frontiers | Frontiers in Robotics and AI

TYPE Original Research
PUBLISHED 14 October 2022
DOI 10.3389/frobt.2022.993359

OPEN ACCESS

EDITED BY
Chao Zeng,
University of Hamburg, Germany

REVIEWED BY
Lizhi Yang,
California Institute of Technology, United States
Markku Suomalainen,
University of Oulu, Finland

*CORRESPONDENCE
Volker Gabler,
v.gabler@tum.de

SPECIALTY SECTION
This article was submitted to Robot Learning and Evolution, a section of the journal Frontiers in Robotics and AI

RECEIVED 13 July 2022
ACCEPTED 08 September 2022
PUBLISHED 14 October 2022

CITATION
Gabler V and Wollherr D (2022), Bayesian optimization with unknown constraints in graphical skill models for compliant manipulation tasks using an industrial robot.
Front. Robot. AI 9:993359.
doi: 10.3389/frobt.2022.993359

Bayesian optimization with unknown constraints in graphical skill models for compliant manipulation tasks using an industrial robot

Volker Gabler* and Dirk Wollherr

All authors are with the Chair of Automatic Control Engineering, TUM School of Computation, Information and Technology, Technical University of Munich, Munich, Germany

This article focuses on learning manipulation skills from episodic reinforcement learning (RL) in unknown environments using industrial robot platforms. These platforms usually do not provide the required compliant control modalities to cope with unknown environments, e.g., force-sensitive contact tooling. This requires designing a suitable controller, while also providing the ability of adapting the controller parameters from collected evidence online. Thus, this work extends existing work on meta-learning for graphical skill-formalisms. First, we outline how a hybrid force–velocity controller can be applied to an industrial robot in order to design a graphical skill-formalism. This skill-formalism incorporates available task knowledge and allows for online episodic RL. In contrast to the existing work, we further propose to extend this skill-formalism by estimating the success probability of the task to be learned by means of factor graphs. This method allows assigning samples to individual factors, i.e., Gaussian processes (GPs) more efficiently and thus allows improving the learning performance, especially at early stages, where successful samples are usually only drawn in a sparse manner. Finally, we propose suitable constraint GP models and acquisition functions to obtain new samples in order to optimize the information gain, while also accounting for the success probability of the task. We outline a specific application example on the task of inserting the tip of a screwdriver into a screwhead with an industrial robot and evaluate our proposed extension against the state-of-the-art methods. The collected data outline that our method allows artificial agents to obtain feasible samples faster than existing approaches, while achieving a smaller regret value. This highlights the potential of our proposed work for future robotic applications.

KEYWORDS

Bayesian optimization, robot learning and control, episodic reinforcement learning, safe learning, compliant manipulation

1 Introduction

Robotic manipulators have been established as a key component within industrial assembly lines for many years. However, applications of robotic systems beyond such well-defined and usually caged environments remain challenging. Simply reversing the process, i.e., asking a robot to disassemble a product that has been assembled by a robot manipulator in the past, uncovers the shortcomings of currently available (industrial) robot manipulators: the impacts of damage, temporal wear-offs, or dirt most often diminish available model knowledge and thus do not allow an accurate perception of the environment. Rather than relying on the well-defined environment model, robot manipulators are required to account for this uncertainty and thus find a suitable control strategy to interact with the object in a compliant manner. While RL has found remarkable success in dealing with unknown environments, most of these approaches rely on a tremendous amount of data, which are usually costly to obtain, cf. Levine et al. (2015), Levine et al. (2016). In contrast, GPs allow acquiring data efficiently but suffer from poor scaling with respect to state-size and dimension. A previous work has proposed to exploit existing model and task knowledge in order to reduce the parameter space from which a robot has to extract a suitable control policy.

Nonetheless, these approaches have usually been applied on (partially) compliant robots, where constraint violations, e.g., unforeseen contact impulses, can easily be compensated and are, thus, neglected. In the context of this article instead, a non-compliant—i.e., position-controlled—industrial robot is intended to solve manipulation tasks that require compliant robot behavior, such as screwdriver insertion, given a noisy goal location. Therefore, this study outlines an episodic RL scheme that uses Bayesian optimization with unknown constraints (BOC) to account for unsafe exploration samples during learning. In order to apply the proposed scheme on an industrial robot platform that does not provide the default interfaces for compliant controllers, such as a hybrid Cartesian force–velocity controller, we outline a slightly modified version of existing controllers. The resulting controller allows enabling force/velocity profiles along selective axes, while using a high-frequency internal position controller as an alternative fallback. The hybrid nature of this controller allows the direct application of a graphical skill-formalism for meta-learning in robotic manipulation from the previous work. Thus, the state complexity can be reduced to a level where the advantages of GPs outweigh their scaling deficiency. The core contribution of this study lies in the extension and adjustment of BOC to the outlined graphical skill-formalism such that safety constraints can not only be incorporated but also directly added to the graphical skill-formalism. Specifically, we outline how the underlying graph structure can be extended to directly account for safety constraints and thus improve exploration behavior during early exploration stages, where a successful episode is unlikely.

Before sketching our contribution against the related work below, we present a brief outline of the notation and terminology used in this article. Given the mathematical problem in Section 2, we shortly sketch the methodical background of our work in Section 3 and outline the technical insights of our approach in Section 4. Eventually, we outline a specific application example in Section 5 and present our experimental results collected with an industrial robot manipulator in Section 6 before concluding this article in Section 7.

1.1 Notation

In order to outline the notation used throughout this article, we use an arbitrarily chosen placeholder variable $\mathfrak{p}$. Given this placeholder variable is a scalar term, we denote vectors as $p \in \mathbb{R}^n$ and matrices as $P \in \mathbb{R}^{m \times n}$. Explicit elements of matrices or vectors are denoted as $[P]_{(i,j)}$. The identity vector and identity matrix are denoted as $\mathbf{1}^p$ and $\mathbf{1}^{p \times p}$, and similarly, as $\mathbf{0}^p$ and $\mathbf{0}^{p \times p}$ as the zero vector and zero matrix, respectively.

A temporal sequence of vectors p over time is described as a trajectory $\vec{p} := (p_1, p_2, \ldots p_T)$. Indexing over time is denoted as $\mathfrak{p}_t$, where the time is indexed as t. In order to increase readability, the time indexing may be omitted and every variable is expected to be denoted as $\mathfrak{p}_t$. For these cases, the temporal successor is denoted as $\mathfrak{p}' := \mathfrak{p}_{t+1}$. If we refer to a member of containers such as sets, lists, and vectors, we denote $\mathfrak{p}_i$ as a specific scalar value of the former. We denote norms as $\|\mathfrak{p}\|_i$, i.e., $\|\mathfrak{p}\|_2$ represents the Euclidean norm. In order to express the dimension of vectors, we use $|\mathfrak{p}|$. If $|\mathfrak{p}|$ is applied on sets, the cardinality is used.

Expected values of a stochastic variable $\mathfrak{p}$ are written as $\mathbb{E}_{\mathfrak{p}}[\cdot]$, whereas the conditional probability given $\mathfrak{p}$ is denoted as $\mathbb{P}[\cdot|\mathfrak{p}]$. If a variable is estimated, we denote this by $\hat{\mathfrak{p}}$. In order to denote binary classification labels or success/failure returns, we denote $\top$ as Boolean *true* and $\bot$ as Boolean *false*.

If $\mathfrak{p}$ is used to optimize an objective, the optimal solution is denoted as $\mathfrak{p}^\star$. Within regression or empirical evaluations, where the actual ground-truth is known, we denote the ground-truth as $\mathfrak{p}^\bigstar$. In case either the true optimum or ground-truth is estimated from collected experience, i.e., evidence, we denote $\mathfrak{p}^\circledast$ as the currently best performing sample and $\mathfrak{p}^\circledcirc$ as the worst performing observed data sample.

For kinematic robot chains, we refer to the origin of the chain as base $\mathfrak{ba}$, the end-effector as $\mathfrak{ee}$, and the control frame as $\mathfrak{ct}$. Coordinate transformation matrices are denoted as as ${}^{\mathfrak{z}}T_{\mathfrak{y}}$ and rotation matrices as $R^{\mathfrak{z}}\mathfrak{y}$, as a transformation/rotation from $\mathfrak{y}$ to $\mathfrak{z}$. We denote $\boldsymbol{R}_\varphi$, $\boldsymbol{R}_\theta$, and $\boldsymbol{R}_\psi$ as the rotation matrices around coordinate axes $\boldsymbol{e}_x$, $\boldsymbol{e}_y$, and $\boldsymbol{e}_z$, respectively Regarding translational notations, ${}^{\mathfrak{ba}}p_{\mathfrak{eect}}$ describes a vector p pointing from the end-effector to the control frame, expressed in the base frame. If no explicit reference frame is provided, the variable is given with

respect to $\mathfrak{ba}$ for robotic systems and as the world-frame for generic settings.

Eventually, we summarize by shortly defining the terminology of this work. For brevity, we only highlight technical terms, which distinctly differ in their meaning across research fields.

- A (manipulation) task describes the challenge for a robot to reach a predefined goal-state, closely related to the definitions from automated planning (Nau et al., 2004). As this work focuses on episodic RL, the result of an episode is equal to the outcome of a task.
- A manipulation primitive (MP) defines a sub-step of a task. In contrast to automated planning, this work does not intend to plan a sequence of (feasible) MPs but instead focuses on the parameterization of a predefined sequence of MPs. In contrast to hierarchical planning, we omit further hierarchal decompositions—e.g., methods (Nau et al., 2004)—such that a task can only be realized as a sequence of primitives.
- Using such MPs in order to solve a manipulation task directly leads to the introduction of the term of a skill. While a (robotic or manipulation) skill denotes the ability of a robot to achieve a task, we explicitly use the term (graphical) skill-formalism to denote a specific realization of sequential MPs to solve a manipulation task.
- Our work seeks to increase the learning speed for episodic RL by limiting learning to a reduced parameter space, which we denote as meta-learning as used in the existing work (Johannsmeier et al., 2019). It still has to be noted that this terminology is different to common terminologies such as meta-RL (Frans et al., 2018).
- Within episodic RL, a robot is usually asked to find an optimal parameter sample or a policy with respect to a numeric performance metric that is obtained at the end of a multi-step episode. In the scope of this work, we specifically focus on the former, i.e., a robot is asked to sample parameter values during the learning phase. Similar to literature in Bayesian optimization, we often denote this sampling process as the acquisition of samples (Rasmussen and Williams, 2006). Eventually, the performance metric is obtained at the very end of a successful trial episode.

1.2 Related work

In the context of learning force-sensitive manipulation skills, a broad variety of research work has been presented in the last decade. Profiting from compliant controllers that were designed to mimic human motor skills (Vanderborght et al., 2013), the concept of adaptive robot skills has found an interesting way beyond an adaptive control design. Thus, this section outlines the state-of-the-art methods across multiple research fields before setting the contribution of this article in relation to these works.

1.2.1 Force-adaptive control for unknown surfaces or objects

As covering all aspects of interacting with unknown surfaces, e.g., tactile sensing (Li Q. et al., 2018), is beyond the scope of this article, we refer to existing surveys (Li et al., 2020) and specifically summarize findings on learning force-adaptive manipulation skills.

In this context, the peg-in-hole problem is one of the most covered research challenges. Early work, such as Gullapalli et al. (1992) and Gullapalli et al. (1994), proposed to apply machine learning (ML), e.g., real-valued RL, to learn a stochastic policy for the peg-in-hole task. The neural networks for the force controllers were trained by conducting a search guided by an evaluative performance feedback.

In addition to ML, many approaches have applied learning from demonstration to obtain suitable Cartesian space trajectories, cf. Nemec et al. (2013) or Kramberger et al. (2016) who adjust dynamic movement primitives conditioned on environmental characteristics using online inference. While initial attempts have focused on adjusting the position of the robot end-effector directly, recent approaches have also investigated the possibility of replicating demonstrated motor skills that also involve interaction wrenches (Cho et al., 2020) or compliant behavior (Deniša et al., 2016; Petric et al., 2018).

Alternative work proposes adaptive controllers that adjust the gains of a Cartesian impedance controller as well as the current desired trajectory based on the collected interaction dynamics. Yanan Li et al. (2018) evaluated observed error-dynamics, current pose, velocity, and excited wrenches.

Even though these works have achieved great results for modern and industrial robot manipulators in their application fields, they do not allow robots to autonomously explore and refine a task. While learning from demonstration always requires a demonstration to be given, adaptive controllers assume to have access to a desired state or trajectory. In addition, the majority of proposed controllers usually require high-frequency update rate on the robot joints, cf. Scherzinger et al. (2019b); Stolt et al. (2012); Stolt et al. (2015), which usually is only accessible for the robot manufacturer. In contrast to this, this study seeks for a setup that can be deployed on off-the-shelf industrial robot manipulators.

A few years ago, the idea of end-to-end learning *via* means of deep RL techniques had been studied thoroughly to combine the efforts of the former and the latter in a confined black-box system. In these studies, the concept of controlling the gains is omitted and instead replaced by a feed-forward torque policy that generates joint-torques from observed image data using a deep neural network (NN). Levine et al. (2015) and Levine et al. (2016) used a guided policy search that leverages the need for well-known models or demonstrations. Instead, the system learns contact-rich manipulation skills and trajectories through time-varying

linear models that are unified into a single control policy. Devin et al. (2017) have tackled the issue of slow converging rates due to the enormous amount of required data by introducing distributed learning, where evidence is shared across robots, and the network structure allows distinguishing between task-specific and robot-specific modules. These models are then trained by means of mix-and-match modules, which can eventually solve new visual and non-visual tasks that were not included in the training data. The issue of low precision has been improved by Inoue et al. (2017), who evaluated the peg-in-hole task with a tight clearance.

Recently, the application of deep RL has reverted to use existing controllers and improve their performance by applying deep NNs in addition, e.g., Luo et al. (2019) proposed to learn the interaction forces as Pfaffian constraints *via* a NN. Beltran-Hernandez et al. (2020) applied an admittance controller for a stiff position-controlled robot in a joint space and applied RL *via* soft actor-critic (Haarnoja et al., 2018) to achieve a compliant robot behavior that successfully learns a peg-in-hole task by adjusting the gains of the admittance controller. Similarly, the feed-forward wrench for an insertion task is learnt from human demonstrations (Scherzinger et al., 2019a) using NNs and a Cartesian admittance controller tailored to industrial platforms (Scherzinger et al., 2017).

Aside from the aspect of meta-RL (Frans et al., 2018; Gupta et al., 2018), which investigates the idea of bridging data generated in simulations to physical platforms, the performance benefits and ability to learn almost arbitrarily complex tasks and existing methods for deep RL still require a tremendous amount of experimental data to be collected to achieve reliable performance.

1.2.2 Robot skill learning on reduced parameter spaces

The size of required data is directly subject to the size of the parameter space that needs to be regressed. Thus, another promising line of research is given by decreasing the search space and problem complexity.

A recent research work has proposed to use available expert knowledge rather than learning a skill from scratch. LaGrassa et al. (2020) proposed to categorize the working space into regions where model knowledge is sufficient and into unknown regions, where a policy is obtained *via* deep RL. Johannsmeier et al. (2019) proposed to incorporate expert knowledge in order to reduce the search space for adaptive manipulation skills by introducing MPs. On this basis, they showcased a peg-in-hole task, where a robot adjusts the stiffness and a feed-forward interaction wrenches of a Cartesian impedance controller by means of Bayesian optimization (BO) and black-box optimization.

The application of such MPs also encouraged the application of deep RL approaches. Zhang et al. (2021) proposed two RL approaches based on the principle of MPs, where the policy is represented by the feed-forward Cartesian wrench and the gains of a Cartesian impedance controller. Martín-Martín et al. (2019) similarly proposed to learn the controller selection and parameterization during a peg-in-hole task. Hamaya et al. (2020) applied a model-based RL *via* GP on a peg-in-hole task for an industrial position-controlled robot by attaching a compliant wrist to the robot end-effector, which compensates for perception inaccuracy. Mitsioni et al. (2021) instead proposed to learn the environment dynamics from an NN in order to apply a model predictive control, if the current state is classified as safe *via* a GP classifier. Alt et al. (2021) also applied NNs *via differentiable shadow programs* that employ the parameterization of robotic skills in the form of Cartesian poses and wrenches in order to achieve force-sensitive manipulation skills, even on industrial robots. They include the success probability in the output of the NNs, in order to minimize the failure rate.

While these approaches have shown promising results by solely collecting experimental data within reasonable time, neither of those approaches include interaction constraints—e.g., maximum contact wrenches—during the acquisition or evaluation of new data samples nor allow the application of the presented results on an industrial platform without an additional compensation unit. As for the former, the majority of research projects have applied BOC to account for safety critical or unknown system constraints during learning, and we continue with a dedicated overview of research in this field.

1.2.3 Bayesian optimization with unknown constraints for robotics

Within robotic applications, BO has shown potential in achieving online RL due to effective acquisition of new samples (Deisenroth et al., 2015; Calandra et al., 2016), that is still used within robotic research applications (Demir et al., 2021).

In the context of BOC, safe RL methods have been proposed that estimate safe or feasible regions of the parameter space into account to allow for safe exploration, cf. Berkenkamp et al. (2016a,b), Sui et al. (2015), or Baumann et al. (2021).

