Constrained Bayesian Optimization of Combined Interaction
Force/Task Space Controllers for Manipulations

Danny Drieß  Peter Englert  Marc Toussaint

Abstract—In this paper, we address the problem of how a robot can optimize parameters of combined interaction force/task space controllers under a success constraint in an active way. To enable the robot to explore its environment robustly, safely and without the risk of damaging anything, suitable control concepts have to be developed that enable compliant and force control in situations that are afflicted with high uncertainties. Instances of such concepts are impedance, operational space or hybrid control. However, the parameters of these controllers have to be tuned precisely in order to achieve reasonable performance, which is inherently challenging, as often no sufficient model of the environment is available. To overcome this, we propose to use constrained Bayesian optimization to enable the robot to tune its controller parameters autonomously. Unlike other controller tuning methods, this method allows us to include a success constraint into the optimization. Further, we introduce novel performance measures for compliant, force controlled robots. In real world experiments we show that our approach is able to optimize the parameters for a task that consists of establishing and maintaining contact between the robot and the environment efficiently and successfully.

I. INTRODUCTION

The vision that robots leave caged industry halls to collaborate safely with humans in unstructured environments is highly appreciated. Recent research [1], [2] has shown that exploiting contacts between the robot and the environment plays an important role here, because it reduces the uncertainty about the state of the world and hence is able to improve the robustness during the manipulation. However, seeking contacts can be dangerous, especially in uncertain and dynamically changing environments. Therefore, the new generation of autonomous robots requires suitable control concepts that are capable of interacting with unknown objects robustly and without the risk of damaging anything. Stiff joint space control, designed for the precise execution of preplanned trajectories, cannot achieve this. Instead, compliant control frameworks like operational space or impedance control are much more appropriate in this situation, where the focus is on establishing a dynamic interaction between the robot and environmental constraints. Such frameworks realize force and variable stiffness control, which increases the safeness and enables the robot to exploit contacts.

However, the performance of those interaction control concepts is subject to parameters which heavily depend on the physical properties like stiffness, mass or shape of the objects in the environment. This contradicts the idea of using those control frameworks in uncertain environments where usually no model of the objects is available a priori.

In an autonomous robot way of thinking, we suggest that the robot should be enabled to tune its controller parameters by itself. Formally, this can be seen as an optimization problem, where the goal is to minimize a certain cost function that measures the performance of the controller. However, standard optimization algorithms cannot be applied here, as the dependency of the cost function on the controller parameters is unknown. Moreover, for manipulation tasks, often discrete success constraints are involved that indicate whether the manipulation was successful or has failed. We think that combining real valued cost terms and discrete success indicators into one single objective function is unfavorable and discards the structure of the problem. Recently, Englert et al. [3] have introduced a method that augments Bayesian optimization, a black-box (unconstrained) optimization algorithm suitable for analytically unknown cost functions, to include a discrete success constraint.

In the present work, we propose to combine this special constrained Bayesian optimization with a class of interaction control frameworks to tune the controller parameters for manipulations that are afflicted with uncertainties. As an important step, we consider how the performance of such controllers in the context of compliant and force controlled interactions can be measured, since standard criteria like deviation from reference trajectories are not suitable to evaluate manipulations focusing on interaction. Our presented controller framework enables to control in multiple weighted task spaces with varying stiffness/damping properties, as well as to limit interaction forces simultaneously. In contrast to existing controller tuning approaches, by including a constraint into the optimization process, we can ensure that the optimized parameters lead to successful manipulations.

Our paper is organized as follows. Section II reviews related work dealing with automatic tuning of controller parameters and interaction control concepts. An overview of our approach is given in section III. Section IV, V and VI give details about our methodology, the controller framework, the evaluation criteria and the constrained Bayesian optimization algorithm. Finally, our approach is demonstrated in real world experiments, see section VII.

II. RELATED WORK

A. Control frameworks for interaction

So-called impedance [4] and operational space [5] control are both suitable concepts for tasks that require interaction
between the robot and the environment. Their goal is to
directly specify the desired behavior in the operational/task
space in terms of virtual mass-spring-damper systems with
variable stiffness and damping properties. This allows to
realize compliance in different directions, making these
control concepts suitable for interaction tasks. More recent
frameworks [6], [7], [8] are based on constraint task de-
scriptions. In [8], a whole body control concept is proposed
that establishes a hierarchy between control objectives by
prioritizing tasks handling constraints over operational tasks.
In order to increase the safeness during the interaction,
it is desirable to regulate contact forces. Villani et al. [9] give
a comprehensive overview of different force control
concepts like hybrid control, where the idea is to decompose
the task space into position and force controlled, orthogonal
subspaces. Force control can either be realized by displace-
ments or directly on the force level.

