Sample-efficient learning of soft priorities for safe control with constrained Bayesian optimization

Jian Li[1,2,3], Yiyao Zhu[1,2], Langcheng Huo[1,2,3], Yongquan Chen[1,2*]
[1] Shenzhen Institute of Artificial Intelligence and Robotics for Society, China
[2] Institute of Robotics and Intelligent Manufacturing, The Chinese University of Hong Kong, Shenzhen, China
[3] Guangdong University of Technology
[*] Corresponding author. email: yqchen@cuhk.edu.cn

***Abstract*—A complex motion can be achieved by executing multiple tasks simultaneously, where the key is tuning the task priorities. Generally, task priorities are predefined manually. In order to generate task priorities automatically, different frameworks have been proposed. In this paper, we employed a black-box optimization method, i.e. a variant of constrained Bayesian optimization to learn the soft task priorities, guaranteeing that the robot motion is optimized with high efficiency and no constraints violations occur during the whole learning process.**

I. Introduction

A manipulator can execute multiple tasks simultaneously to achieve a complex motion while keeping the safety constraints satisfied. This is usually accomplished by a prioritized multi-task controller, where task priorities are introduced to represent the relative importance between different tasks. A set of tasks can be organized hierarchically in the light of strict priorities. However, in many cases, it is difficult to define the strict priorities a priori. Another way to organize the tasks is to assign each task with a weighting function, i.e. soft priority which changes the task's relative importance with respect to time in a continuous way. In [1], a new framework is proposed to address this problem, employing Covariance Matrix Adaptation Evolutionary Strategy(CMA-ES) to tune the soft task priorities.

In this paper, we employ a variant of constrained Bayesian optimization(BO) to learn the soft task priorities. In the experiment, we compare the performance between the variant of constrained BO and (1+1)-CMA-ES with constraints with respect to fitness value and constraints violations. Finally, it is validated that tuning task priorities with constrained Bayesian optimization is sample-efficient and leads to no safety constraints violations.

II. METHOD

A. Multi-task controller

A complex motion can be generated by a combination of several elementary tasks achieved by simple controllers. We use a task space controller [2] for the i-th elementary task with an analytical solution. The i-th controller is described as u_i.

In order to achieve a global task through multiple elementary tasks, we assign a time-dependent task weight, i.e. task priority $\alpha_i(t)$ to each elementary task. Due to the infeasibility of finding the optimal function $\alpha_i^*(t)$, we transform this functional optimization problem into a numerical optimization problem, then the task priorities can be represented as parameterized functional approximators, $\alpha_i(t) \rightarrow \alpha_i(\boldsymbol{\pi}_i, t)$, where $\boldsymbol{\pi}_i$ is a set of parameters that determine the time profile of the i-th task priority or weight function. Then elementary tasks can be superimposed to get the final joint command. Therefore the multi-task controller is formulated as:

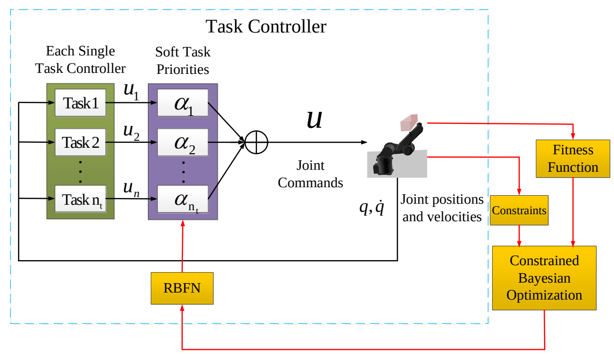

Fig. 1. The framework employed to learning the task priorities automatically. A multi-task controller is formed by superimposing the output of each single task controller assigned with a soft task priority. The time profiles of the task priorities is modeled by a RBFN. A black-box optimization method is used to optimized the parameters of the RBFN, considering the constraints satisfaction.

$$\boldsymbol{u} = \sum_{i=1}^{n_t} \alpha_i \left(\boldsymbol{\pi}_i, t\right) \boldsymbol{u}_i. \tag{1}$$

where t is time, n_t is the number of tasks and $\boldsymbol{u}$ is the output of multi-task controller.

B. Normalized RBFN

For the purpose of transforming a functional optimization problem into a numerical optimization problem, we model the i-th task priority by a normalized radial basis function network (RBFN).

C. Learning soft task priorities

During the whole learning process, in order to find the optimal parameters $\boldsymbol{\pi}^*$ while guarantee safety, fitness function and constraints are introduced. Fitness function ϕ evaluates the performance of the global motion with the present parameters π over T time steps.