Similarly, Englert and Toussaint (2016) proposed the probability of improvement with a boundary uncertainty criterion (PIBU) acquisition function that encourages exploration in the boundaries of safe states. Their approach was further evaluated on generalizing small demonstration data autonomously in Englert and Toussaint (2018) as well as on force-adaptive manipulation tasks by Drieß et al. (2017). A similar acquisition function has been proposed by Rakicevic and Kormushev (2019), even though they do not approximate the success as a GP.

Approaches such as those by Wang et al. (2021), who used GPs to regress the success of an atomic planning skill from data,

have further shown that BOC is well-suited to regress high level, i.e., task-planning constraints from data. While they approximated this success probability as a constraint with a predefined lower bound 0, Marco et al. (2021) outlined a constraint-aware robot learning method based on BOC that allows improving sampling even if no successful sample is available yet. Recent practical application examples of BOC are found in Khosravi et al. (2022), Stenger et al. (2022), and Yang et al. (2022).

While these approaches have achieved promising results within small-scale (robot) learning problems, they suffer from poor scaling properties as GPs require to use the covariance matrix for prediction and acquisition of new data samples, which grows exponentially in the state space of the underlying problem. While various works have focused on finding proper approximation methods to leverage this problem, we propose that within a robotic context, it is preferable to explicitly incorporate structural knowledge whenever possible. To conclude this overview of the state-of-the-art methods, we shortly summarize the contribution of this article in relation to the work stated previously.

1.3 Contribution

This study introduces a novel episodic RL-scheme for compliant manipulation tasks tailored to industrial robots. In order to allow for compliant manipulation tasks, the control interfaces of an industrial robot are adjusted to follow a Cartesian hybrid force–velocity controller (Craig and Raibert, 1979; Khatib and Burdick, 1986). By exploiting the hybrid nature of this controller and available expert knowledge, a complex manipulation task can be reformulated into graphical skill-formalisms—i.e., a sequence of simplified MPs—from existing work. Eventually, we outline an extension of these graphical skill-formalisms by taking into account parameter constraints and success-probabilities at each sub-step. This improves learning especially at early stages and allows refining the individual sub-steps of a robotic manipulation task even when no successful episode could have been observed yet. Furthermore, we define suitable BOC models to estimate the success probability of each MP as well as the overall task, as well as the outline of suitable acquisition functions that allow collecting data efficiently during learning.

2 Problem formulation

The mathematical problem tackled in this article is the optimization of an unknown objective function $\mathcal{J}(\boldsymbol{\xi})$ with respect to meta-parameter vector $\boldsymbol{\xi}$ subject to unknown constraints $\boldsymbol{g}$

$$\begin{aligned} &\min \mathcal{J}(\boldsymbol{\xi}) \quad \boldsymbol{\xi} \in \mathbb{R}^m \\ &\text{s.t.} \quad g_i(\boldsymbol{\xi}) \leq c_i, \forall i \in [1, |\boldsymbol{c}|] \end{aligned}, \tag{1}$$

specifically tailored to robotic applications. In here, the objective $\mathcal{J}(\boldsymbol{\xi})$ describes the performance metric of a task, whereas a finite set of constraints $\boldsymbol{g}(\boldsymbol{\xi}) \leq \boldsymbol{c}$ defines a safe subset of the meta parameter space $\boldsymbol{\xi}$. In the context of this article, this function mapping $\mathcal{J}(\boldsymbol{\xi})$, as well as the constraints—i.e., $\boldsymbol{g}$ and $\boldsymbol{c}$—,are regressed from data by means of episodic RL. In contrast to most RL approaches, where the environment is assumed to be Markovian, episodic RL needs to execute a multi-step exploration before obtaining a feedback, which can be used to update the current model(s). In the scope of this work, an episode is given as a manipulation task, which can be either be evaluated in simulation or directly on a robot platform. In the remainder of this work, we mainly focus on the direct application on the latter. Similar to related work in this area (Marco et al., 2021), we assume that the feedback of an episode is expected to be given in the form of

$$\mathcal{J}_{\text{spl}}, \boldsymbol{g}_{\text{spl}}, \boldsymbol{s}_{\text{spl}} \leftarrow \begin{cases} \mathcal{J}(\boldsymbol{\xi}), \boldsymbol{g}(\boldsymbol{\xi}), & \top & \text{iff } g_i(\boldsymbol{\xi}) \leq c_i, \forall i \in [1, |\boldsymbol{c}|] \\ \infty, & \infty, & \bot & \text{else} \end{cases}, \tag{2}$$

as the current performance sample $\mathcal{J}_{\text{spl}}$ and the constraint and success-return vectors $\boldsymbol{g}_{\text{spl}}, \boldsymbol{s}_{\text{spl}} \in \mathbb{R}^{|\boldsymbol{c}|}$. Therefore, a major challenge lies in handling episodes where *infeasible/unsafe* parameters have been selected, and neither information about $\mathcal{J}$ nor the constraint metric is gained. It is often expensive to select and evaluate new samples within robotic applications. GP regression has shown great potential in ML and robotics, if only a handful of samples should be evaluated. Thus,

$$\begin{aligned} \mathcal{J}(\boldsymbol{\xi}) &\leftarrow \hat{\mathcal{J}}(\boldsymbol{\xi} \mid \mathcal{D}_{\mathcal{J}}) \sim \mathcal{N}\left(\mu_{\mathcal{J}}, \boldsymbol{\Sigma}_{\mathcal{J}}\right) \\ g_i(\boldsymbol{\xi}) \leq c_i &\leftarrow \hat{g}_i\left(\boldsymbol{\xi} \mid \mathcal{D}_{g_i}\right) \sim \mathcal{N}\left(\mu_{gi}, \boldsymbol{\Sigma}_{gi}\right) \quad \forall i \in [1, |\boldsymbol{c}|] \end{aligned}, \tag{3}$$

approximate the objective $\mathcal{J}$ and constraints $\boldsymbol{g}_i$ *via* GPs using collected empirical data $\mathcal{D}_{\mathcal{J}} = \{\boldsymbol{\xi}, \mathcal{J}(\boldsymbol{\xi})\}$ and $\mathcal{D}_{g_i} = \{\boldsymbol{\xi}, (g_i(\boldsymbol{\xi}), \{\top, \bot\})\}$. Finally, the optimal guess for (1) can be obtained by minimizing the posterior of $\hat{\mathcal{J}}(\boldsymbol{\xi})$:

$$\boldsymbol{\xi}^* \leftarrow \arg\min_{\boldsymbol{\xi}} \mathbb{E}_{\hat{\mathcal{J}}}\left[\hat{\mathcal{J}}(\boldsymbol{\xi}) \prod_{i=1}^{|\boldsymbol{c}|} \mathbb{P}\left[\hat{g}_i(\boldsymbol{\xi})\right]\right], \tag{4}$$

weighted by the success probability of $\boldsymbol{\xi}$ given as the joint probability over all constraints. Thus, (4) does not only optimize the main task-objective but also accounts for the probability of violating imposed constraints. This directly allows optimizing the performance of an unknown manipulation task for robotic systems, while accounting for constraints such as limited interaction wrenches during contact tooling.

3 Preliminaries and background

Before outlining our approach in detail, we provide a brief introduction into the graphical skill-formalisms from

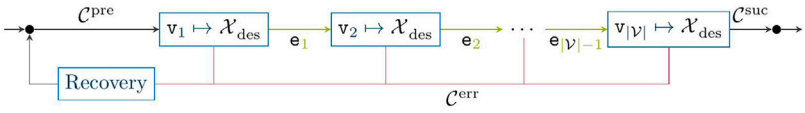

FIGURE 1
Schematic skill-formalism for manipulation tasks as presented in Johannsmeier et al. (2019). Each MP—i.e., node v_i defines the current set-values for the underlying controller, e.g., desired wrench or velocity, as well the current meta parameters that define the performance of the skill, e.g., controller parameterization. Eventually, the *Recovery* node intends to steer the robot to the initial state whenever an error occurs.

Johannsmeier et al. (2019) and the BOC approach from Marco et al. (2021) and Englert and Toussaint (2016), which we use as a baseline comparison in our experimental evaluation.

3.1 Meta-learning for robotic systems using graphical skill formalisms

Within robotic tasks, the hyper-parameter space is usually large due to the degrees of freedom in SE (3) or the configuration space of the robot. Therefore, Johannsmeier et al. (2019) proposed to model tasks in fine-grained Moore finite-state automaton (FSA), according to the schematic shown in Figure 1. The vertices $\mathcal{V}$ of the FSA graph G define MPs as atomic primitive tasks. In these FSAs, the output alphabet is defined by the meta parameters $\boldsymbol{\xi}$ and desired set-values, e.g., $\boldsymbol{x}_{\text{des}}$, that are sent to the robot at each MP, denoted as the dedicated space $\mathcal{X}^{\text{des}}$ in Figure 1. Therefore, the more task knowledge can be exploited for each MP, the smaller the space of the resulting meta-parameter per node.

Eventually, the manipulation skill is further defined by a set of constraints that define the start- and end-constraints, as well as any time constraints that the robot shall never violate. This provides the benefit of exploiting available object knowledge, while also providing a skill-formalism that is closely related to that of automated task planning (Nau et al., 2004). In fact, these constraints are closely related to autonomous planning and first-order logic, where planning primitives are often described by a set of *pre-conditions* and *effects*. In the context of concurrent planning, this is also extended to any time constraints that must not be violated while the task primitive is executed. This results in a skill representation as shown in Figure 1, where the task-constraints are defined as deterministic mapping functions $\mathcal{C}\colon = \mathcal{X} \mapsto \{\bot, \top\}$, which map the state space of the robot to a Boolean return value. In particular, individual manipulation skills are defined by

- Initialization constraints $\mathcal{C}^{\text{pre}}$ or pre-conditions. They define the initialization of the task. In general, $\mathcal{C}^{\text{pre}}$ is given as a set of constraints that only evaluates to $\top$, if all conditions evaluate to $\top$, i.e., if s_0 denotes the initial state of the robot, then $c(\mathrm{s}_0) \mapsto \top, \forall c \in \mathcal{C}^{\text{pre}}$ has to hold.
- Success-constraints $\mathcal{C}^{\text{suc}}$ or termination-conditions. They evaluate if the manipulation skill has been executed successfully. This terminates the overall FSA shown in Figure 1 and requires all conditions to evaluate to $\top$, i.e., if $\mathrm{s}_{T_{\max}}$ denotes the final state of the robot in the manipulation skill, then $c(\mathrm{s}_{T_{\max}}) \mapsto \top, \forall c \in \mathcal{C}^{\text{suc}}$ has to hold.
- Safety and performance constraints $\mathcal{C}^{\text{err}}$ or error conditions. They evaluate if the current MP has violated any constraints, e.g., timeouts or accuracy violations, which may exceed information provided by a task planner. In contrast to $\mathcal{C}^{\text{pre}}$ and $\mathcal{C}^{\text{suc}}$, the error constraint set $\mathcal{C}^{\text{err}}$ evaluates to $\top$ if any condition is violated at any time, i.e., if s_t denotes the state of the robot at any time during the manipulation skill, then $\exists c \in \mathcal{C}^{\text{err}}\colon c(\mathrm{s}_t) \mapsto \top$ has to be fulfilled. Furthermore, the robot enters a recovery node, in which the robot tries to reach the initial state to initiate a new trial-episode—as emphasized by the dashed line in Figure 1.

In the context of the graphical skill-formalism from Johannsmeier et al. (2019), $\mathcal{C}^{\text{pre}}$ are defined by the adjacency matrix of the graph and the success-constraint from the predecessor-node, i.e., if a node raises the success-constraint, there is a unique successor-node, whose precondition holds by design.

3.2 Bayesian optimization with unknown constraints

Within BO, an unknown function or system is regressed from data as a stochastic process. A common model is a GP, which is defined as a collection of random variables, namely, joint normally distributed functions over any subset of these variables. They are fully described by their second-order statistics, i.e., a prior mean and a covariance kernel function $\boldsymbol{k}(\boldsymbol{\xi}, \boldsymbol{\xi}')$, which encodes prior function properties or assumptions.[1] A key benefit of stochastic processes is their ability to draw samples efficiently. This strongly depends on the choice of the acquisition function α, which usually intends to

1 For more information about GPs and GP classification, we refer to Rasmussen and Williams (2006).

maximize the information gain for the estimated posterior y. Famous examples are the expected improvement (EI) and expected improvement with constraints (EIC).

$$\alpha_{\mathrm{EI}}(\boldsymbol{\xi}, \mathcal{D}) = \mathbb{E}_{\mathrm{y}\sim\mathcal{N}_{\mathcal{J}}(\mu,\sigma|\boldsymbol{\xi})}\left[\max\left(\mathrm{y} - \mathcal{J}_{\mathcal{D}}^{\circledast}, 0\right)\right], \tag{5}$$

$$\alpha_{\mathrm{EIC}}(\boldsymbol{\xi}, \mathcal{D}) = \mathbb{E}_{\mathrm{y}\sim\mathcal{N}_{\mathcal{J}}(\mu,\sigma|\boldsymbol{\xi})}\left[\max\left(\mathrm{y} - \mathcal{J}_{\mathcal{D}}^{\circledast}, 0\right)\prod_{j=0}^{G}\left[\boldsymbol{g}_j(\boldsymbol{\xi}) \le \boldsymbol{c}_j\right]\right], \tag{6}$$

where the probability of improvement (PI) is maximized

$$\mathrm{PI}_{\mathrm{GP}(\mathcal{J})}(\boldsymbol{\xi}) = \Phi\left(\frac{\mu^{\boldsymbol{\xi}}_{\mathrm{GP}(\mathcal{J})} - \mathcal{J}_{\mathcal{D}}^{\circledast}}{\sigma^{\boldsymbol{\xi}}_{\mathrm{GP}(\mathcal{J})}}\right). \tag{7}$$

Here, Φ denotes the normal cumulative distribution function (CDF), whereas $\mathcal{J}_{\mathcal{D}}^{\circledast}$ represents the best output sample in the dataset $\mathcal{D}$, which serves as the lower bound for the improvement. The mean $\mu^{\boldsymbol{\xi}}_{\mathrm{GP}(\mathcal{J})}$ and variance $\sigma^{\boldsymbol{\xi}}_{\mathrm{GP}(\mathcal{J})}$ are obtained as the posterior of the GP at new sample candidates $\boldsymbol{\xi}$. While modeling the task-performance *via* a GP commonly applied in BOC, regressing a discriminative success function is non-trivial. In Englert and Toussaint (2016), Drieß et al. (2017), and Englert and Toussaint (2018), GP classification with a sigmoid function to classify the output of a latent GP is proposed. Given this, the authors propose a constrained sensitive acquisition function, which they denote as PIBU

$$\alpha_{\mathrm{PIBU}}(\boldsymbol{\xi}, \mathcal{D}) = \begin{cases} \mathrm{PI}_{\mathrm{GP}(\mathcal{J})}(\boldsymbol{\xi}) & \hat{g}(\boldsymbol{\xi}) > 0 \\ \sigma_{\mathrm{GP}(\hat{g})}(\boldsymbol{\xi}) & \hat{g}(\boldsymbol{\xi}) \mapsto 0 \end{cases}, \tag{8}$$

that uses the PI in admissible regions of the parameter space and the variance σ of the latent GP in the boundary regions to encourage a safe exploration. They further use a constant negative mean prior for the latent GP to limit sampling to the boundary regions of the safe parameter space. In contrast to this, Marco et al. (2021) proposed to use a constraint-aware GP model that allows using EIC, which they denote as a Gaussian process for classified regression (GPCR). GPCR allows updates even if no successful constraint sample has been drawn yet, based on the environmental feedback in (2). Furthermore, Marco et al. (2021) proposed to regress the constraint thresholds $\boldsymbol{c}_j$ directly from data. Thus, having $N_{\mathrm{spl}} \le |\mathcal{D}|$ successful samples, the likelihood is defined as follows:

$$\mathbb{P}\left[\mathcal{D}|\boldsymbol{g}_j\right] = \prod_{j=0}^{N_{\mathrm{spl}}}\mathrm{H}\left(\boldsymbol{c}_j - \boldsymbol{g}_j\right)\mathcal{N}\left(\boldsymbol{g}_j, \sigma^2_{\mathrm{noise}}\right)\prod_{j=N_{\mathrm{spl}}+1}^{|\mathcal{D}|}\mathrm{H}\left(\boldsymbol{g}_j - c\right), \tag{9}$$

where H denotes the Heaviside function. Using a zero-mean Gaussian prior, the posterior is given as follows:

$$\mathbb{P}[\boldsymbol{g}|\mathcal{D}] = \mathcal{N}(\boldsymbol{g}|\boldsymbol{\mu}_{\mathrm{n}}, \boldsymbol{\Sigma}_{\mathrm{n}})\prod_{i=0}^{N_{\mathrm{spl}}}\mathrm{H}\left(\boldsymbol{c}_j - \boldsymbol{g}_j\right) \times \prod_{j=N_{\mathrm{spl}}+1}^{|\mathcal{D}|}\mathrm{H}\left(\boldsymbol{g}_j - \boldsymbol{c}_j\right) \approx \mathcal{N}(\boldsymbol{g}|\boldsymbol{\mu}_{\mathrm{EP}}, \boldsymbol{\Sigma}_{\mathrm{EP}}), \tag{10}$$

where the Gaussian distribution $\mathcal{N}(\boldsymbol{g}|\mu_{\mathrm{n}}, \boldsymbol{\Sigma}_{\mathrm{n}})$ is obtained by the multivariate Gaussian from the observation noise and the observation samples. As the Heaviside functions in (10) do not allow obtaining an analytic solution for (10), the authors propose to use a variational approximation, namely, expectation propagation (EP), such that the predictive distribution at unobserved samples $\boldsymbol{\xi}'$ is obtained *via* a Gaussian distribution defined by mean and variance:

$$\begin{aligned} \mu_{\boldsymbol{g}_j}(\boldsymbol{\xi}') &= \boldsymbol{k}_{\boldsymbol{\mathfrak{X}}}(\boldsymbol{\xi}')^{\mathrm{T}}\boldsymbol{K}^{-1}\mu_{\mathrm{EP}} \\ \sigma_{\boldsymbol{g}_j}(\boldsymbol{\xi}') &= \boldsymbol{k}(\boldsymbol{\xi}', \boldsymbol{\xi}') - \boldsymbol{k}_{\boldsymbol{\mathfrak{X}}}(\boldsymbol{\xi}')^{\mathrm{T}}\boldsymbol{K}^{-1}\left(\boldsymbol{I}^{|\mathcal{D}|\times|\mathcal{D}|} - \boldsymbol{\Sigma}_{\mathrm{EP}}\boldsymbol{K}^{-1}\right)\boldsymbol{k}_{\boldsymbol{\mathfrak{X}}}(\boldsymbol{\xi}') \end{aligned}, \tag{11}$$

where $\boldsymbol{\mathfrak{X}}$ denotes observed parameter samples in $\mathcal{D}$. The success probability is then given as follows:

$$\mathbb{P}\left[\boldsymbol{g}(\boldsymbol{\xi}) \le \boldsymbol{c}(\boldsymbol{\xi}')\right] = \prod_{i=0}^{|\boldsymbol{c}|}\Phi\left(\frac{\boldsymbol{c}_j - \mu_{\boldsymbol{g}_j}(\boldsymbol{\xi}')}{\sigma_{\boldsymbol{g}_j}(\boldsymbol{\xi}')}\right). \tag{12}$$