Estimating reasonable parameters for these compli-
ant/force controllers can be challenging [9], especially if
little knowledge about the structural properties (stiffness,
mass etc.) of the objects in the environment is known.
This emphasizes the necessity of our method to enable the
robot to tune its parameters autonomously in unmodeled
environments.

B. Learning control and (safe, constrained) Bayesian op-
timization in robotics

Learning/tuning controllers has a great history in robotics.
Often, this consists of identifying dynamics models [10],
[11]. Although system identification is also important for our
controller, it is not suitable to enable the robot to optimize its
interaction controller parameters in uncertain environments.

Marco et al. [12] applied (unconstrained) Bayesian optimi-
zation to tune the weights of the cost functional of a linear-
quadratic regulator. They demonstrated this tuning process on
balancing an inverted pendulum. In contrast to our approach,
they do not distinguish between unstable controllers and
successful ones, instead, they assign high costs to failures,
which, we think, discards the structure of the problem and
has disadvantages. The work of Schreiter et al. [13] deals on
how safety constraints can be included in an active learning
process. Their focus lies on safe exploration, i.e. they try to
avoid sampling of parameters that lead to system failures,
as they consider them to be dangerous. Berkenkamp et al. [14]
adohere to a similar spirit. They optimize the controller gains
for a quadrocopter with Bayesian optimization augmented
by safety constraints. They consider safety constraints like a
certain maximum deviation from a reference trajectory.

The key difference to our approach is that we are interested in
successful parameters, that is, we want to find parameters
that not only minimize some costs, but simultaneously lead
to task success, since manipulation tasks often involve such
discrete success signals. Moreover, we think that optimizing
parameters in situations that are inherently afflicted with
uncertainties like robots interacting with the environment
is much more necessary than to tune well-defined systems
like quadrocopters, where the tuning mainly consists of
compensating imprecise dynamics models.

III. OVERVIEW: CONSTRAINED BAYESIAN OPTIMIZATION
OF INTERACTION FORCE/TASK SPACE CONTROLLERS

An overview of our approach is visualized in Fig. 1. We
utilize a combined interaction force/task space controller
(green box) for manipulation tasks, whose parameters \( \xi \in \mathcal{P} \subset \mathbb{R}^p \) should be tuned. The set \( \mathcal{P} \) represents a priori
feasible parameters, e.g. the controller stability range. Then,
we define a cost function \( \mathcal{J} \) that measures the performance
of the controller. Furthermore, a binary success indicator \( S \) is
specified that determines whether an execution was success-
ful or not (for example, manipulation success). Every specific
choice of a parameter \( \xi \) yields a certain cost \( \mathcal{J}(\xi) \in \mathbb{R} \) and
success \( S(\xi) \in \{0, 1\} \), observable only, when executed in
real world. The parameter tuning process therefore consists of
the constrained optimization problem

\[
\min_{\xi \in \mathcal{P}} \mathcal{J}(\xi) \quad \text{subject to} \quad S(\xi) = 1. \quad (1)
\]

Unfortunately, standard methods to solve (1) are not appli-
cable, because no structural information like gradients about
\( \mathcal{J} \) and \( S \) is available. Instead, only (probably noisy) samples
of \( \mathcal{J} \) and \( S \) can be drawn for specific parameter choices
\( \xi \). In order to overcome this, we utilize the method from
Englert et al. [3]. This constrained Bayesian optimization
(blue box) iteratively selects parameters \( \xi_i \) that should be
tested on the real robot in order to improve its performance,
while trying to be successful. The outcome of each trial is
collected in a dataset \( \mathcal{D} = \{(\xi_i, J_i, S_i)\}_{i=1}^w \) with \( J_i = \mathcal{J}(\xi_i) \) and \( S_i = S(\xi_i) \). Based upon this dataset, we learn
two Gaussian processes (GP), one regression \( g_J \) that is a
surrogate model for the cost function \( \mathcal{J} \) and a classifier \( g_S \)
that represents the parameter region leading to success.
The information encoded in the two GPs is combined in a so-
called acquisition function \( a \), that decides which parameter
\( \xi_{w+1} \) should be rolled out next. The decision is made based
upon the goal of both exploring the success region without
sampling many failures and improving the parameter inside
the feasible region.

