0.05$, $M = 15$;
- Two-link planar robot (Siciliano et al., 2009), with $d_x = 4$, $d_u = 2$, $\sigma_\epsilon^2 = 0.05$, $M = 15$;
- Double inverted pendulum on a cart (DIPC) from the MuJoCo environment (Todorov et al., 2012) in Gym (Brockman et al., 2016), with $d_x = 8$, $d_u = 1$, $\sigma_\epsilon^2 = 0.05$, $M = 15$, and added damping;
- Unicycle (Udwadia and Kalaba, 2007), with $d_x = 6$, $d_u = 2$, $\sigma_\epsilon^2 = 0.001$, $M = 15$;
- Half-cheetah, also from MuJoCo in Gym, with $d_x = 18$, $d_u = 6$, $\sigma_\epsilon^2 = 0.001$, $M = 10$.

Each method uses the same number of data points and the same planning horizon (except for the greedy method which has a horizon of 1) and starts from the same stable equilibrium. We make sure each system has enough damping to be sufficiently stable and controllable in the exploration region, and choose the bounds on $\mathcal{U}$ such that exploring the state space is neither too easy (even random signals can easily go everywhere) nor too hard (even active exploration methods cannot go far). We run 10 trials of `rec` since it is computationally heavy, and 100 trials of all other methods.

Evaluation criteria We quantify the accuracy of the learned model by monitoring the root mean square prediction error (RMSE) over a grid of uniformly randomly distributed states and inputs in a predefined region of interest, and the quality of exploration by computing the percentage of coverage of this region at the end of the experiment. The region of interest is chosen a priori in $\mathcal{X}$ for each system, by heuristically picking bounds in each dimension in which most of our experiments stay when started from the same stable equilibrium. The percentage of coverage is computed by discretizing this region and computing the average number of cells visited during the experiment.

1. Code available at https://git-amd.tuebingen.mpg.de/mbuissonfenet/active_learning_gp.git



(*a*) Pendulum (*b*) Two-link robot

(*c*) DIPC (*d*) Unicycle

(*e*) Half-cheetah

Figure 2: RMSE over time for all benchmark systems (mean $\pm$ standard deviation).

In this paper, we are not as much interested in the absolute results but rather in the comparison between the different methods, and in demonstrating that some explore the state space more than others, yielding a more accurate model.

Results The results shown in Table 1 and Figure 2 confirm that our optimization-based exploration methods (`rec` and `p&a`) yield the lowest prediction error and thus, the best models. Indeed, they can push each nonlinear system to unknown regions of $\mathcal{X} \times \mathcal{U}$, generating informative data points in the whole state space, which yields an overall more consistent model. The `sep` method performs reasonably well, but significantly worse than `rec` and `p&a`. The greedy method is not able to explore as much of the state space as it is too shortsighted. The standard signals from sys-

Table 1: Final RMSE and coverage of the state space (mean $\pm$ standard deviation).

	Pendulum	Two-link	DIPC	Unicycle	Half-cheetah
Final RMSE					
PRBS	5.7 ± 2.6	5.2 ± 0.4	2.00 ± 0.15	3.3 ± 1.0	5.1 ± 0.3
chirp	9.0 ± 1.0	11.1 ± 0.7	1.89 ± 0.14	4.2 ± 0.6	5.6 ± 0.3
sep	5.8 ± 2.4	6.9 ± 1.4	2.03 ± 0.19	3.1 ± 1.1	5.1 ± 0.5
rec	1.4 ± 0.5	5.2 ± 0.4	1.79 ± 0.12	2.9 ± 0.7	4.2 ± 0.6
p&a	2.5 ± 2.0	5.0 ± 0.2	1.79 ± 0.09	3.0 ± 0.7	4.8 ± 0.6
greedy	9.9 ± 0.2	11.3 ± 0.1	1.92 ± 0.13	4.2 ± 0.7	5.4 ± 0.4
Coverage of state space (in percent of the region of interest)					
PRBS	22.1 ± 11.0	52.8 ± 7.6	67.4 ± 8.2	65.8 ± 7.3	50.7 ± 4.9
chirp	14.0 ± 2.6	20.0 ± 6.3	72.4 ± 9.0	41.9 ± 13.8	54.4 ± 9.0
sep	22.0 ± 9.1	39.7 ± 7.3	64.4 ± 7.5	66.0 ± 7.1	57.2 ± 4.3
rec	46.4 ± 6.6	53.2 ± 7.7	74.8 ± 6.5	65.6 ± 5.2	69.1 ± 3.1
p&a	37.1 ± 9.0	51.6 ± 7.5	75.9 ± 5.3	66.1 ± 5.9	65.3 ± 3.8
greedy	10.5 ± 1.0	16.4 ± 1.1	68.2 ± 4.6	51.9 ± 5.3	67.1 ± 3.2

tem identification do not consider the current model and therefore, can perform arbitrarily badly. Nonetheless, PRBS explored surprisingly well for systems that are close to linear, e.g., rigid-body dynamics with torque control, where several states are linear in the input. However, this is not the case for other types of systems (e.g., DIPC), and PRBS is often not a desirable system behavior.