4 Technical approach

In order to allow online RL to be applied from a handful of exploration samples, it is favorable to exploit available knowledge and thus decrease the overall meta parameter space of the observed system. As mentioned before, we thus extend the concept of modeling robotic tasks as skill-graphs from Johannsmeier et al. (2019) to allow compliant manipulation tasks to be tuned online. In contrast to preliminary work, we outline how a stiff position-controlled industrial robot platform can be controlled in order to allow for compliant robot behavior. Building upon this, we emphasize how a graphical skill-formalism can exploit the structure of the presented controller, such that the controller parameters can be adjusted online. As crash constraints are critical, if a stiff robot is asked to interact with unknown objects, we conclude our technical contributions by not only outlining how the structure of the skill-graph can be further exploited to simplify the BOC-RL algorithm but also proposing suitable BOC models and acquisition functions in order to improve the overall learning performance.

4.1 Compliant Controller design for an industrial Robot

In the context of this article, we use a COMAU robot[2]. While this robot prohibits the control of the motor torques or impedance-based controller interfaces, it allows controlling the position of the end-effector $\boldsymbol{x}$ of the robot *via* an external

2 The extension to arbitrary robots is subject to the internal robot control and dynamics. As the methods in this article are dynamically independent, this extension is left for future work.

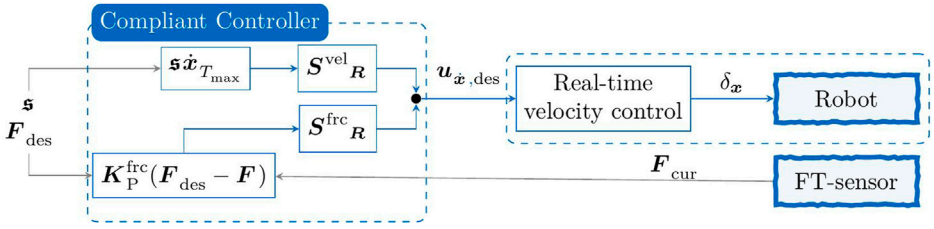

FIGURE 2
Schematic overview of a hybrid force–velocity controller (Craig and Raibert, 1979; Khatib and Burdick, 1986) using the Cartesian deviation control interface of an industrial COMAU robot. Here, the selection matrices $S^{vel}{}_R$ and $S^{frc}{}_R$ activate velocity—and force control modalities along selective axes using a scaled feed-forward velocity profile $s\dot{x}_{T_{max}}$ and a proportional force controller with the gain matrix K_P, Cartesian FT readings F, and the desired wrench F_{des}. Eventually, the Cartesian velocity is emulated on the COMAU robot by using the Cartesian deviation command interface δ_x.

client in the form of a Cartesian deviation relative to the current end-effector pose, such that the controlled system simplifies to

$$x_{t+1} := x_t + \delta_x \approx x_t + u_{\dot{x},des}\delta_t, \tag{13}$$

where δ_x forms the control command being sent to the robot. As the robot runs at a real-time safe, constant update rate δ_t, the Cartesian deviation command $u_{\dot{x},des}$ can also be used to command a feed-forward Cartesian velocity command to the robot. In order to achieve a hybrid force–velocity control policy for the robot system, this feed-forward end-effector velocity follows to a hybrid Cartesian force–velocity controller (Khatib and Burdick, 1986):

$$\begin{aligned} u_{\dot{x},des} &:= S^{vel}{}_R \dot{x}_{eedes} + S^{frc}{}_R K_P (F_{des} - F) \\ &= S^{vel}{}_R \dot{x}_{eemax} + S^{frc}{}_R K_P (F_{des} - F) \end{aligned}, \tag{14}$$

where $s \mapsto [0,1]^6$ is a scaling vector given the maximum end-effector velocity $\dot{x}_{max}$ and K_P is a positive definite proportional control gain matrix. The selection matrices $S^{frc}{}_R$ and $S^{vel}{}_R$ in (14) are given as follows:

$$\begin{aligned} S^{frc}{}_R &:= \begin{bmatrix} R_{ct}^{ba}\,\mathrm{diag}\left(s^{frc}_{1:3}\right)R_{ba}^{ct} & 0^{3\times3} \\ 0^{3\times3} & R_{ct}^{ba}\,\mathrm{diag}\left(s^{frc}_{3:6}\right)R_{ba}^{ct} \end{bmatrix}, \\ S^{vel}{}_R &:= \begin{bmatrix} R_{ct}^{ba}\,\mathrm{diag}\left(s^{vel}_{1:3}\right)R_{ba}^{ct} & 0^{3\times3} \\ 0^{3\times3} & R_{ct}^{ba}\,\mathrm{diag}\left(s^{vel}_{3:6}\right)R_{ba}^{ct} \end{bmatrix}, \end{aligned} \tag{15}$$

for position and force control.

Thus, a Cartesian velocity and the force-profile F can be followed along selective axes. The presented controller differs from classic hybrid force-position control by the fact that disabling the force control along an axis does not directly result in position control. If $s_i^{vel} = s_i^{frc} = 0$, the robot automatically holds the current position according to the internal control loop and (13). Nonetheless, for a correct decoupling of the individual control policies, the selection matrices need to hold $s_i^{vel} s_i^{frc} = 0$. The final control architecture, as visualized in Figure 2, is well-suited for a graphical skill-formalism from Johannsmeier et al. (2019), as it can directly exploit hybrid policies along selective axes.

4.2 Applying Bayesian optimization with unknown constraints on graphical skill representations

Even though a skill graph can decrease the search space complexity, the resulting space may still suffer from the curse of dimensionality. Furthermore, collecting data from actual experiments is at risk of gathering various incomplete and, thus, useless data samples. In the context of episodic RL, one (successful) graph iteration represents a single episode. This requires all steps to succeed for a useful return value. Thus, we outline how the BOC problem from Section 2 can be reformulated to exploit available model knowledge in this skill-graph to improve sampling and learning. We assume that the feedback from (2) can be obtained at each node of the skill-graph and that each parameter in ξ is bounded. Given a graphical skill representation as in Figure 1, represented by MP nodes $\mathcal{V}$ and transitions $\mathcal{E}$, the objective $\mathcal{J}$ can be decomposed into the sum of all nodes, while the dedicated constraints need to be fulfilled at each step.

$$\begin{aligned} \mathcal{J}(\xi) &:= \sum_{v\in\mathcal{V}} \hat{\mathcal{J}}_v(\xi_v) \quad \xi_v \in \mathbb{R}^n, n \le m \\ &\text{s.t.} \quad g_v(\xi_v) \le c_v \quad \forall v \in \mathcal{V}. \end{aligned}. \tag{16}$$

Thus, (4) results in

$$\xi^* \leftarrow \arg\min_{\xi} \Lambda_{\mathcal{V}}^{\xi} \sum_{v\in\mathcal{V}} \mathbb{E}_{\hat{\mathcal{J}}_v}\left[\hat{\mathcal{J}}_v(\xi_v)\right], \tag{17}$$

where $\Lambda_{\mathcal{V}}^{\xi}$ denotes the joint success probability over all MP nodes. While preliminary work (Johannsmeier et al., 2019) has shown that the application of the summation in (17) improves learning speed and quality, we claim that it is,

furthermore, beneficial to exploit the structure of the MP-graph in order to regress $\Lambda_{\mathcal{V}}^{\xi}$ and thus design suitable acquisition functions from it. Due to the structure of the graph and the underlying BOC problem, the objective and success probability are conditionally independent. This allows outlining specific graph-based representations for the success probability of $\Lambda_{\mathcal{V}}^{\xi}$, which we outline below.

4.2.1 Naive Bayes approach

In order to approximate $\Lambda_{\mathcal{V}}^{\xi}$ from the underlying MP-graph, a commonly applicable solution is given as a naive Bayes approach, i.e., assuming conditional independence for all nodes. This is usually valid due to the condition-checking within the MP-graph from Section 3.1. Recalling the constraint in (16), the success probability of each MP node is subject to

$$\Gamma_{\mathrm{v}}^{\xi} = \prod_{j=1}^{|\boldsymbol{g}|} \begin{cases} \mathbb{P}\left[\hat{g}_{\mathrm{v}_j}(\boldsymbol{\xi}_{\mathrm{v}})\right] & \text{if } \texttt{active}(\boldsymbol{g}, j, \mathrm{v}) \\ 1 & \text{else} \end{cases}, \tag{18}$$

where $\texttt{active}(\boldsymbol{g}, j, \mathrm{v}) \mapsto \{\top, \bot\}$ encodes if the constraint is active in the current node or not. This allows directly encoding the structure of the graph—i.e., available task-knowledge—in the success probability of each node. Given the sequential structure of an MP-graph, the overall success probability results in the following:

$$\Lambda_{\mathcal{V}}^{\xi} = \prod_{\mathrm{v} \in \mathcal{V}} \Gamma_{\mathrm{v}}^{\xi}, \tag{19}$$

while the success probability of each intermediate node is obtained as the product of individual terms $\Gamma_{\mathrm{v}}^{\xi}$ from the initial to the current node. In order to estimate $\Lambda_{\mathcal{V}}^{\xi}$ from data, we thus regress each active success-constraint per node as an individual GP. These GPs are independent and use the success or failure as well as the constraint metric of the current subset of the meta parameter at each MP node. This results in at most $|\mathcal{V}||\boldsymbol{g}|$ GPs for the overall task. Nonetheless, the naive Bayes approach suffers from two disadvantages. First, the success function for the current node may depend on the full vector $\boldsymbol{\xi}$ instead of $\boldsymbol{\xi}_{\mathrm{v}}$ in (18). This contradicts the assumption of conditional independence and limits the applicability of the naive Bayes approach to tasks, where not only the task but also the constraints can be modeled individually for each MP. Given the structure of the MP-graph, namely, the existence of error constraints at each transition, the naive Bayes approach is still applicable to a broad variety of tasks but may not allow adding constraints that affect the choice of parameters across multiple MP nodes. Second, in case an episode fails at a dedicated node, no labels can be added to the subsequent nodes as each node is handled fully independently. While this still allows collecting samples earlier during the learning stage, the number of samples needed is expected to increase until successful samples can be obtained. In order to diminish these effects, we propose to model the success probability by a specialized factor graph in the next section.

4.2.2 Modeling the success function as a factor graph

In addition to the naive Bayes approach, it is also possible to directly impose the structural task knowledge that results from the graph structure. Namely, we propose to model the overall success probability as a factor graph representation Kschischang et al. (2001) for the task-constraints, where the scalar elements of $\boldsymbol{\xi}$ form the variables, and the constraints from (1) form the factors, cf. Figure 3.

Having obtained the general factor graph for a manipulation skill, this graph is fully described by an adjacency matrix $\boldsymbol{A}^{\mathrm{G}}$, where element $[\boldsymbol{A}^{\mathrm{G}}]_{(i,j)} \mapsto 1$ denotes an existing edge from i to j. Within factor graphs, an edge is only connecting a variable with a factor-node, such that it is sufficient to denote the adjacency matrix as $\boldsymbol{A}^{\mathrm{G}} \in \mathbb{R}^{|\boldsymbol{\xi}|\times|\boldsymbol{c}|}$. Therefore, columns denote individual constraints, and the rows define the subset of $\boldsymbol{\xi}$ for each individual constraint. Consequently, the success probability results in

$$\Lambda_{\mathcal{V}}^{\xi} = \prod_{j=0}^{|\boldsymbol{c}|} \min\left(\sum_{i=0}^{|\boldsymbol{\xi}|} [\boldsymbol{A}^{\mathrm{G}}]_{(i,j)}, 1\right) \mathbb{P}\left[\hat{g}^{c}\left(\boldsymbol{\xi}_{c_j}\right)\right] \quad \text{where } [\boldsymbol{A}^{\mathrm{G}}]_{(i,j)} \mapsto 1 \quad \forall \xi_i \in \boldsymbol{\xi}_{c_j}. \tag{20}$$

If, for example, only the active success-constraint per MP node is introduced and each constraint has the same input dimension, the naive Bayes approach is reconstructed. In contrast to the naive Bayes approach, each MP constraint can depend on arbitrary subsets of $\boldsymbol{\xi}$. In order to fully exploit the structure of the MP-graph, we propose to embed the underlying success probability for each vertex in the skill-graph. This can be directly achieved by extending the current set-values commanded to the robot system by an MP-specific adjacency matrix $\boldsymbol{A}^{\mathrm{G}}{}_i$. The success-constraint at each MP $\Gamma_{\mathrm{v}}^{\xi}$ can then be obtained by replacing $\boldsymbol{A}^{\mathrm{G}}$ with $\boldsymbol{A}^{\mathrm{G}}{}_i$ in (20). As a result, samples can be added to each constraint metric dependent on the current progress within the MP-graph. Thus, if a skill fails at a specific node, the samples obtained until the aforesaid notes can be added to the dataset as successful, while the samples for the failed node can be assigned to the current and subsequent MP success-estimators.

4.3 BOC model and acquisition function

Given the extended MP-graph, the objective of acquiring samples efficiently is again subject to the choice of the acquisition function and underlying GP model. Recalling Section 3.2, a key benefit of the method from Marco et al. (2021) is the ability to push the probability mass above the current threshold estimate, which allows gaining more knowledge from failed samples. Nonetheless, this model relies on approximating the posterior due to nonlinear components in (10). Instead, we propose to

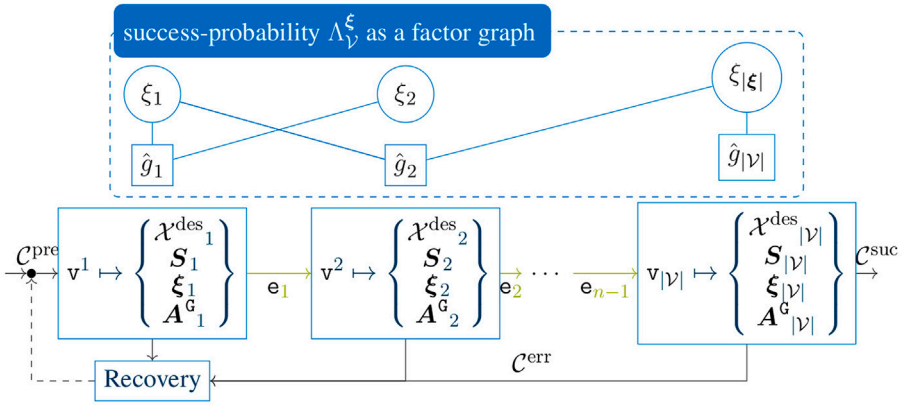

FIGURE 3
Graphical skill-formalism with an additional factor graph representation for the task success probability. The individual node parameters $\boldsymbol{\xi}_{\mathrm{v}}$ denote the meta parameters for each node, whereas $\mathcal{X}^{\text{des}}{}_{\mathrm{v}}$ denote the set-values and $\boldsymbol{S}_{\mathrm{v}}$ denotes the selection matrices $\boldsymbol{S}^{\text{frc}}{}_{\boldsymbol{R}}$ and $\boldsymbol{S}^{\text{vel}}{}_{\boldsymbol{R}}$ for the controller from Figure 2. The factor graph denotes the overall task completion probability, while the adjacency matrix $\boldsymbol{A}^{\mathcal{G}}{}_i$ for each vertex defines the *active* sub-graph for each vertex, which define the success probability for the current node $\Gamma^{\xi}_{\mathrm{v}}$. Again, the *Recovery* node intends to steer the robot to the initial robot state whenever an error occurs in order to initiate a new episode.

induce artificial data points and fit GPs on this artificial dataset instead. The algorithmic skeleton is sketched in Algorithm 1, where we again assume to have safe and failed data samples in the data buffer $\mathcal{D}$ for each constraint.