Our purposed combined force/task space controller allows
to specify the desired behavior of the robot in multiple task spaces. This includes task references, variable stiffness and damping properties. Simultaneously, interaction forces in these task spaces can be limited. The force controller is designed in a way that no switching in control laws is required, i.e. the same controller can be used for the free and constrained motion. The controller consists of two nested loops. In an outer loop (100 Hz), the computational complex quantities for realizing the task space behaviors are calculated by solving an optimal control problem. The solution is then projected to the inner loop (realtime 1 kHz), a joint space PD/limit force controller.

IV. Combined Interaction Force/Task Space Controller

The control concept is an important ingredient for the vision of autonomous robots that exploit contacts during manipulations. We consider rigid body manipulators with \( n \) degrees of freedom, whose dynamics can be modeled by

\[
u = M(q)\ddot{q} + F(q, \dot{q}),
\]

with the mass/inertia matrix \( M(q) \in \mathbb{R}^{n \times n} \), the vector \( F(q, \dot{q}) \in \mathbb{R}^n \) representing Coriolis, centripetal and gravity forces, \( \dot{u} \in \mathbb{R}^n \) the motor commands and \( q, \dot{q}, \ddot{q} \in \mathbb{R}^n \) denote the joint state, velocity and acceleration of the robot.

A. Task Space Controller

We describe the desired behavior of the robot in terms of task maps \( \phi : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^d \), \( \dot{y} = \phi(q) \) which are (differentiable) functions from the robot configuration \( q \) to a \( d \)-dim. space. Common task maps are the (3D) position or orientation of the end-effector of the robot. But also other tasks like distances to objects, collisions, joint limits etc. are possible. The idea behind task space control is to realize a mass-spring-damper system in the task space, specified by positive definite stiffness \( K_p \in \mathbb{R}^{d \times d} \) and damping \( K_d \in \mathbb{R}^{d \times d} \) matrices that realize a PD behavior around the desired references \( y^{\text{ref}}, \dot{y}^{\text{ref}}, \ddot{y}^{\text{ref}} \in \mathbb{R}^d \) in that task space. More formally, in \( m \) task spaces \( \phi_i : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^{d_i} \), \( i = 1, \ldots, m \), we want to impose certain acceleration laws

\[
\ddot{y}_i = \ddot{y}^{\text{ref}}_i + \dot{K}_p (y_i^{\text{ref}} - y_i) + K_d (\dot{y}_i^{\text{ref}} - \dot{y}_i). \tag{3}
\]

With suitable tuned gain matrices \( K_p, \) and \( K_d, \) these accelerations accomplish the desired references asymptotically. Given these task space PD laws, the sake of the task space controller is to generate motor commands \( u \) such that the robot behaves like these virtual mass-spring-damper-systems. Peters et al. [6] have found out that a variety of control laws suitable for this goal can be derived from one unifying methodology, namely by minimizing squared motor commands while requiring to achieve a task perfectly. This is realized by formulating (3) into a constraint after differentiating \( \phi \) with respect to time twice, i.e. \( J_\phi \dot{q} + \ddot{J}_\phi \dot{q} = \ddot{y}^{\text{ref}}_i \) with the Jacobian \( J_\phi \) of \( \phi_i \). In the formulation of [6], only one task \((m = 1)\) is considered and a constrained optimization problem is solved. In contrast, our task space controller solves for multiple tasks in every of its loop cycles the unconstrained optimization problem

\[
\begin{align*}
\min_{\dot{q}} & \quad \| M\ddot{q} + F - u_0 \|_H^2 + \sum_{i=1}^m \| J_\phi \ddot{y}^{\text{ref}}_i + \dot{J}_\phi \dot{q} \|_{C_i}^2 \tag{4}
\end{align*}
\]

