A survey on policy search algorithms for learning robot controllers in a handful of trials

Konstantinos Chatzilygeroudis[†], Vassilis Vassiliades[†*], Freek Stulp[‡], Sylvain Calinon[◇] and Jean-Baptiste Mouret[†]

Abstract—**Most policy search algorithms require thousands of training episodes to find an effective policy, which is often infeasible with a physical robot. This survey article focuses on the extreme other end of the spectrum: how can a robot adapt with only a handful of trials (a dozen) and a few minutes? By analogy with the word "big-data", we refer to this challenge as "micro-data reinforcement learning". We show that a first strategy is to leverage prior knowledge on the policy structure (e.g., dynamic movement primitives), on the policy parameters (e.g., demonstrations), or on the dynamics (e.g., simulators). A second strategy is to create data-driven surrogate models of the expected reward (e.g., Bayesian optimization) or the dynamical model (e.g., model-based policy search), so that the policy optimizer queries the model instead of the real system. Overall, all successful micro-data algorithms combine these two strategies by varying the kind of model and prior knowledge. The current scientific challenges essentially revolve around scaling up to complex robots, designing generic priors, and optimizing the computing time.**

Index Terms—**Learning and Adaptive Systems, Autonomous Agents, Robot Learning, Micro-Data Policy Search**

I. Introduction

Reinforcement learning (RL) [1] is a generic framework that allows robots to learn and adapt by trial-and-error. There is currently a renewed interest in RL owing to recent advances in deep learning [2]. For example, RL-based agents can now learn to play many of the Atari 2600 games directly from pixels [3], [4], that is, without explicit feature engineering, and beat the world's best players at Go and chess with minimal human knowledge [5]. Unfortunately, these impressive successes are difficult to transfer to robotics because the algorithms behind them are highly data-intensive: 4.8 million games were required to learn to play Go from scratch [5], 38 days of play (real time) for Atari 2600 games [3], and, for example, about 100 hours of simulation time (much more for real time) for a 9-DOF mannequin that learns to walk [6].

By contrast, robots have to face the real world, which cannot be accelerated by GPUs nor parallelized on large clusters. And the real world will not become faster in a few years, contrary to computers so far (Moore's law). In concrete terms, this means that most of the experiments that are successful in simulation cannot be replicated in the real world because they would take too much time to be technically feasible. As an example, Levine et al. [7] recently proposed a large-scale algorithm for learning hand-eye coordination for robotic grasping using deep learning. The algorithm required approximately 800000 grasps, which were collected within a period of 2 months using 6-14 robotic manipulators running in parallel. Although the results are promising, they were only possible because they could afford having that many manipulators and because manipulators are easy to automate: it is hard to imagine doing the same with a farm of humanoids.

What is more, online adaptation is much more useful when it is fast than when it requires hours — or worse, days — of trial-and-error. For instance, if a robot is stranded in a nuclear plant and has to discover a new way to use its arm to open a door; or if a walking robot encounters a new kind of terrain for which it is required to alter its gait; or if a humanoid robot falls, damages its knee, and needs to learn how to limp: in most cases, adaptation has to occur in a few minutes or within a dozen trials to be of any use.

By analogy with the word "big-data", we refer to the challenge of learning by trial-and-error in a handful of trials as "micro-data reinforcement learning" [8]. This concept is close to "data-efficient reinforcement learning" [9], but we think it captures a slightly different meaning. The main difference is that efficiency is a ratio between a cost and benefit, that is, data-efficiency is a ratio between a quantity of data and, for instance, the complexity of the task. In addition, efficiency is a relative term: a process is more efficient than another; it is not simply "efficient". In that sense, many deep learning algorithms are data-efficient because they require fewer trials than the previous generation, regardless of the fact that they might need millions of time-steps. By contrast, we propose the terminology "micro-data learning" to represent an absolute value, not a relative one: how can a robot learn in a few minutes of interaction? or how can a robot learn in less than 20 trials[1]? Importantly, a micro-data algorithm might reduce the number of trials by incorporating appropriate prior knowledge; this does not necessarily make it more "data-efficient" than another algorithm that would use more trials but less prior knowledge: it simply makes them different because the two algorithms solve a different challenge.

[†]Inria, CNRS, Université de Lorraine, LORIA, F-54000 Nancy, France
[*]Research Centre on Interactive Media, Smart Systems and Emerging Technologies, Dimarcheio Lefkosias, Plateia Eleftherias, 1500, Nicosia, Cyprus
[‡]German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Wessling, Germany
[◇]Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland

[1]It is challenging to put a precise limit for "micro-data learning" as each domain has different experimental constraints, this is why we will refer in this article to "a few minutes" or a "a few trials". The commonly used word "big-data" has a similar "fuzzy" limit that depends on the exact domain.