Algorithm 1. Induce artificial data points to fit GP on datasets with failed samples.

input : $\mathrm{p}_{\text{safe}}, \mathrm{p}_{\text{fail}}, \mathcal{D}, \upsilon, \kappa_{\text{spl}}, \zeta_{\text{spl}}$
output : $\hat{g}_i$

ConstraintFit:
- $\hat{g}^{\text{safe}} \leftarrow \texttt{ParameterFit}\left(\boldsymbol{\xi}_{\text{safe}}, \boldsymbol{g}_{\text{safe}}\right)$ ▷ fit valid constraint samples
- /* estimate threshold for safe GP */
- $\boldsymbol{\xi}_{\text{spl}} \sim \boldsymbol{\xi}_{\text{safe}} + \kappa_{\text{spl}} \mathcal{N}\left(\mathbf{0}^{|\boldsymbol{\xi}|}, \mathbf{1}^{|\boldsymbol{\xi}|\times|\boldsymbol{\xi}|}\right)$
- $\hat{\boldsymbol{c}}_i \leftarrow \Phi^{-1}(\mathrm{p}_{\text{safe}}) \sigma^{\text{safe}}_{\boldsymbol{g}_i}\left(\boldsymbol{\xi}_{\text{spl}}\right) + \max\left(\boldsymbol{g}_{\text{safe}}, \mu^{\text{safe}}_{\boldsymbol{g}_i}\left(\boldsymbol{\xi}_{\text{spl}}\right)\right)$
- $\hat{\boldsymbol{g}}^{\text{safe}}_{j,\text{fail}} \leftarrow \max\left(\Phi^{-1}(\mathrm{p}_{\text{fail}}), \zeta_{\text{spl}}\right) \sigma^{\text{safe}}_{\boldsymbol{g}_i}(\boldsymbol{\xi}_{\text{fail}})$ ▷ approximate failed data (safe GP)
- $\mathcal{D}_{\text{art}} \leftarrow \left\{\boldsymbol{\xi}_{\text{safe}}, \boldsymbol{g}_{\text{safe}} - \hat{\boldsymbol{c}}_i\right\} \cup \left\{\boldsymbol{\xi}_{j,\text{fail}}, \hat{\boldsymbol{g}}^{\text{safe}}_{j,\text{fail}}\right\}$ ▷ generate artificial data set
- $\hat{g}^{\text{fail}} \leftarrow \texttt{ParameterFit}\left(\mathcal{D}_{\text{art}}\right)$ ▷ fit artificial data set
- $\hat{\boldsymbol{g}}^{\text{fail}}_{i,\text{fail}} \leftarrow \max\left(\Phi^{-1}(\mathrm{p}_{\text{fail}}), \zeta_{\text{spl}}\right) \sigma^{\text{fail}}_{\boldsymbol{g}_i}(\boldsymbol{\xi}_{\text{fail}})$ ▷ approximate failed data (virtual GP)
- $\hat{\boldsymbol{g}}_i \leftarrow (1-\upsilon)\hat{\boldsymbol{g}}^{\text{safe}}_{j,\text{fail}} + \upsilon \hat{\boldsymbol{g}}^{\text{fail}}_{j,\text{fail}}$ ▷ Polyak average approxmiated data
- $\mathcal{D}_{\text{art}} \leftarrow \left\{\boldsymbol{\xi}_{\text{safe}}, \boldsymbol{g}_{\text{safe}} - \hat{\boldsymbol{c}}_i\right\} \cup \left\{\boldsymbol{\xi}_{i,\text{fail}}, \hat{\boldsymbol{g}}_i\right\}$ ▷ generate artificial data set
- $\hat{g}_i \leftarrow \texttt{ParameterFit}\left(\mathcal{D}_{\text{art}}\right)$ ▷ fit GP to artificial data set

We propose fitting a GP into the safe dataset first. Given this safe distribution, we propose to estimate the constraint value $\hat{\boldsymbol{c}}_i$. This can be achieved by evaluating the posterior at the safe input samples and applying the inverse CDF and a predefined probability threshold p_{safe} that should be held for legal samples. As the variance is usually small in the near distance of collected evidence, mean-free Gaussian noise is added on the existing samples. Taking the maximum of the predictive mean and the collected samples from the safe dataset, the predictive variance can be used to calculate the value of the constraint from the inverse CDF. Using the estimated constraint value $\hat{\boldsymbol{c}}_i$, the predictive variance at the infeasible data samples can be used to estimate the predictive mean value that would result in a posterior infeasibility probability threshold p_{fail}. In this estimation, we treat zero as the decision threshold for the current constraint GP and limit the inverse CDF to a lower bound $\zeta_{\text{spl}} \in \mathbb{R}^+$. As the safe GP does not contain any data sample within the unsafe parameter space, the variance of the posterior is expected to be large. Thus, we propose to apply a model-fit with the artificial dataset and repeat the aforementioned process to obtain new virtual output values. As the decision threshold is set to 0, the safe data samples are shifted by the current constraint estimate. Within our implementation, we also normalize the collected safe samples, but we omitted this in Algorithm 1 for brevity. The predictive posterior distribution of this artificial dataset will usually impose a conservative variance given the added data sample support. Thus, the final artificial value is obtained by a Polyak average of the two estimated posterior values. Given this, another parameter fit returns the final constraint GP. In order to embed ambiguity over unobserved parameters, the GPs use a zero-mean prior, which is equal to the constraint threshold of the virtual GP. In case there is no feasible dataset found, Algorithm 1 shortens. Instead of fitting existing data, the constraint is explicitly set to zero and the artificial data are set to $\min(\mathrm{p}_{\text{safe}}, \mathrm{p}_{\text{fail}})$. Eventually, a parameter fit is obtained to get an estimate of the constraint GP. For the naive Bayes approach, each MP and for the factor graph approach, each factor is finally realized by a constrained GP according to Algorithm 1. Given the structure of the factor graph and the dedicated skill, we propose to use a sequential form of (6). For the naive Bayes approach, this results in applying (6) at all nodes

$$\alpha_{\text{EIC},\mathcal{G}}(\boldsymbol{\xi}, \mathcal{D}_{\mathcal{G}}) = \Lambda^{\xi}_{\mathcal{V}} \sum_{\mathrm{v}\in\mathcal{V}} \mathbb{E}_{\mathrm{y}\sim\mathcal{N}_{\mathcal{J}_{\mathrm{v}}}(\mu,\sigma|\xi_{\mathrm{v}})}\left[\max\left(\mathrm{y} - \mathcal{J}^{\circledast}_{\mathrm{v}_{\mathcal{D}}}, 0\right)\Gamma^{\xi}_{\mathrm{v}}\right], \quad (21)$$

and weigh the sum of acquisition functions by the overall success probability to encourage acquisition of samples that are expected to succeed in the overall task. Due to the linear structure and the conditional independence of each node, $\Gamma_{\mathrm{v}}^{\xi}$ is directly obtained by $\hat{g}_i$ according to Algorithm 1, given that each node can be represented by a dedicated constraint metric. For the factor graph version, we do not assume conditional independence for the success-probabilities. Instead, the adjacency graph of each node is used to calculate $\Gamma_{\mathrm{v}}^{\xi}$ in (21)

$$\Gamma_{\mathrm{v}}^{\xi} = \prod_{i=1}^{|\boldsymbol{c}|} \begin{cases} \hat{g}_i & \text{if } \exists j \in [1, |\boldsymbol{\xi}|]\colon \left[\boldsymbol{A}^{\mathcal{G}}{}_{\mathrm{v}}\right]_{(\cdot,j)} \mapsto 1 \\ 1 & \text{else} \end{cases}, \tag{22}$$

using $\hat{g}_i$ according to Algorithm 1. Eventually, the best sample estimate is given by optimizing over the best guess of each MP-objective estimate at each MP and setting all samples with a success probability below $\mathrm{p}_{\mathrm{safe}}$ to $\mathcal{J}_{\mathcal{D}}^{\odot}$. Thus, the EI in (21) is replaced by the objective of each node, and $\Gamma_{\mathrm{v}}^{\xi}$ is set to 1 for feasible estimates. For infeasible samples, we assume the worst observed objective for the current objective estimate, such that the optimal parameter estimate is obtained as follows:

$$\boldsymbol{\xi}^{*} \leftarrow \arg\min_{\boldsymbol{\xi}} \sum_{\mathrm{v} \in \mathcal{V}} \begin{cases} \mathcal{J}_{\mathrm{v}}(\boldsymbol{\xi}_{\mathrm{v}}) & \text{if } \Gamma_{\mathrm{v}}^{\xi} \geq \mathrm{p}_{\mathrm{safe}} \\ \mathcal{J}_{\mathcal{D}}^{\odot} & \text{else} \end{cases}. \tag{23}$$

4.4 Exploit conditional dependencies for collected samples

Algorithm 2. Overall BOC algorithm.

input : $\mathrm{p}_{\mathrm{safe}}, \mathrm{p}_{\mathrm{fail}}, \mathcal{D}, \upsilon, \kappa_{\mathrm{spl}}, \zeta_{\mathrm{spl}}, N_{\mathrm{eps}}$
output : $\boldsymbol{\xi}^{*}$
1 $\mathcal{D}_{\mathcal{J}}^{\mathrm{v}} \leftarrow \emptyset, \mathcal{D}_{g}^{\mathrm{v}} \leftarrow \emptyset \ \forall \mathrm{v} \in \mathcal{V}$ ▷ init data sets
2 **for** $k = 1$ **to** N_{eps} **do**
3 $\quad \boldsymbol{\xi}_{\mathrm{spl}} \leftarrow \alpha_{\mathrm{EIC},\mathcal{G}}(\boldsymbol{\xi}, \mathcal{D}_{\mathcal{G}})$ ▷ cf. Section 4.3
4 $\quad \mathcal{J}_{\mathrm{spl}\mathcal{V}}, \boldsymbol{g}_{\mathrm{spl}}, \boldsymbol{s}_{\mathrm{spl}}, \mathrm{v}_i \leftarrow \mathtt{evaluate}(\boldsymbol{\xi}_{\mathrm{spl}})$ ▷ evaluate sample
/* assign environment feedback to data sets */
5 $\quad$ **for** $\mathrm{v} \in \mathcal{V}$ **do**
6 $\quad\quad \mathcal{D}_{g}^{\mathrm{v}} \leftarrow \mathtt{AddConstraint}\left(\mathcal{D}_{g}^{\mathrm{v}}, \boldsymbol{g}_{\mathrm{spl}}, \boldsymbol{s}_{\mathrm{spl}}, \mathrm{v}_i\right)$
7 $\quad\quad \mathcal{D}_{\mathcal{J}}^{\mathrm{v}} \leftarrow \mathtt{AddObjective}\left(\mathcal{D}_{\mathcal{J}}^{\mathrm{v}}, \mathcal{J}_{\mathrm{spl}\mathcal{V}}, \boldsymbol{s}_{\mathrm{spl}}, \mathrm{v}_i\right)$
8 $\quad\quad \mathcal{J}_{\mathrm{v}} \leftarrow \mathtt{ParameterFit}\left(\mathcal{D}_{\mathcal{J}}^{\mathrm{v}}\right)$ ▷ fit objective-GP
/* apply Algorithm 1 for all constraints */
9 $\quad$ **for** $k = 1$ **to** $|\boldsymbol{c}|$ **do**
10 $\quad\quad \hat{g}_i \leftarrow \mathtt{ConstraintFit}\left(\mathrm{p}_{\mathrm{safe}}, \mathrm{p}_{\mathrm{fail}}, \mathcal{D}, \upsilon, \kappa_{\mathrm{spl}}, \zeta_{\mathrm{spl}}\right)$
11 $\hat{\boldsymbol{\xi}}^{*} \leftarrow \arg\min_{\boldsymbol{\xi}} \sum_{\mathrm{v} \in \mathcal{V}} \hat{\mathcal{J}}_{\mathrm{v}}(\boldsymbol{\xi}_{\mathrm{v}}) \mathrm{H}\left(\Gamma_{\mathrm{v}}^{\xi}(\boldsymbol{\xi}_{\mathrm{v}}) \geq \mathrm{p}_{\mathrm{safe}}\right)$ ▷ get optimal parameter-guess

The final BOC algorithm for the proposed online RL approach is sketched in Algorithm 2. In contrast to learning the full parameterization of the task, the sequential skill-graph receives the additional feedback as to which node was explored last in Line 4. This information is crucial to assign samples correctly for the success- and constraint data buffers in Line 7 and Line 6. While the assignment for the MP node objectives is straightforward, i.e., only valid samples for explored nodes are assigned to the datasets, invalid samples may also be assigned even though the related MP or constraint factor has not yet been evaluated.

The necessary condition for a sample to be added to the dedicated dataset is that at least one scalar component has to be explored or visited. Due to the sequential procedure of the skill-graph, the mapping of the last explored node v_i to the dedicated datasets is deterministic and known beforehand. For the factor graph representation, it is further possible to add artificial samples to the dataset if a conditional dependence exists. If a subsequent node contains scalar components that have not yet been explored or visited, while other scalar components have been explored before a failure is detected, artificial data can be added to the dataset of the said constraint GP. Thus, the unexplored sample can be exchanged by drawing samples from a SoBol sequence (Sobol', 1967) or linearly distributed data. Using the adjacency matrices of the factor graph, the visited parameters can be obtained by $\mathrm{diag}\left(\boldsymbol{A}^{\mathcal{G}}{}_{\mathrm{v}_i}\boldsymbol{A}^{\mathcal{G}\intercal}{}_{\mathrm{v}_i}\right) \geq 1$ for each MP and thus, for the partially explored MP-graph as $\sum_{i=1}^{\mathrm{v}_i} \mathrm{diag}\left(\boldsymbol{A}^{\mathcal{G}}{}_{\mathrm{v}_i}\boldsymbol{A}^{\mathcal{G}\intercal}{}_{\mathrm{v}_i}\right) \geq 1$. Similarly, the samples that can be replaced by artificial samples are obtained as follows:

$$\left(\mathrm{diag}\left(\boldsymbol{A}^{\mathcal{G}}{}_{\mathrm{v}_i}\boldsymbol{A}^{\mathcal{G}\intercal}{}_{\mathrm{v}_i}\right) - \sum_{j=1}^{\mathrm{v}_i} \mathrm{diag}\left(\boldsymbol{A}^{\mathcal{G}}{}_{\mathrm{v}_j}\boldsymbol{A}^{\mathcal{G}\intercal}{}_{\mathrm{v}_j}\right)\right) \geq 1. \tag{24}$$

Before outlining an application example, we briefly outline the theoretical improvements of our approach, i.e., the scaling with respect to size of the meta parameter space.

4.5 Complexity analysis

In this section, we analyze the proposed method in terms of scaling with respect to size of the meta parameter space. It has to be noted that we do not emphasize improving GP scaling against big data, for which there is existing work (e.g., Ambikasaran et al. (2016)) available. For brevity, we denote the dimension of the original learning problem as n_{ξ}, i.e., $\boldsymbol{\xi} \in \mathbb{R}^{n_{\xi}}$ and denote the largest dimension of all nodes within a graphical skill-formalism as $m_{\xi,\mathcal{G}}$, i.e., $\boldsymbol{\xi}_i \in \mathbb{R}^{m_{\xi,\mathcal{G}}}$.

Definition 4.1. Valid MP-Graph

An MP-graph is a valid representation for (4), if the following constraints are given.

- *The graph has no absorbing nodes.*
- *There exists a finite path from the start-to the end-node.*
- *The underlying objective can be represented by a convex composition of sub-objectives.*

Definition 4.2. Feasible MP-graph

An MP-graph is a feasible representation for (4), if the following constraints are given.

- *The meta parameter space for each node of the MP-graph is bounded by* $m_{\xi,\mathcal{G}} < n_{\xi}$

- *The meta parameter space for all constraints is bounded by* $m_{\xi,\mathcal{G}} < n_\xi$
- *The number of active constraints per node is bounded by* $|\mathcal{C}|$

Claim 4.1

Regressing a general robot task 1) as a stochastic representation 4) *via* GPs according to Definition 4.2, the resulting complexity can be reduced from $\mathcal{O}(n_\xi{}^3)$ to $\mathcal{O}(\max(|\mathcal{C}|, 1)|\mathcal{V}|m_{\xi,\mathcal{G}}{}^3)$ by modeling the task as an MP-graph, using the naive Bayes approach.

Proof. Recalling (16), the objective function of the algorithm scales linearly with the numbers of nodes within the graph. According to Definition 4.2, the meta-parameter space of each MP node is bounded; thus,

$$\mathcal{O}(\mathcal{J}) = \mathcal{O}(|\mathcal{V}|m_{\xi,\mathcal{G}}{}^3). \tag{25}$$

This proves claim 4.1 if there are no success-constraints active, i.e., $|\mathcal{C}| = 0$. In case there is a success-constraint active, the upper bound of the complexity is defined by the complexity of the success probability as it may contain feasible and infeasible data samples. For the naive Bayes approach, $|\mathcal{C}|$ constraints have to be evaluated at $|\mathcal{V}|$ nodes *via* GPs, for which the meta parameter space is bounded by $m_{\xi,\mathcal{G}}$, thus resulting in an overall complexity of $\mathcal{O}(|\mathcal{V}||\mathcal{C}|m_{\xi,\mathcal{G}}{}^3)$.