by relaxing the constraints to become a squared penalty. Similar to [6], the motor commands can be weighted with the positive definite \( H \in \mathbb{R}^{n \times n} \). A reasonable choice is \( H = M^{-1} \) [6]. The advantage of our optimal control framework is that it is not only singularity robust, but also allows to control in multiple, maybe interfering task spaces. In addition, these task spaces can be weighted against each other with the positive definite matrices \( C_i \in \mathbb{R}^{d_i \times d_i} \). Those with higher eigenvalues correspond to more important tasks. In the limit \( C \rightarrow \infty \) in any norm, this controller converges for one task to the same framework presented in [6]. For multiple tasks, taking limits \( C_i \rightarrow \infty \) individually, the squared penalties are transformed into hierarchical hard-constraints. With \( u_0 \in \mathbb{R}^n \) a nullcost control law can be realized, for example to stabilize in the joint space or to not penalize gravity compensation costs \((u_0 = F)\). The solution of (4), given by \( \ddot{q}^* = A^{-1} (MH (u_0 - F) + \sum_{i=1}^m J^T_\phi C_i (\ddot{y}_i^{\text{ref}} - \dot{J}_\phi \dot{q}) \) with the symmetric positive definite generalized inverse \( A = MH + \sum_{i=1}^m J^T_\phi C_i J_\phi \), can be translated to motor signals via (2). If more complex task maps are involved, this control law is not suitable to be executed in realtime. Therefore, we linearize this controller at the actual joint configuration \( q_0 \) with \( y_i \approx \phi_i (q_0) + J_\phi_i (q - q_0) \). This yields an affine control law

\[
u^* = u_0 - \ddot{K}_p q - \dot{K}_d \dot{q}
\]

in terms of the joint state with the projected stiffness/damping matrices

\[
\begin{align*}
\ddot{K}_p = M A^{-1} \sum_{i=1}^m J^T_\phi C_i K_p & \in \mathbb{R}^{n \times n} \tag{6} \\
\dot{K}_d = M A^{-1} \sum_{i=1}^m J^T_\phi C_i \left( K_d J_\phi + J_\phi \right) & \in \mathbb{R}^{n \times n} \tag{7}
\end{align*}
\]

and the bias vector \( u_0 \in \mathbb{R}^n \) with

\[
u_0 = F + MA^{-1} \left[ MH (u_0 - F) + \sum_{i=1}^m J^T_\phi C_i \left( \ddot{y}_i^{\text{ref}} + K_p (y_i^{\text{ref}} - \phi_i (q_0) + J_\phi_i (q_0) + K_d \ddot{y}_i^{\text{ref}}) \right) \right] \tag{8}
\]

The outer loop calculates (8), (6), (7), sends these to the inner loop, where (5) is executed and commanded to the motors.

B. Limit Force Controller

In addition to the task space controller, we would like to regulate interaction forces in the task space \( \phi \) to a desired reference \( f^{\text{ref}} \in \mathbb{R}^d \). The force/torque measurements \( f_{it} \in \mathbb{R}^6 \) of the sensors located at the wrists of our PR2 can be projected to the actual force \( f \) in the task space via a pseudo-inverse of the Jacobian from the sensor frame to the task space. Many force control concepts [9] realize the force reference by the feed-forward term \( u_f = J^T_\phi f^{\text{ref}} \)
with an additional control law to compensate external disturbances. However, such kind of force control concepts suffer from the problem that different controllers are necessary for the free motion, where no force should be controlled, and the constrained motion that generates the interaction force. Unfortunately, this is not favorable in unstructured environments, where it is difficult to estimate the moment of contact when the control laws should be switched. Moreover, switching controllers are known to raise serious stability issues, especially if the contact is lost unforeseen. To overcome this, we propose to use a limit force controller that intervenes only, if the measured forces are higher than the references. We realize this by a discounted integral term $e^{\text{ft}} \in \mathbb{R}^d$, that is initialized by $\mathbf{0}$ and updated in every loop with

$$e^{\text{ft}}_j \leftarrow \gamma_j e^{\text{ft}}_j + [|f_j| > |f^\text{ref}_j|] (f^\text{ref}_j - f_j) \quad \forall j = 1, \ldots, d. \quad (9)$$

The Iverson bracket $[\cdot]$ is has value one, if the measured force is higher than the reference, i.e. the force error is updated in this case only, otherwise, it is zero. The discounting factor $\gamma \in (0, 1]^d$ ensures that the force integral term vanishes exponentially, if no force control is necessary anymore. The applied motor commands for the force controller are

$$u^*_j = J^T_{\phi} \alpha e^{\text{ft}} = \tilde{K}_d e^{\text{ft}}. \quad (10)$$

With the positive definite matrix $\alpha \in \mathbb{R}^{d \times d}$, an exponential decay in the (limit) force error is realized. Adding (10) to (5) leads to our combined interaction force/task space controller, which has the advantage that it does not require switching control laws. Very similar to hybrid control [9], we parameterize the tasks such that the force control objective does not interfere with other, position controlled subspaces. This can be done in our framework by assigning a zero eigenvalue of the task space stiffness matrices in the direction of the force controller. To actually realize a certain force reference $f^\text{ref}$ in the task space $\phi$, we set a velocity reference $y^\text{ref}$ towards the environmental constraint. By proper tuning of the velocity gain $K_d$ and the force controller coefficients $\alpha$, $\gamma$, the force reference is realized, as the constant velocity towards the constraint increases the force, depending on $K_d$ and $y^\text{ref}$, while the limit force controller intervenes to stabilize the force reference. Doing this, simultaneous force and position control is accomplished. Since our controller allows to weight tasks according to their importance, non strict decompositions in position/force controlled subspaces are also possible.