Then, the numerical optimization problem can be written as

$$\begin{gathered} \boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}} \boldsymbol{\phi} \\ s.t \quad g_i \leq 0, \quad \forall i \in I_{ie} \end{gathered}, \tag{2}$$

where g_i is safety constraint and I_{ie} represents the set of inequality constraints.

There is no explicit mapping between π and ϕ, so a derivative-free learning method is more general and desired. Additionally, fast computation is possible in trial-and-error learning according to recent research.

D. constrained Bayesian optimization

Bayesian optimization is sample-efficient, requiring very few function evaluations to minimize expensive objective functions. However, there are some safety constraints that need to be considered. So constrained BO is considered a good solution. Some algorithms only guarantee constraints coupled with the performance function that is not desirable in robotics, but this variant of constrained BO [3] can satisfy multiple safety constraints separate from the performance function. Given an initial set of safe parameters, the algorithm minimizes the performance function by evaluating parameters that guarantee safety satisfaction for all constraints with high probability.

III. SIMULATION VERIFICATION

A. Simulation setup

We design a global task as reaching the desired point behind a cuboid with the robot arm's end-effector. Then this global task can be decomposed into three elementary tasks. The first one is to reach a Cartesian position p^* with its end-effector. The second is also a position task, reaching a Cartesian position with its 4-th link. The third is a joint angle position task, reaching a joint angle configuration.

The fitness function is designed as follow:

$$\boldsymbol{\phi} = \frac{1}{2}\left(\frac{\sum_{t=1}^{T} \|\boldsymbol{p}_t - \boldsymbol{p}^*\|_2}{e_{max}} + \frac{\sum_{t=1}^{T} \|\boldsymbol{u}_t\|_2}{u_{max}}\right)$$

where t represents the order of the control step, T is the number of total steps, p_t denotes the end-effector position at control step t, p^* is the goal position, and e_{max} and u_{max} both are scaling factors. The first term of the fitness function penalizes the accumulated error of reaching the final goal while the second term penalizes the sum of joint command at each control step.

B. Comparison with (1+1)-CMA-ES with constraints

In this experiment, we compare constrained Bayesian optimization with the state-of-the-art priorities learning method, (1+1)-CMA-ES with constraints [1], with respect to fitness value and constraints violations. Both of them are starting from a set of safe RBFN parameters. We conduct each simulation

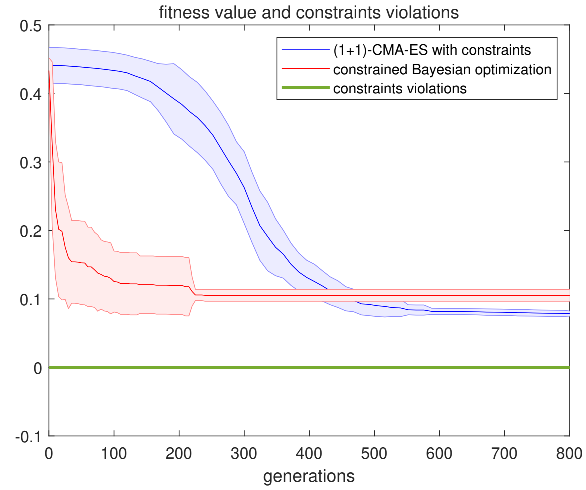

Fig. 2. Fitness and constraints violations comparison between constrained BO and constrained (1+1)-CMA-ES.

over 800 generations for 10 times, and the means, the standard deviation of fitness and the constraints violations are shown in Fig. 2.

As we can see, both of them never break safety constraints during the whole learning process. Moreover, the variant of constrained BO converges a lot faster and is more computationally affordable. When the fitness converged, the fitness mean of (1+1)-CMA-ES with constraints is only slightly better than constrained BO.

IV. Conclusion

This paper employs a sample-efficient approach that explores the solution without violating any safety constraints to tune the task priorities automatically.

This framework is only able to solve the optimization problem offline, due to the long computation time. Future work focuses on reducing the computation time and transferring this feature, i.e. guaranteeing safety constraints satisfaction, to online tuning task priorities.

Acknowledgment

This paper is partially supported by Shenzhen Fundamental Research grant (JCYJ20180508162406177) and the National Natural Science Foundation of China (U1613216) from The Chinese University of Hong Kong, Shenzhen. This paper is also partially supported by funding from Shenzhen Institute of Artificial Intelligence and Robotics for Society.

References

[1] V. Modugno, G. Neumann, E. Rueckert, G. Oriolo, J. Peters, and S. Ivaldi, "Learning soft task priorities for control of redundant robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 221–226.

[2] Y. Wang and A. Kheddar, "Impact-friendly robust control design with task-space quadratic optimization," in *Robotics: Science and Systems (RSS)*, 2019.

[3] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics," *arXiv preprint arXiv:1602.04450*, 2016.