Claim 4.2. Using the factor graph method and a task-representation as outlined in claim 4.1, the complexity from $\mathcal{O}(n_\xi{}^3)$ can be reduced to $\mathcal{O}(|\mathcal{C}|m_{\xi,\mathcal{G}}{}^3)$ if there are active constraints, i.e., $|\mathcal{C}| \geq 1$.

Proof. In contrast to the naive Bayes approach, the system complexity grows linearly with respect to the number of constraints $|\mathcal{C}|$. While the complexity follows (25), the success probability for $|\mathcal{C}| \leq 1$ and thus the overall system complexity results in $\mathcal{O}(|\mathcal{C}|m_{\xi,\mathcal{G}}{}^3)$.

Eventually, it has to be noted that adding artificial data adds data to the datasets of MP nodes or factors, which decreases scaling behavior. Nonetheless, it has to be noted that adding artificial data is not mandatory and intends to add support during early exploration when datasets are usually small. Therefore, we omitted the possibility of adding artificial data in the aforementioned complexity analysis.

5 Application example—screw insertion

In this section, we outline an application example for the proposed manipulation learning framework that uses the proposed controller from Section 4.1: the insertion of a screwdriver into a screwhead. Even though the environment suffers from high uncertainty, there exists available pre-knowledge that can be incorporated to reduce the problem size and thus use a skill-graph according to Section 4.2. While the previous sections have outlined the generic modalities of our method, this section intends to present an application example, which is eventually used to evaluate our approach. The main motivation of constructing a graphical skill-formalism is the reduction of the actual search space for the episodic RL task, i.e., the dimension of the parameter vector $\boldsymbol{\xi}$. Therefore, we assume the following constraints to be given:

- the screw is accessible by the robot end-effector, i.e., there exists a robot configuration that does not result in a self-collision of the robot with any surrounding object when the screwdriver is inserted. Furthermore, the robot configuration is singularity-free as this would not allow using the underlying Cartesian robot controller reliably.
- In case the position of the screwhead is subject to uncertainty, the condition above needs to be guaranteed for the full range of the uncertain region.[3]
- The robot is equipped with a screwdriver, and the transformation from the screwdriver pin, i.e., control frame $\mathfrak{ct}$, to the robot end-effector, i.e., ${}^{\mathfrak{ee}}\boldsymbol{T}_{\mathfrak{ct}}$ is known.
- The type of screw matches the pin of the screwdriver of the robot.

Given these assumptions, motion planning or pose optimization against infeasible states or collisions can be omitted. Instead, the framework focuses on finding a correct parameterization of the controller presented in Section 4.1. In approaches such as end-to-end learning, the problem could be represented as an RL-problem, with sparse rewards that penalize any constraint violations and add positive feedback for a successful task. While this allows to learn such a skill from visual data on arbitrary robot platforms, first, a supervised learning method is required to classify task success or constraint violation, and infeasible amount of data needs to be collected from experimental trial and error, where a supervised learning method is required, which will violate feasible time-budgets. In contrast, directly applying a GP policy would result in extremely large datasets, which will in return affect the evaluation or acquisition calculation. Thus, we propose to exploit the available expert knowledge and construct a skill-graph formalism similar to that of Johannsmeier et al. (2019). First, the normal vector $\boldsymbol{n}$ of the surface and the screw[4] is approximately known from visualization. Furthermore, we assume that an expert has set the desired contact wrench-magnitudes beforehand. Similarly, a designer has chosen a tilting angle for the robot end-effector to ease the contact tooling.

3 This condition also includes that a Cartesian path applied within this region will not result in a collision of the robot or a singularity since the actual control input is commanded directly in task- and not in joint space.

4 We set these normal vectors as constant within this evaluation, but it is possible to update the normal vectors online if needed.

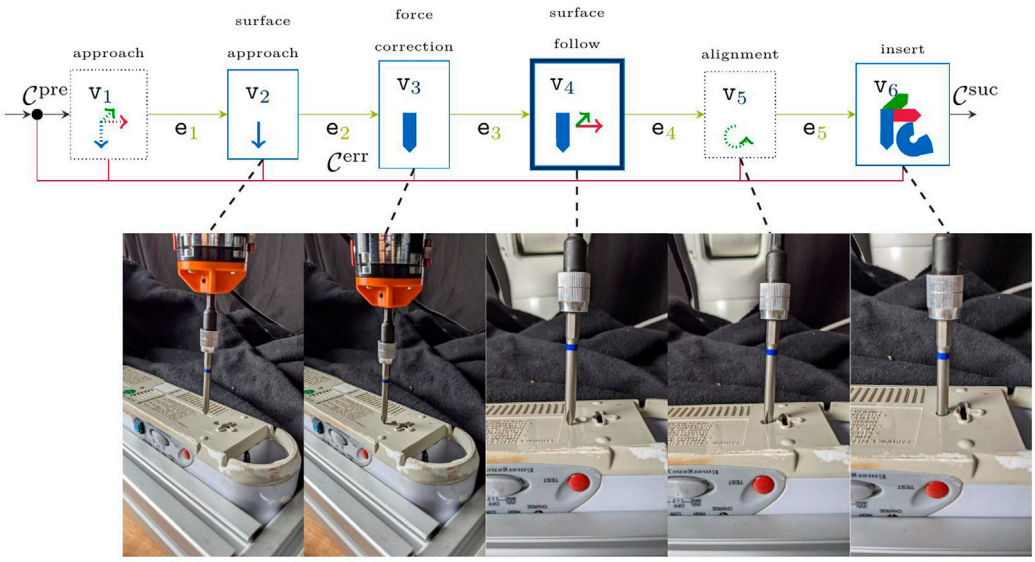

FIGURE 4
Schematic screw-insertion skill as an MP-graph. Each vertex shows the dedicated control direction, and thus the selection matrix from (14), where bold arrows represent force control and thin lines velocity control. Straight arrows in each MP denote translation with respect to axes ${}^{t}\boldsymbol{e}_x$, ${}^{t}\boldsymbol{e}_y$ and ${}^{t}\boldsymbol{e}_z$. Circular arrows denote a rotation along the dedicated axis. Parametric nodes are highlighted by a solid edge, whereas a bold edge denotes a hybrid control policy.

Given this, we outline the resulting skill-graph as visualized in Figure 4 from left to right. In this skill-graph, we explicitly denote the output alphabet, i.e., the desired set-values per node as well as the MP-parameters $\boldsymbol{\xi}_i$. For brevity, only non-zero values are explicitly mentioned, e.g., if not explicitly noted, all values of ${\boldsymbol{S}^{\text{vel}}}_{\boldsymbol{R}}$ and ${\boldsymbol{S}^{\text{frc}}}_{\boldsymbol{R}}$ are set to 0. The first node v_1 is non-parametric and describes the *approach*-MP, where the robot is asked to steer the tip of the tool and hover above the surface. As obtaining a suitable trajectory is beyond the scope of the presented method, we refer, e.g., to Bari et al. (2021) for further insights. The success of this node, thus advancing the graph to v_2, is evaluated *via*

$$\mathcal{C}^{\text{suc}}_{v_1}: = \|\boldsymbol{x}_{\text{des}} - \boldsymbol{x}_{\text{cur}}\|_2 \leq \zeta_{\text{pos}}. \tag{26}$$

The second node v_2—*approach-surface*—contains the first parametric node and describes the motion of the robot toward the surface until contact with the environment is established. Thus, a constant velocity along the negative surface-normal is applied, such that the set-values for the robot controller for this node are given as follows:

$$\begin{aligned} \mathcal{X}^{\text{des}} &= \{\dot{\boldsymbol{x}}_{\text{des}} \longleftarrow -\,\mathbb{s}_{\text{pos}} v_{\max} \boldsymbol{n}\} \\ \boldsymbol{\xi}_2 &= \mathbb{s}_{\text{pos}} \\ {\boldsymbol{S}^{\text{frc}}}_{\boldsymbol{R}} &= \text{diag}(0,0,1,0,0,0) \end{aligned}. \tag{27}$$

The success of this MP is given as an established contact with the environment, which is defined as follows:

$$\mathcal{C}^{\text{suc}}_{v_2}: = \mu_{\boldsymbol{F}} > \sigma_{\boldsymbol{F}}, \tag{28}$$

where the variance $\sigma_{\boldsymbol{F}}$ denotes the approximated sensor noise and $\mu_{\boldsymbol{F}}$ the filtered force–torque (FT) sensor readings over a sliding window of fixed size N_{FT}. This node further checks against the maximum allowed contact force $\boldsymbol{F}_{\max}$, as follows:

$$\mathcal{C}^{\text{err}}_{v_2}: = \left|\mu_{\boldsymbol{F}} - \sigma_{\boldsymbol{F}}\right| \geq |\boldsymbol{F}_{\max}|, \tag{29}$$

to raise a failure of the skill. The subsequent node v_3—*force correction*—corrects the encountered force impulse stemming from the contact at the end of the previous MP. Thus, the controller switches from the feed-forward velocity command to force-control along the normal vector of the surface:

$$\begin{aligned} \mathcal{X}^{\text{des}} &= \left\{\boldsymbol{F}_{\text{des}} \longleftarrow -\,\boldsymbol{F}_{\text{des}}\boldsymbol{n},\, [\boldsymbol{K}_P]_{(z)}\right\} \\ \boldsymbol{\xi}_3 &= [\boldsymbol{K}_P]_{(z)} \\ \boldsymbol{S}^{\boldsymbol{R}}_{\boldsymbol{F}_{\text{des}}} &= \text{diag}(0,0,1,0,0,0) \end{aligned}. \tag{30}$$

The success of this MP is evaluated by the accumulated force-error for a fixed window-size N_{cont}:

$$\mathcal{C}^{\text{suc}}_{v_3}: = \sum_{i=1}^{N_{\text{cont}}} \left(\boldsymbol{F}^{f}_{\text{cur},t-1} - \boldsymbol{F}_{\text{des}}\right)\boldsymbol{n} \leq \zeta_{\boldsymbol{F}}, \tag{31}$$

using only the force-measurement $\boldsymbol{F}^{f}_{\text{cur}}$ of the wrench $\boldsymbol{F}$. The error constraint not only checks against the force-threshold in (29) but also evaluates the following:

$$\mathcal{C}^{\text{err}}_{\text{v}_3} := \mathcal{C}^{\text{err}}_{\text{v}_2} \wedge \left(\mu_F - \sigma_F\right)^{\top} \left[\, \boldsymbol{n}, \boldsymbol{0}^3 \,\right] \geq 0.0, \tag{32}$$

to detect contact loss with the environment as an error constraint. During the next node v_4—surface search—the robot steers along the surface of the object in order to detect the screw. This implies a hybrid force–velocity profile, where the robot seeks to regulate the normal force with the surface, while following a velocity profile along the surface. Using a parameterized velocity profile $\dot{\boldsymbol{x}}_{\text{des},\kappa_{xy}}$, the output of v_4 is given as follows:

$$\begin{aligned} \mathcal{X}^{\text{des}} &= \left\{\dot{\boldsymbol{x}}_{\text{des}} \leftarrow \dot{\boldsymbol{x}}_{\text{des},\kappa_{xy}}, \boldsymbol{F}_{\text{des}} \leftarrow -\boldsymbol{F}_{\text{des}}\boldsymbol{n}, [\boldsymbol{K}_P]_{(z)}\right\} \\ \boldsymbol{\xi}_4 &= \left\{\kappa_{xy}, [\boldsymbol{K}_P]_{(z)}\right\} \\ \boldsymbol{S}^{\boldsymbol{R}}_{\boldsymbol{x}_{\text{des}}} &= \text{diag}\,(1,1,0,0,0,0) \\ \boldsymbol{S}^{\boldsymbol{R}}_{\boldsymbol{F}_{\text{des}}} &= \text{diag}\,(0,0,1,0,0,0) \end{aligned} . \tag{33}$$

The success of this MP is evaluated *via* the force impulse encountered in the current motion direction, i.e., $\frac{\dot{\boldsymbol{x}}_{\text{cur}}}{\|\dot{\boldsymbol{x}}_{\text{cur}}\|_2}$ and the perpendicular torque.

$$\mathcal{C}^{\text{suc}}_{\text{v}_4} := \left|\boldsymbol{F}^{f}_{\text{cur}} \frac{\dot{\boldsymbol{x}}_{\text{cur}}}{\|\dot{\boldsymbol{x}}_{\text{cur}}\|_2}\right| + \left|\boldsymbol{F}^{\tau}_{\text{cur}}\left(\boldsymbol{n} \times \frac{\dot{\boldsymbol{x}}_{\text{cur}}}{\|\dot{\boldsymbol{x}}_{\text{cur}}\|_2}\right)\right| \geq \zeta_{\text{impls}}. \tag{34}$$

For the error constraint, this node applies (29) and (32) and also checks against the robot position.

$$\mathcal{C}^{\text{err}}_{\text{v}_4} := \mathcal{C}^{\text{err}}_{\text{v}_2} \wedge \mathcal{C}^{\text{err}}_{\text{v}_3} \wedge \left\|\boldsymbol{p}_{\text{cur}} - \boldsymbol{p}_{\text{v}_3}\right\|_2 \geq \zeta_{\text{dspl}}, \tag{35}$$

where $\boldsymbol{p}_{\text{cur}}$ denotes the translational component of the tool-tip of the robot, whereas $\boldsymbol{p}_{\text{v}_3}$ represents the tool-tip position at the end of node v_3. The node v_5—*alignment*—is non-parametric and optional. It denotes the alignment of the tool-tip to be perpendicular to the surface. Thus, if the initial tilting angle is set to 0, this step is omitted. The success-constraint is identical to v_1, but the translation component is ignored. The final node v_6—*insert*—describes the insertion MP that applies a Cartesian wrench control. Thus, the MP is defined as follows:

$$\begin{aligned} \mathcal{X}^{\text{des}} &= \left\{\boldsymbol{F}_{\text{des}} \leftarrow -\boldsymbol{F}_{\text{des}}\boldsymbol{n}, \text{diag}\big([\boldsymbol{K}_P]_{(xy)}, [\boldsymbol{K}_P]_{(xy)}, [\boldsymbol{K}_P]_{(z)}, 0, 0, [\boldsymbol{K}_P]_{(\psi)},\big)\right\} \\ \boldsymbol{\xi}_6 &= \left\{[\boldsymbol{K}_P]_{(xy)}, [\boldsymbol{K}_P]_{(z)}, [\boldsymbol{K}_P]_{(\psi)}\right\} \\ \boldsymbol{S}^{\boldsymbol{R}}_{\boldsymbol{F}_{\text{des}}} &= \text{diag}\,(1,1,1,0,0,1) \end{aligned} . \tag{36}$$

While the error constraint is identical to v_3, that is, $\mathcal{C}^{\text{err}}_{\text{v}_6} := \mathcal{C}^{\text{err}}_{\text{v}_3}$, the success-constraint is checked *via* comparing the displacement along the normal vector

$$\begin{aligned} \mathcal{C}^{\text{suc}}_{\text{v}_6} := {} & \left\|\left(\boldsymbol{p}_{\text{cur}} - \boldsymbol{p}_{\text{v}_5}\right)\left(\boldsymbol{I}^3 - \boldsymbol{n}\right)\right\|_2 \geq \zeta_{\text{dspl}} \wedge \\ & \left(\left(\boldsymbol{p}_{\text{cur}} - \boldsymbol{p}_{\text{v}_5}\right)\boldsymbol{n}\right)^{\top}\left(\left(\boldsymbol{p}_{\text{cur}} - \boldsymbol{p}_{\text{v}_5}\right)\boldsymbol{n}\right) \geq \zeta_{\text{insrt,min}} \wedge, \\ & \left(\left(\boldsymbol{p}_{\text{cur}} - \boldsymbol{p}_{\text{v}_5}\right)\boldsymbol{n}\right)^{\top}\left(\left(\boldsymbol{p}_{\text{cur}} - \boldsymbol{p}_{\text{v}_5}\right)\boldsymbol{n}\right) \leq \zeta_{\text{insrt,max}} \end{aligned} \tag{37}$$

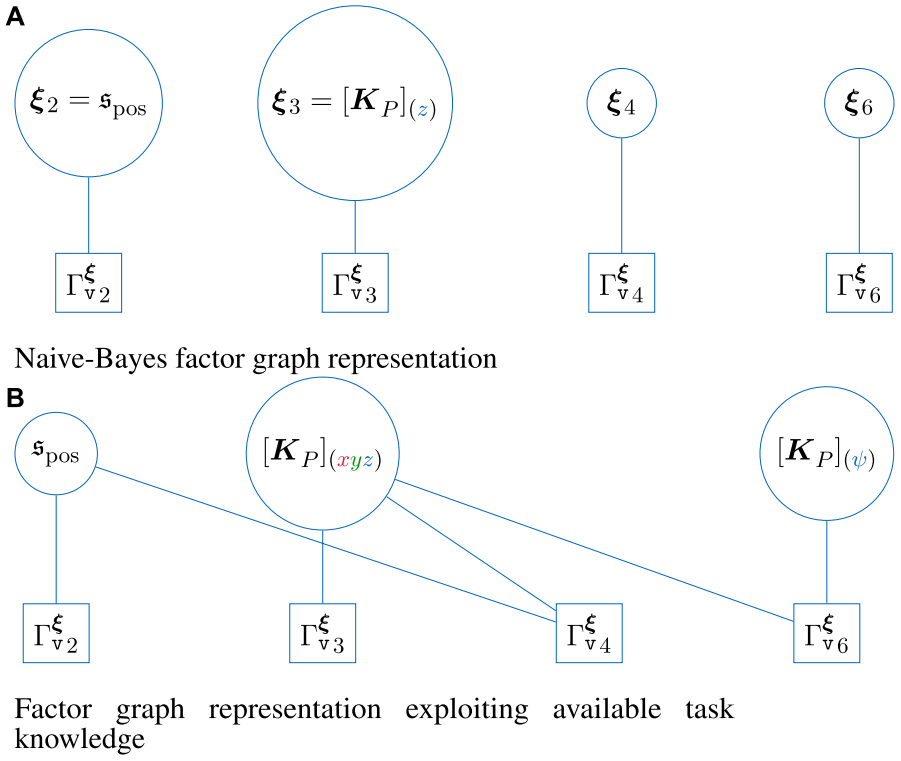

FIGURE 5
Representation of the success probability of the unscrewing skill as factor graphs. Here, the naive Bayes approach is also highlighted as a factor graph, while the actual factor graph exploits available task knowledge to introduced conditional dependence and independence in the regression problem that allows adding samples efficiently during learning.