6. Conclusion

When learning models of dynamical systems, efficient exploration is key as it determines how informative the collected data is. In this paper, we propose and benchmark three main algorithms for actively learning dynamical systems with GPs. The separated search and control method is inspired by active learning for static GPs. However, its performance is suboptimal, and the theoretical guarantees of the static case are not directly applicable. More efficient exploration can be obtained by computing optimal excitation signals with respect to an information criterion. The receding horizon variant of this approach performs well but yields a high computational burden. Hence, we also propose a batch update that trades off computation against performance. This framework is efficient but also versatile, as further modifications of the cost function are straightforward. We show on a numerical benchmark of diverse dynamical systems that the proposed methods are capable of exploring the state-action space efficiently, yielding more informative data, and hence, a more accurate model. In future work, we intend to study the effects of different cost functions and to generalize our framework to more realistic GP models, for example with noisy inputs and latent states. Validation on hardware experiments would also be relevant, however, the methods need to be computationally optimized first.

Acknowledgments

This work was supported in part by the Cyber Valley Initiative, the International Max Planck Research School for Intelligent Systems, and the Max Planck Society.

References

Charu C. Aggarwal. Active Learning: A Survey. In *Data Classification: Algorithms and Applications*, chapter 22, pages 571–605. Chapman and Hall/CRC, 2014.

Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2018.

Jonathan Binney, Andreas Krause, and Gaurav S. Sukhatme. Informative path planning for an autonomous underwater vehicle. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4791–4796, 2010.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Marc P. Deisenroth and Carl E. Rasmussen. PILCO: A model-based and data-efficient approach to policy search. *Proceedings of the 28th International Conference on Machine Learning*, pages 465–472, 2011.

Andreas Doerr, Christian Daniel, Duy Nguyen-Tuong, Alonso Marco, Stefan Schaal, Marc Toussaint, and Sebastian Trimpe. Optimizing long-term predictions for model-based policy search. *Proceedings of the 1st Conference on Robot Learning*, 78:227–238, 2017.

Andreas Doerr, Christian Daniel, Martin Schiegg, Duy Nguyen-Tuong, Stefan Schaal, Marc Toussaint, and Sebastian Trimpe. Probabilistic recurrent state-space models. *Proceedings of the 35th International Conference on Machine Learning*, pages 1280–1289, 2018.

Stefanos Eleftheriadis, Thomas F.W. Nicholson, Marc P. Deisenroth, and James Hensman. Identification of Gaussian process state space models. *Advances in Neural Information Processing Systems*, pages 5310–5320, 2017.

Jaime F. Fisac, Anayo K. Akametalu, Melanie N. Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J. Tomlin. A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2019.

Michael Green and John B. Moore. Persistence of excitation in linear systems. *Systems and Control Letters*, 7:351–360, 1986.

Steve Heim, Alexander von Rohr, Sebastian Trimpe, and Alexander Badri-Spröwitz. A Learnable Safety Measure. *Proceedings of the 3rd Conference on Robot Learning*, 2019.

Achin Jain, Truong Nghiem, Manfred Morari, and Rahul Mangharam. Learning and control using Gaussian processes. *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*, pages 140–149, 2018.

Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. *Proceedings of the 57th IEEE Conference on Decision and Control*, pages 6059–6066, 2018.

Andreas Krause and Carlos Guestrin. Near-optimal Nonmyopic Value of Information in Graphical Models. *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pages 324–331, 2005.

Andreas Krause and Carlos Guestrin. Nonmyopic Active Learning of Gaussian Processes: An Exploration-Exploitation Approach. *Proceedings of the 24th International Conference on Machine Learning*, pages 449–456, 2007.

Andreas Krause, Ajit Paul Singh, and Carlos Guestrin. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

Lennart Ljung. System identification. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 2001.

David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

Andrew McHutchon and Carl Edward Rasmussen. Gaussian Process Training with Input Noise. *Advances in Neural Information Processing Systems*, pages 1341–1349, 2011.

Oliver Nelles. *Nonlinear System Identification*. Springer, Berlin, Heidelberg, 2001.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, 14:265–294, 1978.

Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: A survey. *Cognitive Processing*, 12:319–340, 2011.

Carl E. Rasmussen and Christopher K.I Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Thomas B Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.

Johan Schoukens and Lennart Ljung. Nonlinear System Identification: A User-Oriented Roadmap. *arXiv preprint arXiv:1902.00683*, 2019.

Matthias Schultheis, Boris Belousov, Hany Abdulsamad, and Jan Peters. Receding Horizon Curiosity. *Proceedings of the 3rd Conference on Robot Learning*, 2019.

Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer, London, 1st edition, 2009.

Max Simchowitz, Horia Mania, Stephen Tu, Michael I. Jordan, and Benjamin Recht. Learning Without Mixing: Towards A Sharp Analysis of Linear System Identification. *Proceedings of the 31st Annual Conference on Learning Theory*, 75:1–35, 2018.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58:3250–3265, 2012.

Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and Stabilization of Complex Behaviors through Online Trajectory Optimization. *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913, 2012.

Yuval Tassa, Nicolas Mansard, and Emanuel Todorov. Control-Limited Differential Dynamic Programming. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1168–1175, 2014.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

Firdaus E Udwadia and Robert E Kalaba. *Analytical Dynamics: A New Approach*. Cambridge University Press, 2007.

Jack Umenberger, Mina Ferizbegovic, Thomas B Schön, and Håkan Hjalmarsson. Robust exploration in linear quadratic reinforcement learning. *arXiv preprint arXiv:1906.01584*, 2019.

Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2005.

Christoph Zimmer, Mona Meister, and Duy Nguyen-Tuong. Safe active learning for time-series modeling with Gaussian processes. *Advances in Neural Information Processing Systems*, pages 2730–2739, 2018.