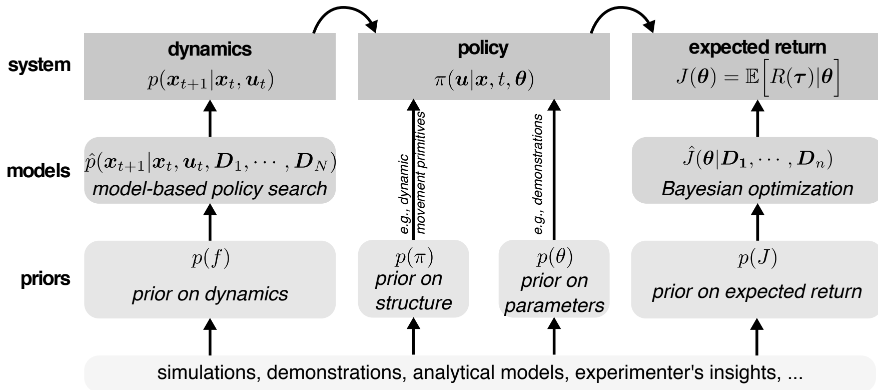

Fig. 1: Overview of possible strategies for Micro-Data Policy Search (MDPS). The first strategy (bottom) is to leverage prior knowledge on the dynamics, on the policy parameters, on the structure of the policy, or on the expected return. A second strategy is to learn surrogate models of the dynamics or of the expected return.

Among the different approaches for RL, most of the recent work in robotics focuses on *Policy Search* (PS), that is, on viewing the RL problem as the optimization of the parameters of a given policy [10] (see the problem formulation, Section II). A few PS algorithms are explicitly focused on requiring very little *interaction time* with the robot, which often implies that they authorize themselves to substantially increase the *computing time* and the amount of *prior knowledge*. The purpose of this paper is to survey such existing micro-data policy search techniques that have been successfully used for robot control [2], and to identify the challenges in this emerging field. In particular, we focus on policy search approaches that have the *explicit goal* of reducing the interaction time between the robot and the environment to a few seconds or minutes [3].

Most published algorithms for micro-data policy search implement and sometimes combine two main strategies (Fig. 1): leveraging prior knowledge (Sections III, IV-B, and V-B) and building surrogate models (Sections IV and V).

Using prior knowledge requires balancing carefully between what can be realistically known before learning and what is left to be learnt. For instance, some experiments assume that demonstrations can be provided, but that they are imperfect [13], [14]; some others assume that a damaged robot knows its model in its intact form, but not the damaged model [15]–[17]. This knowledge can be introduced at different places, typically in the structure of the policy (e.g., dynamic movement primitives [18], Section III), in the reward function (e.g., reward shaping, Section IV-B), or in the dynamical model [17], [19] (Section V-B).

The second strategy is to create models from the data gathered during learning and utilize them to make better decisions about what to try next on the robot. We can further categorize these methods into (a) algorithms that learn a surrogate model of the expected return (i.e., long-term reward) from a starting state [20], [21] (Section IV); and (b) algorithms that learn models of the transition dynamics and/or the immediate reward function (e.g., learning a controller for inverted helicopter flight by first learning a model of the helicopter's dynamics [13], Section V). The two strategies — priors and surrogates — are often combined (Fig. 2); for example, most works with a surrogate model impose a policy structure and some of them use prior information to shape the initial surrogate function, before acquiring any data.

This article surveys the literature along these three axes: priors on policy structure and parameters (Section III), models of expected return (Section IV), and models of dynamics (Section V). Section VI lists the few noteworthy approaches for micro-data policy search that do not fit well into the previous sections. Finally, Section VII sketches the challenges of the field and Section VIII proposes a few "precepts" and recommendations to guide future work in this field.

II. Problem formulation

We model the robots as discrete-time dynamical systems that can be described by transition probabilities of the form:

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t,\boldsymbol{u}_t) \quad (1)$$

where the robot is at state $\boldsymbol{x}_t \in \mathbb{R}^E$ at time t, takes control input $\boldsymbol{u}_t \in \mathbb{R}^F$ and ends up at state $\boldsymbol{x}_{t+1}$ at time $t+1$.

If we assume deterministic dynamics and Gaussian system noise, this equation is often written as:

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t,\boldsymbol{u}_t) + \boldsymbol{w}. \quad (2)$$

[2]Planning-based and model-predictive control [11] methods do not search for policy parameters, this is why they do not fit into the scope of this paper. Although trajectory-based policies and planning-based methods share the same goal, they usually search in a different space: planning algorithms search in the state-action space (e.g., joint positions/velocities), whereas policy methods will search for the optimal parameters of the policy, which can encode a subspace of the possible trajectories.

[3]The scarcity of data in robotics makes it necessary to follow specific strategies when designing learning algorithms. The authors of the present survey organized a very successful workshop on this exact topic at IROS 2017 (Micro-Data: the new frontier of robot learning?) and we think it is the right time to summarize the recent efforts in this direction: while there have been survey articles on policy search in the past (in particular [10], [12]), there have been many exciting developments in the last years (e.g., 50% of the papers cited in our survey have been published between 2013 and 2018). Moreover, our survey focuses on policy search algorithms that have the explicit goal of minimizing the interaction time as much as possible (and not RL or PS algorithms in general), whereas previous surveys had a broader region of interest. Consequently, we can be more thorough in our review and explain the algorithms in more detail.