TABLE 1 This table summarizes the unknown controller parameters for the unscrewing skill given the presented controller from Section 4.1.

Parameter	Lower bound	Upper bound
$\|\dot{\boldsymbol{x}}\|_2\,(\mathfrak{s}_{\text{pos}})$	1 mm s^{-1}	20 mm s^{-1}
$[\boldsymbol{K}_P]_{(\{xyz\})}$	1e-5	1e-3
$[\boldsymbol{K}_P]_{(\psi)}$	1e-5	1e-3

TABLE 2 Predefined parameters for the unscrewing skill. The value for $\frac{\dot{x}_{\text{des}}}{\|\dot{x}_{\text{des}}\|_2}$ denotes the motion direction that is to be followed during the search on the surface of this object.

Parameter	$\|\boldsymbol{F}_{\text{des,cont}}\|_2$	$\|\boldsymbol{F}_{\text{des,insrt}}\|_2$	$\boldsymbol{n}$	$\frac{\dot{\boldsymbol{x}}_{\text{des}}}{\|\dot{\boldsymbol{x}}_{\text{des}}\|_2}$
Value	10.0 N	30.0 N	$[0\ 0\ 1]^{\mathrm{T}}$	$[0.71\ \ 0.71\ \ 0]^{\mathrm{T}}$

where $\boldsymbol{p}_{\mathrm{v}_5}$ again denotes the tool-tip position when the current node is initiated.

Having introduced the general MP-graph, we now outline how the success-constraint of the overall skill can be derived as a factor graph for the outlined skill graph. First, the naive Bayes approach retrieves the success-constraint as the joint probability of

$$\Lambda^{\xi}_{\mathcal{V}} = \prod_{i=\{2,3,4,6\}} \Gamma^{\xi}_{\mathrm{v}}(\boldsymbol{\xi}_i). \tag{38}$$

This results in the factor graph from Figure 5a. For the factor graph representation, the actual parameter vector needs to be decomposed into the scalar components to obtain the underlying factors. Thus, this strongly depends on the actual parameterization of v_4 and v_6. As both nodes v_2 and v_3 are scalar parameters, we evaluate the presented approach by introducing two further simplifications:

- the search pattern on the surface of the object is restricted to a constant velocity, where the direction is set by an expert, while only the velocity needs to be adjusted to prevent the robot to miss the screw. Thus, we replace $\kappa_{xy} \leftarrow \mathfrak{s}_{\text{pos}}$.
- For the force controller, the proportional gain is set equally for all translational components x, y, and z.

As a result, the overall success probability results in the factor graph from Figure 5b.

The according adjacency matrices are then given as follows:

$$\begin{aligned} \boldsymbol{A}^{\mathrm{G}}{}_{\mathrm{v}_2} :&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{A}^{\mathrm{G}}{}_{\mathrm{v}_3} :&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ \boldsymbol{A}^{\mathrm{G}}{}_{\mathrm{v}_4} :&= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{A}^{\mathrm{G}}{}_{\mathrm{v}_6} :&= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{39}$$

Recalling Section 4.4, the graph structure needs to be respected when assigning samples. Failed trials at v_2 can be added to the failure of v_2 and v_4, while failures at v_3 and forward can be added to all nodes. In addition, the factor graph from Figure 5b allows creating artificial data samples for $[\boldsymbol{K}_P]_{(xyz)}$ in $\Gamma^{\xi}_{\mathrm{v}}4$ if a failure at v_2 is detected and similarly to generate samples for $[\boldsymbol{K}_P]_{(\psi)}$ in $\Gamma^{\xi}_{\mathrm{v}}6$ if a failure for v_3 or v_4 is encountered. Eventually, the RL-problem for the unscrewing task results in regressing the parameter vector $\boldsymbol{\xi} \in \mathbb{R}^3$, as well as $\boldsymbol{\xi}_4 \in \mathbb{R}^2$ and $\boldsymbol{\xi}_6 \in \mathbb{R}^2$ for the naive Bayes approach.Using the bounds from Table 1 a normalized parameter vector $\boldsymbol{\xi} \mapsto [0,\ 1]^3$ can be incrementally evaluated using the acquisition functions from Section 4.3 and existing work. We continue with comparing the improvements of our method against existing work in the next section.

6 Experimental results

Given the exemplary MP-graph for the unscrewing task from Figure 4, a suitable controller parameterization is regressed from data by setting the objective $\mathcal{J}$ as the negative overall runtime. A parameterization is set as successful if the full graph has been executed without indicating an error. In addition, each node can be repeated up to five times in case a timeout is encountered.The set-values chosen by a designer in our experimental recordings are listed in Table 2, where the insertion force is set higher than the environment contact force to enforce an insertion into the screwhead. In order to arrange for a fair comparison over the presented algorithms, the start pose has been chosen identically for all algorithms and the search direction is set to the static straight line on the object surface as shown in Figure 4. Similarly, the tilting angle is chosen to 2° for all approaches and is tilted perpendicular to the motion direction along the object surface. Furthermore, the constraint thresholds are set to $\zeta_{\text{pos}} = 0.1$ mm in translation and $\zeta_{\text{rot}} = 0.1$ rad in rotation. The variance of the FT sensor has been obtained before running the experiment from collected sensor data and evaluated to 0.3 N for the force-measurements and 0.2 N/m for the Cartesian torque-measurements. The window-size N_{FT} to evaluate the sensor readings has been chosen as 50 using a reading-rate of 170 Hz. Unfortunately, the presented force controller from Section 4.1 suffers from noisy sensor data and thus misses a proper damping term that could stabilize an aggressive proportional gain controller. To diminish the sensitivity to unstable controller behavior, an explored sample is set to failed if the standard-deviation of the observed force signal during contact is above 2.5 N using a sliding window of 1 s, with a sampling rate of 50 Hz, i.e., $N_{\text{cont}} = 50$ and $\zeta_{\boldsymbol{F}} = 0.25$ N. In order to detect a contact impulse during planar search, we set $\zeta_{\text{impls}} = 5.0$ N and allowed a maximum search range of $\zeta_{\text{dspl}} = 25.0$ mm. For the GP

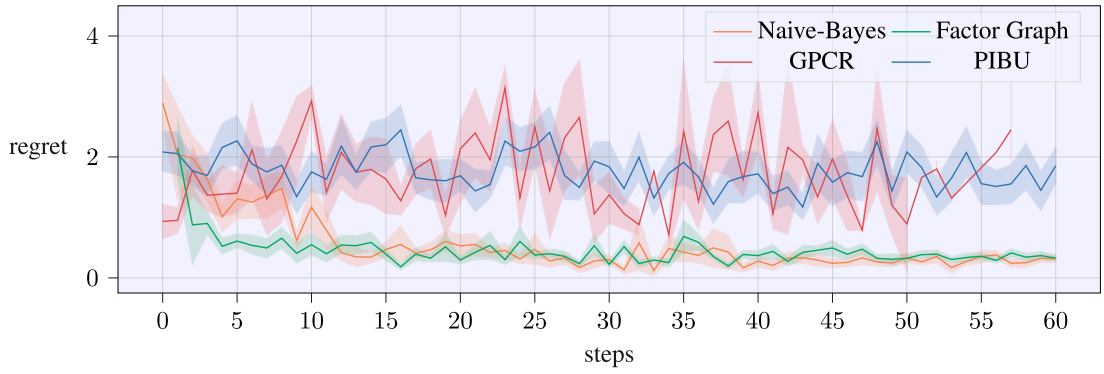

FIGURE 6
Regret evolution of the experimental screw-insertion task over the number of trials. The data are averaged over 25 runs per algorithm, with the shaded areas denoting a CI of 70 %.

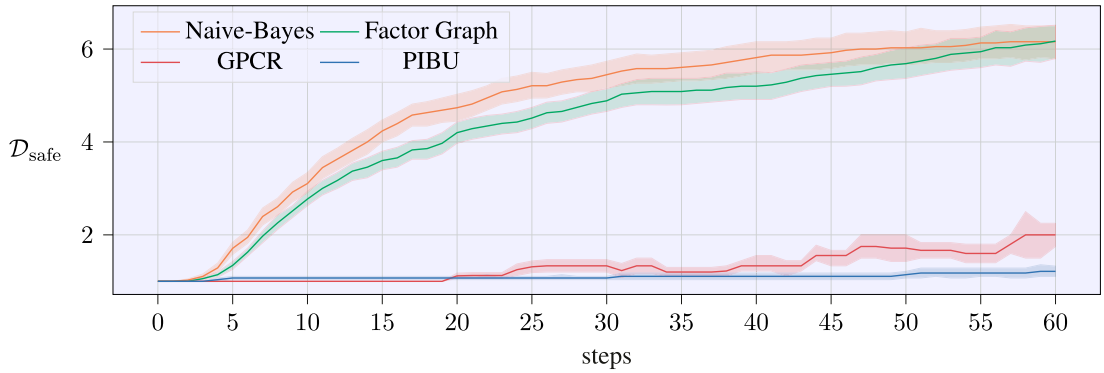

FIGURE 7
Number of safe samples for the experimental screw-insertion task over the number of trials. The data are averaged over 25 runs per algorithm, with the shaded areas denoting a CI of 70 %.

models, we assumed a zero-mean prior and used a Matern kernel $\frac{5}{2}$ assuming a prior gamma distribution with concentration of 3 and a rate of 6 for the length-scale and a concentration of 2 and a rate of 0.15 for the variance of the kernel. For the related work, we initialized their models according to their manuscripts (Englert and Toussaint, 2016; Marco et al., 2021). In order to allow for a fair comparison of the proposed algorithms and existing work, a grid search was recorded to collect empirical evidence database and mapped to a normalized hypercube of $\boldsymbol{\xi}$, given the parameter-bounds from Table 1.

Given this, each algorithm was run 25 times using $N_{\text{eps}} = 60$ iteration steps for each run. In each run new samples were added to the dedicated datasets, and the current optimum guess is stored at each step. Using the collected empirical evidence as ground-truth, the best empirical sample $\boldsymbol{\xi}^{\star} = [\,0.421 \;\; 0.316 \;\; 0.495\,]^{\mathrm{T}}$ is used to calculate the regret $\text{regret} = \mathcal{J}(\hat{\boldsymbol{\xi}}^{\star}) - \mathcal{J}^{\bigstar}$.

The averaged regrets over 25 trials per method are plotted in Figure 6, where the shaded area highlights the CI of 70%. The presented data underline that our graphical representations allow acquiring feasible data distinctly faster than GPCR and PIBU. This improved learning performance mainly stems from the decreased meta parameter space and the ability to collect evidence of the individual factors rather than learning the full task.

This is further underlined by the evolution of successful samples that are collected by the algorithms as visualized in Figure 7. Again, a CI of 70% is added over the averaged temporal evolution of the successful samples. It also has to be noted that this number is only increased if all nodes of the proposed graphical structures receive a successful sample, i.e., the

TABLE 3 Rate of estimating a correct optimal sample. The best performing, i.e., highest success-percentage is highlighted in bold.

	GPCR	PIBU	Naive Bayes	Factor graph
Success-probabilities (in %)	29.1	**53.7**	29.7	23.6

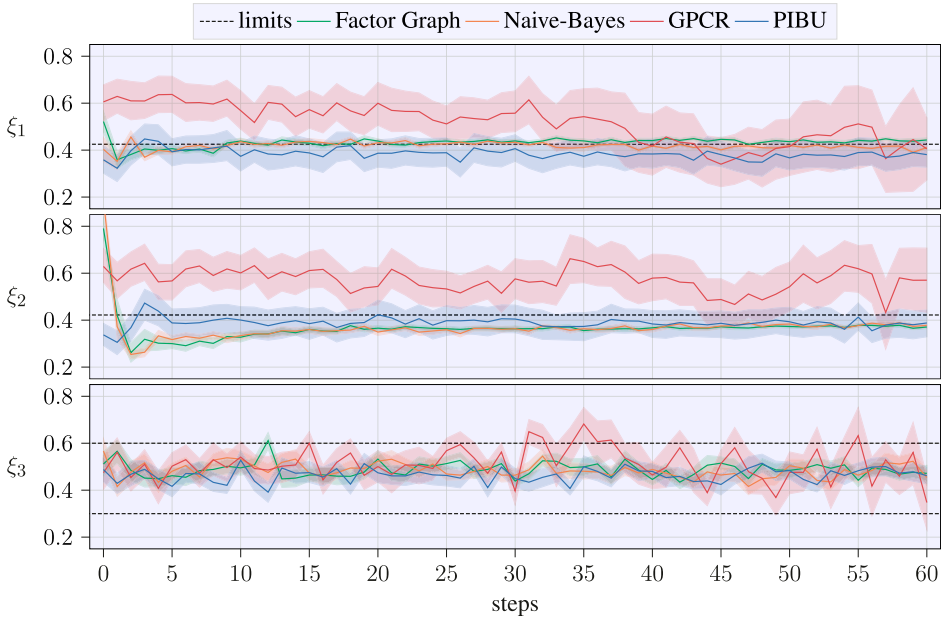

FIGURE 8
Temporal evolution of the estimated optimum parameter value, where the dashed lines denote the upper bound for the parameter, while the lower dashed line in the bottom plot denotes the dedicated lower bound. Shaded areas denote the CI of 70 %. The optimal values to be regressed from data are $\xi^* = [0.421 \;\; 0.316 \;\; 0.495]^T$.

overall exploration sample returns a successful sample. In this experiment, the naive Bayes approach could collect new successful samples earlier than the factor graph version. Nonetheless, the difference diminishes by the end of the 60 trials, and the evolution of successful samples equals out for both graphical approaches. Within our experimental evaluations, the GPCR method suffered from numerical instability after latest 60 iterations, while our approaches could evaluate further trials. As samples above 60 do not allow for a fair comparison, we omit the continuation of the plots. Still, we ran extended simulations for the proposed graphical methods with 80 steps, and the evolution of the successful samples converged to similar values for the final trial-episodes. While Figure 6 denotes the performance of the evaluated methods, Figure 7 denotes how many safe samples are explored. Nonetheless, Figure 6 only contains valid evaluations of the MPs or the tasks, as even if only a single MP fails, the regret would return an infinite value. In order to compare our algorithms in terms of safety awareness, the rates of estimating a valid optimal sample are listed for each algorithm in Table 3. As it can be seen, the pure GP classification within PIBU outperforms the remaining methods distinctly.

This effect mainly stems from the structure of the task, where the approaching speed scaling is linearly increasing the objective, while the constraint is given as a strict upper threshold, that also represents the optimal value. With only a handful samples, estimating the constraint rather than the classification labels remains numerically challenging.

In contrast, the application of a pure classification GP may also be overly conservative, and being only provided with a small number of successful samples, the classification may not be capable of returning a useful solution for the task to be learned. In addition to receiving a distinctly smaller regret, our approaches also converge closer to the actual optimum parameter samples. This is visualized by the temporal evolution of the estimated optimal parameter samples in Figure 8, where the shaded areas again denote a CI of 70%. In contrast to related work, our approaches quickly converge to a

solution for ξ_1 and ξ_2, while ξ_3 is only slowly converging toward the optimal value. This delay stems from ξ_3 being conditionally dependent on the performance of the remaining data samples. Even though the estimation of ξ_3 also suffers from higher variance than that of related work, our approaches distinctly outperform the related work in this aspect. This underlines that our approaches do not result in suitable parameter estimation by chance but due to efficient data acquisition.

6.1 Discussion

Having collected the experimental data, our approaches outperform existing work in terms of data efficiency and allow obtaining suitable results from only a handful of samples. Furthermore, our approaches apply standard GPs on smaller meta parameter spaces. Even though our regression method requires multiple parameter fits for multiple nodes, each GP is conditionally independent by definition within a factor graph. This allows for full parallelization, even though we evaluated our method in a purely sequential manner.

Nonetheless, the presented results also highlighted a particular downside of our method, which is exposed by the small chance of drawing successful samples. While our method outperformed existing work in drawing successful samples during exploration, this effect can be neglected during exploitation. If the current estimate is to be applied on safety-critical applications, the provided success rate needs to be improved. While it has to be noted that neither GPCR nor a classification GP can provide a safety guarantee when drawing success-estimate, the combination of our method with one of the former methods allows alleviating this issue. Thus, the overall graph success probability can be replaced by a product of experts, where the experts are given as the individual success-models. Another possible solution is given by using a negative prior mean similar to PIBU in the constraint GP and evaluating the constraint metric by shifting the probability of the posterior. Using this, the search of the optimal value is constrained to a tightened set of the parameter space, which automatically results in an increased probability to draw a correct sample.

Eventually, it has to be mentioned that the presented problem on regressing ξ_1 is a special case, while in general cases, where the optimum value is not in the near distance of the success-constraint, our method reliably converges to the correct parameter guess. Given the overall improvements of our method that is evident in the collected experimental data and the overall framework, it can be summarized that our method improves existing methods on regressing task-parameters for autonomous robots in a constraint-aware manner. Referring to the ability of converging to correct values within a reasonable time and amount of data, this makes our application a reasonable method to be applied on future robot platforms and manipulation tasks.

Finally, the question of whether either our factor graph or the naive Bayes approach is favorable needs discussion. Referring to the overall results, the performance of both methods is comparably similar. This mainly stems from the fact that the first parameter and thus the first sample is the most critical evaluation parameter of the task to be learnt. As this node is conditionally independent of the last parameter, the benefit of generating artificial data samples can only be applied rarely. Nonetheless, the preferable major advantage of the factor graph is given by the ability to apply it to arbitrary tasks and allows regressing constraints that have a different input space than the current objective node. Given that both approaches obtained almost identical performance results, the factor graph method forms the generic representation and preferable method, whereas the naive Bayes version is distinctive by its simplicity and simple adjustment to alternative models.

7 Conclusion

In this study, we proposed an episodic RL-scheme that uses BOC to account for unsafe exploration samples during learning. In order to apply the proposed scheme online, we further outlined a suitable control architecture for an industrial robot platform that uses a Cartesian displacement control interface at a comparably low update rate. The hybrid controller interface is well-suited to apply selective control strategies along individual axes, which can then be embedded into a graphical skill-formalism from previous work to reduce the required parameter space for the task to be learned.

In contrast to existing work, we further claimed that it is beneficial to not only exploit available task knowledge to decrease the parameter- or search space for the current task but also to incorporate task knowledge on regressing the failure constraints. For this reason, we proposed a graphical skill-formalism for the overall success probability as factor graphs. Here, we proposed a pure naive Bayes method that regresses the failure of the overall task as the joint probability of each node failing for a given sample. While this method improves the overall sampling, it may hinder assigning failed samples to subsequent nodes, even though conditional dependencies are well-known beforehand. Thus, we further proposed to incorporate these relations into a graphical skill-formalism for the success probability and thus improve scaling behavior to eventually regress feasible samples. In addition, we proposed suitable acquisition functions for the individual representations and proposed a novel conservative acquisition method.

Finally, we outlined an application example for the proposed method as the screw-insertion task for an industrial robot, where the exact goal-pose is unknown and the controller parameterization of our proposed controller needs to be regressed from data.

Given the outlined screw-insertion task, we compared our approaches against existing state-of-the-art methods for BOC-

based RL using an industrial robot manipulator in a laboratory environment. Given the collected experimental data, our method distinctly outperformed the state-of-the-art in performance, which we have evaluated by the collected objective regret. Furthermore, our method required distinctly smaller number of data samples and thus learning time and steps compared to existing work. These results underline that it is preferable to not only incorporate available task knowledge for the objective but also the constraints of robotic manipulation tasks during learning whenever possible in order to decrease the number of samples needed.

Future work

Building upon the data collected and the presented method, a promising path for future research projects lies in combining our method with visual feedback. This may further allow defining robust success- and error constraints, as, for example, missing the screwhead or hole remains unreliable solely from FT data, especially if a constant velocity vector results in a robot missing the screwhead completely. If such feedback is obtained, the presented method would strongly benefit in learning advanced motion policies, i.e., comparing different search patterns, e.g., spirals or straight-line patterns. Nonetheless, regressing the optimal search pattern usually is preferably solved by visual servoing. In these scenarios, the interaction does not rely on accurate FT data and feedback control. Thus, this allows collecting data within simulated environments and applying recent results from machine learning, especially meta-RL.

Eventually, future research should evaluate the possibility of self-evaluating models, i.e., artificial agents should be aware that some of the imposed model knowledge may be subject to false design. Thus, another line of research is given by designing new methods that allow not only exploiting available task knowledge but also evaluating the accuracy and discrepancy of the assumed model against the empirical evidence.

Data availability statement

The original contributions presented in this study are available at the repository https://gitlab.com/vg_tum/graph-boc. Further inquiries can be directed to the corresponding author.

Author contributions

VG proposed, implemented, and outlined the methods presented in the article, performed the experiments, and evaluated the collected evidence. VG and DW verified the approach. All authors discussed the results and contributed to the final manuscript.

Funding

The research, leading to the results presented in this work, has received funding from the Horizon 2020 research and innovation program under grant agreement №820742 of the project "HR-Recycler—Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment".

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Alt, B., Katic, D., Jäkel, R., Bozcuoglu, A. K., and Beetz, M. (2021). "Robot program parameter inference via differentiable shadow program inversion," in *IEEE international conference on robotics and automation (ICRA)* (Xi'an, China: IEEE), 4672–4678. doi:10.1109/ICRA48506.2021.9561206

Ambikasaran, S., Foreman-Mackey, D., Greengard, L., Hogg, D. W., and O'Neil, M. (2016). Fast direct methods for Gaussian processes. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 252–265. doi:10.1109/TPAMI.2015.2448083

Bari, S., Gabler, V., and Wollherr, D. (2021). "MS2MP: A min-sum message passing algorithm for motion planning," in *IEEE international conference on robotics and automation (ICRA)* (Xi'an, China: IEEE), 7887–7893. doi:10.1109/ICRA48506.2021.9561533

Baumann, D., Marco, A., Turchetta, M., and Trimpe, S. (2021). "Gosafe: Globally optimal safe robot learning," in *IEEE international conference on robotics and automation (ICRA)* (Xi'an, China: IEEE), 4452–4458. doi:10.1109/ICRA48506.2021.9560738

Beltran-Hernandez, C. C., Petit, D., Ramirez-Alpizar, I. G., Nishi, T., Kikuchi, S., Matsubara, T., et al. (2020). Learning force control for contact-rich manipulation tasks with rigid position-controlled robots. *IEEE Robot. Autom. Lett.* 5, 5709–5716. doi:10.1109/LRA.2020.3010739

Berkenkamp, F., Krause, A., and Schoellig, A. P. (2016a). Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics. *CoRR* abs/1602.04450.

Berkenkamp, F., Schoellig, A. P., and Krause, A. (2016b). "Safe controller optimization for quadrotors with Gaussian processes," in *IEEE international conference on robotics and automation (ICRA)*. Editors D. Kragic, A. Bicchi, and A. D. Luca (Stockholm, Sweden: IEEE), 491–496. doi:10.1109/ICRA.2016.7487170

Calandra, R., Seyfarth, A., Peters, J., and Deisenroth, M. P. (2016). Bayesian optimization for learning gaits under uncertainty - an experimental comparison on a dynamic bipedal walker. *Ann. Math. Artif. Intell.* 76, 5–23. doi:10.1007/s10472-015-9463-9

Cho, N. J., Lee, S. H., Kim, J. B., and Suh, I. H. (2020). Learning, improving, and generalizing motor skills for the peg-in-hole tasks based on imitation learning and self-learning. *Appl. Sci.* 10, 2719. doi:10.3390/app10082719

Craig, J. J., and Raibert, M. H. (1979). "A systematic method of hybrid position/force control of a manipulator," in The IEEE Computer Society's Third International Computer Software and Applications Conference, COMPSAC 1979, Chicago, Illinois, USA, 6-8 November, 1979 (Chicago, Illinois, United States: IEEE), 446–451. doi:10.1109/CMPSAC.1979.762539

Deisenroth, M. P., Fox, D., and Rasmussen, C. E. (2015). Gaussian processes for data-efficient learning in robotics and control. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 408–423. doi:10.1109/TPAMI.2013.218

Demir, S. O., Culha, U., Karacakol, A. C., Pena-Francesch, A., Trimpe, S., and Sitti, M. (2021). Task space adaptation via the learning of gait controllers of magnetic soft millirobots. *Int. J. Rob. Res.* 40, 1331–1351. doi:10.1177/02783649211021869

Deniša, M., Gams, A., Ude, A., and Petrič, T. (2016). Learning compliant movement primitives through demonstration and statistical generalization. *Ieee. ASME. Trans. Mechatron.* 21, 2581–2594. doi:10.1109/TMECH.2015.2510165

Devin, C., Gupta, A., Darrell, T., Abbeel, P., and Levine, S. (2017). "Learning modular neural network policies for multi-task and multi-robot transfer," in *IEEE international conference on robotics and automation (ICRA)* (Singapore: IEEE), 2169–2176. doi:10.1109/ICRA.2017.7989250

Drieß, D., Englert, P., and Toussaint, M. (2017). "Constrained bayesian optimization of combined interaction force/task space controllers for manipulations," in *IEEE international conference on robotics and automation (ICRA)* (Singapore: IEEE), 902–907. doi:10.1109/ICRA.2017.7989111

Englert, P., and Toussaint, M. (2016). "Combined optimization and reinforcement learning for manipulation skills," in *Robotics: Science and systems (RSS)*. Editors D. Hsu, N. M. Amato, S. Berman, and S. A. Jacobs (AnnArbor, Michigan) http://www.roboticsproceedings.org.

Englert, P., and Toussaint, M. (2018). Learning manipulation skills from a single demonstration. *Int. J. Rob. Res.* 37, 137–154. doi:10.1177/0278364917743795

Frans, K., Ho, J., Chen, X., Abbeel, P., and Schulman, J. (2018). "Meta learning shared hierarchies," in *International conference on learning representations (ICLR)*. Vancouver, BC, Canada: OpenReview.net.

Gullapalli, V., Franklin, J. A., and Benbrahim, H. (1994). Acquiring robot skills via reinforcement learning. *IEEE Control Syst. Mag.* 14, 13–24.

Gullapalli, V., Grupen, R. A., and Barto, A. G. (1992). "Learning reactive admittance control," in *IEEE international conference on robotics and automation (ICRA)* (Nice, France: IEEE Computer Society), 1475–1480. doi:10.1109/ROBOT.1992.220143

Gupta, A., Mendonca, R., Liu, Y., Abbeel, P., and Levine, S. (2018). "Meta-reinforcement learning of structured exploration strategies," in *Annual conference on neural information processing systems (NeurIPS)*. Editors S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Montreal, QC, Canada: Curran Associates, Inc.), 5307–5316.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., et al. (2018). *Soft actor-critic algorithms and applications*. *CoRR* abs/1812.05905.

Hamaya, M., Lee, R., Tanaka, K., von Drigalski, F., Nakashima, C., Shibata, Y., et al. (2020). "Learning robotic assembly tasks with lower dimensional systems by leveraging physical softness and environmental constraints," in *IEEE international conference on robotics and automation (ICRA)* (Paris, France: IEEE), 7747–7753. doi:10.1109/ICRA40945.2020.9197327

Inoue, T., Magistris, G. D., Munawar, A., Yokoya, T., and Tachibana, R. (2017). "Deep reinforcement learning for high precision assembly tasks," in *IEEE international workshop on intelligent robots and systems (IROS)* (Vancouver, BC, Canada: IEEE), 819–825. doi:10.1109/IROS.2017.8202244

Johannsmeier, L., Gerchow, M., and Haddadin, S. (2019). "A framework for robot manipulation: Skill-formalism, meta learning and adaptive control," in *IEEE international conference on robotics and automation (ICRA)* (Montreal, QC, Canada: IEEE), 5844–5850. doi:10.1109/ICRA.2019.8793542

Khatib, O., and Burdick, J. (1986). "Motion and force control of robot manipulators," in *IEEE international conference on robotics and automation (ICRA)* (San Francisco, CA, United States: IEEE), 1381–1386. doi:10.1109/ROBOT.1986.1087493

Khosravi, C., Khosravi, M., Maier, M., Smith, R. S., Rupenyan, A., and Lygeros, J. (2022). "Safety-aware cascade controller tuning using constrained bayesian optimization," in *IEEE Trans. Ind. Electron.*, 1. doi:10.1109/tie.2022.3158007

Kramberger, A., Gams, A., Nemec, B., Schou, C., Chrysostomou, D., Madsen, O., et al. (2016). "Transfer of contact skills to new environmental conditions," in *IEEE-RAS international workshop on humanoid robots (humanoids)* (Cancun, Mexico: IEEE), 668–675. doi:10.1109/HUMANOIDS.2016.7803346

Kschischang, F. R., Frey, B. J., and Loeliger, H. (2001). Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory* 47, 498–519. doi:10.1109/18.910572

LaGrassa, A., Lee, S., and Kroemer, O. (2020). "Learning skills to patch plans based on inaccurate models," in *IEEE international workshop on intelligent robots and systems (IROS)* (Las Vegas, NV, United States: IEEE), 9441–9448. doi:10.1109/IROS45743.2020.9341475

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* 17, 1–40.

Levine, S., Wagener, N., and Abbeel, P. (2015). "Learning contact-rich manipulation skills with guided policy search," in *IEEE international conference on robotics and automation (ICRA)* (Seattle, WA, United States: IEEE), 156–163. doi:10.1109/ICRA.2015.7138994

Li, Q., Kroemer, O., Su, Z., Veiga, F., Kaboli, M., and Ritter, H. J. (2020). A review of tactile information: Perception and action through touch. *IEEE Trans. Robot.* 36, 1619–1634. doi:10.1109/TRO.2020.3003230

Li, Q., Natale, L., Haschke, R., Cherubini, A., Ho, A. V., and Ritter, H. J. (2018a). Tactile sensing for manipulation. *Int. J. Hum. Robot.* 15, 1802001. doi:10.1142/S0219843618020012

Li, Y., Gowrishankar, G., Jarrassé, N., Haddadin, S., Albu-Schäffer, A., and Burdet, E. (2018b). Force, impedance, and trajectory learning for contact tooling and haptic identification. *IEEE Trans. Robot.* 34, 1170–1182. doi:10.1109/TRO.2018.2830405

Luo, J., Solowjow, E., Wen, C., Ojea, J. A., Agogino, A. M., Tamar, A., et al. (2019). "Reinforcement learning on variable impedance controller for high-precision robotic assembly," in *IEEE international conference on robotics and automation (ICRA)* (Montreal, QC, Canada: IEEE), 3080–3087. doi:10.1109/ICRA.2019.8793506

Marco, A., Baumann, D., Khadiv, M., Hennig, P., Righetti, L., and Trimpe, S. (2021). Robot learning with crash constraints. *IEEE Robot. Autom. Lett.* 6, 1439–1446. doi:10.1109/LRA.2021.3057055

Martín-Martín, R., Lee, M. A., Gardner, R., Savarese, S., Bohg, J., and Garg, A. (2019). "Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks," in *IEEE international workshop on intelligent robots and systems (IROS)* (Macau, SAR, China: IEEE), 1010–1017. doi:10.1109/IROS40897.2019.8968201

Mitsioni, I., Tajvar, P., Kragic, D., Tumova, J., and Pek, C. (2021). "Safe data-driven contact-rich manipulation," in *IEEE-RAS international workshop on humanoid robots (humanoids)* (Munich, Germany: IEEE), 120–127. doi:10.1109/HUMANOIDS47582.2021.9555680

Nau, D., Ghallab, M., and Traverso, P. (2004). *Automated planning: Theory & practice*. Elsevier.

Nemec, B., Abu-Dakka, F. J., Ridge, B., Ude, A., Jørgensen, J. A., Savarimuthu, T. R., et al. (2013). "Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile," in *IEEE international conference onn advanced robotics (ICAR)* (Montevideo, Uruguay: IEEE), 1–7. doi:10.1109/ICAR.2013.6766568

Petric, T., Gams, A., Colasanto, L., Ijspeert, A. J., and Ude, A. (2018). Accelerated sensorimotor learning of compliant movement primitives. *IEEE Trans. Robot.* 34, 1636–1642. doi:10.1109/TRO.2018.2861921

Rakicevic, N., and Kormushev, P. (2019). Active learning via informed search in movement parameter space for efficient robot task learning and transfer. *Auton. Robots* 43, 1917–1935. doi:10.1007/s10514-019-09842-7

Rasmussen, C. E., and Williams, C. K. I. (2006). Gaussian processes for machine learning. *Adaptive computation and machine learning*. Cambridge, United States: MIT Press.

Scherzinger, S., Roennau, A., and Dillmann, R. (2019a). "Contact skill imitation learning for robot-independent assembly programming," in *IEEE international workshop on intelligent robots and systems (IROS)* (Macau, SAR, China: IEEE), 4309–4316. doi:10.1109/IROS40897.2019.8967523

Scherzinger, S., Roennau, A., and Dillmann, R. (2019b). "Inverse kinematics with forward dynamics solvers for sampled motion tracking," in *International conference on advanced robotics (ICAR)* (Horizonte, Brazil: IEEE), 681–687. doi:10.1109/ICAR46387.2019.8981554

Scherzinger, S., Rönnau, A., and Dillmann, R. (2017). "Forward dynamics compliance control (FDCC): A new approach to cartesian compliance for robotic manipulators," in *IEEE international workshop on intelligent robots and systems (IROS)* (Vancouver, BC, Canada: IEEE), 4568–4575. doi:10.1109/IROS.2017.8206325

Sobol', I. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Comput. Math. Math. Phys.* 7, 86–112. doi:10.1016/0041-5553(67)90144-9

Stenger, D., Nitsch, M., and Abel, D. (2022). "Joint constrained bayesian optimization of planning, guidance, control, and state estimation of an autonomous underwater vehicle," in 2022 European Control Conference (ECC), London, United Kingdom, 12-15 July 2022. *CoRR* abs/2205.14669. doi:10.48550/arXiv.2205.14669

Stolt, A., Carlson, F. B., Ardakani, M. M. G., Lundberg, I., Robertsson, A., and Johansson, R. (2015). "Sensorless friction-compensated passive lead-through programming for industrial robots," in *IEEE international workshop on intelligent robots and systems (IROS)* (IEEE), 3530–3537. doi:10.1109/IROS.2015.7353870

Stolt, A., Linderoth, M., Robertsson, A., and Johansson, R. (2012). "Force controlled robotic assembly without a force sensor," in *IEEE international conference on robotics and automation (ICRA)* (Hamburg, Germany: IEEE), 1538–1543. doi:10.1109/ICRA.2012.6224837

Sui, Y., Gotovos, A., Burdick, J. W., and Krause, A. (2015). "Safe exploration for optimization with Gaussian processes," in *International conference on machine learning (ICML)*. Editors F. R. Bach and D. M. Blei (Lille, France: JMLR Workshop and Conference Proceedings), 997–1005.

Vanderborght, B., Albu-Schäffer, A., Bicchi, A., Burdet, E., Caldwell, D. G., Carloni, R., et al. (2013). Variable impedance actuators: A review. *Robotics Aut. Syst.* 61, 1601–1614. doi:10.1016/j.robot.2013.06.009

Wang, Z., Garrett, C. R., Kaelbling, L. P., and Lozano-Pérez, T. (2021). Learning compositional models of robot skills for task and motion planning. *Int. J. Rob. Res.* 40, 866–894. doi:10.1177/02783649211004615

Yang, L., Li, Z., Zeng, J., and Sreenath, K. (2022). "Bayesian optimization meets hybrid zero dynamics: Safe parameter learning for bipedal locomotion control," in *IEEE international conference on robotics and automation (ICRA)* (Philadelphia, PA, United States: IEEE), 10456–10462. doi:10.1109/ICRA46639.2022.9812154

Zhang, X., Sun, L., Kuang, Z., and Tomizuka, M. (2021). Learning variable impedance control via inverse reinforcement learning for force-related tasks. *IEEE Robot. Autom. Lett.* 6, 2225–2232. doi:10.1109/LRA.2021.3061374

Nomenclature

Acronyms

BO Bayesian optimization

BOC Bayesian optimization with unknown constraints

CDF Cumulative distribution function

CI Confidence-interval

EI Expected improvement

EIC Expected improvement with constraints

EP Expectation propagation

FSA Finite-state automaton

FT Force–torque

GP Gaussian process

GPCR Gaussian process for classified regression

ML Machine learning

MP Manipulation primitive

NN Neural network

PDF Probability density function

PI Probability of improvement

PIBU Probability of improvement with a boundary uncertainty criterion

RL Reinforcement learning

List of indices

art Artificial variable

cont Contact with object or environment

cur Current value, e.g., measured state of a plant

des Desired value, e.g., a desired trajectory

dspl Displacement related variable, e.g., a maximum distance

eps Current variable is related to current or all episodes

err Error-term for current value

fail Failed trial/sample

frc Force/wrench-related variable

impls Impulse variable, e.g., force-impulse during contact

insrt Insertion related variable, e.g., time needed for a screwdriver insertion

max Maximum value of the current variable

min Minimum value of the current variable

noise Noise-related variable, may be systematic or artificially injected noise

pos Position-related variable or term

pre Pre-condition, e.g., within a planning domain

$\mathfrak{ba}$ Reference frame—robot base frame

$\mathfrak{ct}$ Reference frame—control frame

$\mathfrak{ee}$ Reference frame—end-effector frame

$\boldsymbol{R}$ Rotated variableRotation matrix in SO(3), i.e., in $\mathbb{R}^{3\times3}$

rot Rotation-related variable or term

safe Safe variable with respect to a constraint metric

spl Sampled version of the current variable

suc Indicating success for the current task or episode

vel Velocity-related variable

List of operators and functions

$\boldsymbol{\alpha}$ Acquisition function to generate new data samples in $\mathbb{R}^n$

$\boldsymbol{g}$ Scalar inequality constraint in $\mathbb{R}^1$Inequality constraint vector in $\mathbb{R}^n$

$\boldsymbol{g}$ Scalar inequality constraint in $\mathbb{R}^1$Inequality constraint vector in $\mathbb{R}^n$

diag Get diagonal elements from a matrix as vector in $\mathbb{R}^n$

H Heaviside function in $\mathbb{R}^1$, i.e., $\mathrm{H}(\mathfrak{p}) = 1 \Leftrightarrow \mathfrak{p} > 0$

$\Lambda_{\mathcal{V}}^{\xi}$ joint success probability for a sequential manipulation task

$\boldsymbol{k}$ Kernel function $\boldsymbol{k}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)$ in $\mathbb{R}^1\boldsymbol{k}$ applied on a batch of samples p, and $\boldsymbol{\xi}$, such that $[\boldsymbol{k}_p(\boldsymbol{\xi})]_{(i)} = \boldsymbol{k}(\mathfrak{p}_i, \boldsymbol{\xi})$

$\hat{\mathcal{J}}$ model of the actual objective function $\mathcal{J}$ in $\mathbb{R}^1$

$\hat{g}$ model of a success function, i.e., a binary function mapping $\mathbb{R}^n \mapsto \top, \bot$

$\mathcal{J}$ general objective function for an optimization problem in $\mathbb{R}^1$

$\Gamma_{\mathrm{v}}^{\xi}$ success probability of an MP node in $\mathbb{R}^1$

Notation

$\mathfrak{p}$ Placeholder variable in notationVector in $\mathbb{R}^n$

$\top$ Boolean *true*Boolean *true*

$\bot$ Boolean *false*Boolean *false*

$\top$ Boolean *true*Boolean *true*

$\bot$ Boolean *false*Boolean *false*

$|\mathfrak{p}|$ Cardinality of a set or dimension of a vector, i.e., n for $\mathfrak{p} \in \mathbb{R}^n$

$\mathfrak{p}^{\circledast}$ Best observed data samples from collected experience data

$\mathfrak{p}^{\ominus}$ Worst observed data samples from collected experience data

$:=$ Equal by definition

$\mathfrak{p}^{\star}$ Ground-truth data within a regression problem, where the correct data assignment is known

$\boldsymbol{\kappa}$ Hyper-parameter; indexing defines actual meaning

$\mathfrak{p}_t$ Value of $\mathfrak{p}$ at time t

$\mathfrak{p}_i$ Content indexing of vectors, lists, sets, e.g., $\mathfrak{p}_2$ denotes the second value of $p\mathrm{L}_i$-norm, where i is usually 1 (sum of absolutes), 2 (euclidean) or ∞ (maximum value)

$\boldsymbol{1}^{\mathfrak{p}\times\mathfrak{p}}$ Identity matrix of dimension $\mathfrak{p}\times\mathfrak{p}$

$\boldsymbol{0}^{\mathfrak{p}\times\mathfrak{p}}$ Zero matrix of dimension $\mathfrak{p}\times\mathfrak{p}$

$\mathfrak{P}$ Matrix in $\mathbb{R}^{m\times n}$

$[\mathfrak{P}]_{(i,j)}$ Matrix element, usually $\mathbb{R}^1$

$\|\mathfrak{p}\|_i$ Content indexing of vectors, lists, sets, e.g., $\mathfrak{p}_2$ denotes the second value of $p\mathrm{L}_i$-norm, where i is usually 1 (sum of absolutes), 2 (euclidean) or ∞ (maximum value)

$\mathfrak{p}^\star$ Optimal or true value of $\mathfrak{p}$

$\mathbb{R}$ Rational numbers

$\mathbb{R}^+$ Positive non-negative rational numbers

$\boxtimes$ Empty set

$\hat{\mathfrak{p}}$ Estimated value of $\mathfrak{p}$

$\mathfrak{p}'$ Temporal successor of $\mathfrak{p}$, i.e., in a discrete setting $\mathfrak{p}' = \mathfrak{p}_{t+1}$

ζ Threshold-value; indexing defines actual meaning

t Current time or temporal indexing variable

$T_{\mathbf{max}}$ Maximum runtime (continuous) or number of time steps (discrete) in $\mathbb{R}^1$

$\vec{\mathfrak{p}}$ Trajectory as a sequence of T variables $\mathfrak{p}$ in $T\times\mathbb{R}^n$

p Placeholder variable in notationVector in $\mathbb{R}^n$

$\boldsymbol{1}^{\mathfrak{p}}$ Identity vector of dimension $\mathfrak{p}$

$\boldsymbol{0}^{\mathfrak{p}}$ Zero vector of dimension $\mathfrak{p}$

List of symbols

$\boldsymbol{A}^{\mathtt{G}}$ Adjacency matrix of a graph $\mathtt{G}$, where $[a]_{(i,j)} = 1 \Leftrightarrow \exists \mathtt{e}_{i,j} \in \mathcal{E}$

$\boldsymbol{\delta_x}$ Cartesian displacement in SE(3), i.e., in $\mathbb{R}^1$

$\boldsymbol{x}$ Cartesian pose of and object or the end-effector in SE(3)Cartesian x-coordinate in $\mathbb{R}^1$. If not phrased explicitly, ${}^{\mathfrak{b}\mathfrak{a}}x$ is assumed

$\boldsymbol{p}$ Cartesian position in $\mathbb{R}^3$

$\boldsymbol{F}$ Cartesian wrench as force–torque measures in SE(3)

$\mathtt{O}$ Algorithmic complexity (*Big O*—convention)

$\boldsymbol{c}$ Scalar constraint value in $\mathbb{R}^1$Constraint value as a vector in $\mathbb{R}^n$

$\boldsymbol{c}$ Scalar constraint value in $\mathbb{R}^1$Constraint value as a vector in $\mathbb{R}^n$

$\mathcal{C}$ Set of constraints

$\boldsymbol{u}$ Control input signal

$\boldsymbol{e}$ SE(3) Coordinate system axis in $\mathbb{R}^3$, where norm $2\boldsymbol{e} = 1$ holds

$\boldsymbol{x}$ Cartesian pose of and object or the end-effector in SE(3)Cartesian x-coordinate in $\mathbb{R}^1$. If not phrased explicitly, ${}^{\mathfrak{b}\mathfrak{a}}x$ is assumed

$\boldsymbol{y}$ Cartesian y-coordinate in $\mathbb{R}^1$. If not phrased explicitly, ${}^{\mathfrak{b}\mathfrak{a}}y$ is assumed

$\boldsymbol{z}$ Cartesian z-coordinate in $\mathbb{R}^1$. If not phrased explicitly, ${}^{\mathfrak{b}\mathfrak{a}}z$ is assumed

$\boldsymbol{e}_x$ SE(3) Coordinate system x-axis in $\mathbb{R}^3$

$\boldsymbol{e}_y$ SE(3) Coordinate system y-axis in $\mathbb{R}^3$

$\boldsymbol{e}_z$ SE(3) Coordinate system z-axis in $\mathbb{R}^3$

$\boldsymbol{\Sigma}$ Covariance matrix of a multivariate probability density function (PDF)

$\mathcal{D}$ Data buffer containing experiences usable for RL

$\mathfrak{X}$ Observed data samples

$\mathcal{D}_{\mathbf{art}}$ Data buffer with artificially generated data samples

$\mathfrak{p}_{\mathbf{fail}}$ Desired probability threshold for posterior of failed failures to be infeasible

$\mathfrak{p}_{\mathbf{safe}}$ Desired probability threshold for posterior of safe failures to be safe

$\mathcal{X}^{\mathrm{des}}$ Desired state space or sub-space of the state space $\mathcal{X}$

$\boldsymbol{f}$ Force magnitude or scalar force component of translational component of $\boldsymbol{F}$ in $\mathbb{R}^1$

$\mathcal{X}$ Fully observable state space

$\boldsymbol{K}_P$ Proportional force controller gain matrix (quadratic, positive semi-definite)

$\mathtt{y}$ Evaluation of GP while applying an acquisition function

$\mathtt{G}$ Arbitrary graph consisting of vertices $\mathcal{V}$ and edges $\mathcal{E}$

$\mathtt{e}$ Specific edge of the edges-set $\mathcal{E}$ of a graph $\mathtt{G}$

$\mathcal{E}$ Edge-set of a graph $\mathtt{G}$

$\mathtt{v}$ Specific vertex of the vertex-set $\mathcal{V}$ of a graph $\mathtt{G}$

$\mathcal{V}$ Vertex-set of a graph $\mathtt{G}$

$\boldsymbol{K}$ Gram matrix, where $[\boldsymbol{K}]_{(i,j)} = \boldsymbol{k}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)$

$\boldsymbol{k}$ Kernel function $\boldsymbol{k}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)$ in $\mathbb{R}^1\boldsymbol{k}$ applied on a batch of samples p, and $\boldsymbol{\xi}$, such that $[\boldsymbol{k}_p(\boldsymbol{\xi})]_{(i)} = \boldsymbol{k}(\mathfrak{p}_i, \boldsymbol{\xi})$

$\boldsymbol{\mu}$ Mean of a PDF

$\boldsymbol{\Phi}$ Normal cumulative distribution function

$\boldsymbol{n}$ Normal vector of a surface/object in Cartesian space in $\mathbb{R}^3$

$\boldsymbol{N}_{\mathbf{cont}}$ Number of evaluation measurements to check against environment contact

$\boldsymbol{N}_{\mathbf{spl}}$ Number of samples

$\boldsymbol{N}_{\mathbf{eps}}$ Number of steps, e.g., within an episode

$\boldsymbol{\xi}$ Unknown meta parameter vector in $\mathbb{R}^n$, obtained by means of episodic RLUnknown scalar meta parameter in $\mathbb{R}^1$, $\xi \in \boldsymbol{\xi}$

$\boldsymbol{\xi}$ Unknown meta parameter vector in $\mathbb{R}^n$, obtained by means of episodic RLUnknown scalar meta parameter in $\mathbb{R}^1$, $\xi \in \boldsymbol{\xi}$

$\boldsymbol{v}$ Polyak-averaging weight, e.g., used to update target network

$\boldsymbol{S}$ Hybrid force/position controller selection matrix

$\boldsymbol{s}$ Diagonal element of the force/position controller selection matrix $\boldsymbol{S}$

$\boldsymbol{R}$ Rotated variableRotation matrix in SO(3), i.e., in $\mathbb{R}^{3\times3}$

$\boldsymbol{R}_{\varphi}$ Rotation matrix around $\boldsymbol{e}_x$ in $\mathbb{R}^{3\times3}$

$\boldsymbol{R}_{\theta}$ Rotation matrix around $\boldsymbol{e}_y$ in $\mathbb{R}^{3\times3}$

$\boldsymbol{R}_{\psi}$ Rotation matrix around $\boldsymbol{e}_z$ in $\mathbb{R}^{3\times3}$

$\boldsymbol{\psi}$ Yaw angle in $\mathbb{R}^1$, i.e., angular rotation around $\boldsymbol{e}_z$

$\boldsymbol{g}_{\text{spl}}$ Episodic constraint-vector sample

$\mathcal{J}_{\mathbf{spl}}$ Episodic objective sample

$\boldsymbol{s}_{\text{spl}}$ Episodic success sample, where each scalar evaluates $g_i(\boldsymbol{\xi}) \leq c_i$

$\mathfrak{s}$ Scaling term $\mathfrak{s} \in [01]$State value in a general state space

$m_{\xi,\mathfrak{G}}$ Largest dimension all nodes within a graphical skill-formalism to reduce the state space of a regression problem $\boldsymbol{\xi}_i \in \mathbb{R}^{m_{\xi,\mathfrak{G}}}$

$\boldsymbol{n}_{\boldsymbol{\xi}}$ Dimension of the parameter of a regression problem $\boldsymbol{\xi} \in \mathbb{R}^{n_{\xi}}$

$\mathfrak{s}$ Scaling term $\mathfrak{s} \in [01]$State value in a general state space

$\boldsymbol{\delta}_{\boldsymbol{t}}$ Update time step for discrete control processes in $\mathbb{R}^1$

$\boldsymbol{\tau}$ Torque magnitude or scalar force component of rotational component of $\boldsymbol{F}$ in $\mathbb{R}^1$

$\boldsymbol{T}$ Coordinate transformation matrix using homogeneous transformation in SE(3), i.e., in $\mathbb{R}^{4\times4}$

$\boldsymbol{v}$ Translational velocity in SE(3)

$\boldsymbol{\sigma}$ Variance of a one-dimensional PDF

$\boldsymbol{N}_{\mathbf{FT}}$ Size of sliding window to evaluate data obtained from a FT sensor