V. EVALUATION CRITERIA FOR MANIPULATIONS

In order to optimize controller parameters, it is necessary to define proper criteria to evaluate and distinguish controllers based on their performance. Standard ways of measuring this are cost functions that penalize deviations from a desired reference to the actual value or control costs penalizing motor commands. Although important, none of these criteria seem suitable to assign costs to compliant robots that should safely interact with their environments, where precise position control is not the main objective.

A. Compliance Objective Measures

Optimizing for high compliance is desirable, because it reduces the risk of damaging anything during the interaction in unstructured environments. However, to our best knowledge, it has not been considered yet to define a single scalar value that represents the compliance of a complex structured system like a robot manipulator. We propose to use the eigenvalues of the stiffness and damping matrices involved in the various task spaces as an indicator for the compliance of the robot. The higher the eigenvalues of the stiffness matrix, the stiffer the robot behaves on external disturbances and high eigenvalues of the damping matrices enforce velocity references significantly, which also corresponds to a form of dynamic stiffness. In order to get comparable results, these task space stiffness/dampings have to be projected into a common space, the joint space, where the actual control happens. Therefore, we propose to use the sum of the eigenvalues of into the joint space projected stiffness/damping matrices as a scalar indicator for the compliance of a robot. Fortunately, our controller framework perfectly fits with this idea, as the controller computes the projected joint space stiffness/damping matrices (6), (7). The compliance indicator can be calculated efficiently for the set $\tilde{K}_d^{n \times n}$ of consecutive projected stiffness matrices by $\frac{1}{n} \sum_{t=1}^{n} \text{trace} (\tilde{K}^T_t)$ for the static stiffness, or with $\tilde{K}_d$ for a dynamic compliance. This definition is not limited to the specific control framework presented here. For instance, the symmetric matrix $J^T_{\phi} K_d J_{\phi}$ corresponds to general projected task space stiffnesses. We believe that in many control frameworks similar structures can be found, such that our definition can generally be applied.

B. Contact Force Objective Measures

If the goal is to exert a certain reference force $f^\text{ref}$ on the environment, then the root-mean-squared-error (RMSE) between the measured force and the reference defines reasonable costs. Further, in order not to brake anything, the transition from the free motion to the contact should be as smooth as possible, which can be measured by the force peak $f_{\text{os}}$ at the moment of contact. An alternative is to quantify the smoothness of the force signal, for example with

$$J = \frac{1}{T} \int_T \left( \left| \frac{d}{dt} f(t) \right| + \left| \frac{d^2}{dt^2} f(t) \right| + \left| \frac{d^3}{dt^3} f(t) \right| \right) \, dt. \quad (11)$$

VI. CONSTRAINED BAYESIAN OPTIMIZATION

Finally, we solve the optimization problem (1), i.e. finding a controller parameter $\xi$ that minimizes the costs $J(\xi)$ under the success constraint $S(\xi) = 1$, with the method from Englert et al. [3], which will be summarized shortly, for details refer to [3]. The initial parameter $\xi_1$ is specified by hand such that the execution is successful. The algorithm then iteratively selects parameters $\xi_t$ that should be tested on the real robot. The outcome of each experiment yields a certain cost $J_t = J(\xi_t)$ and success indicator $S_t = S(\xi_t)$, which are collected in a dataset $D = \{(\xi_t, J_t, S_t)\}_{t=1}^n$. From this dataset, two GPs are learned in every iteration, one regression GP $g_J$ for the cost function and one binary classifier $g_S$ for the success region. In the regression case,
a GP models the probability of the model \( g_{\mathcal{D}} \) conditioned on the dataset \( \mathcal{D} \) as a Gaussian distribution \( P(g_{\mathcal{D}}(\xi)|\mathcal{D}) = \mathcal{N}(g_{\mathcal{D}}(\xi)|\mu(\xi),\Sigma(\xi)) \) around the mean model function \( \mu(\xi) \) and the model variance \( \Sigma(\xi) \). For the binary classifier \( S(\xi) \in \{0, 1\} \), the GP defines a discriminative function \( g_{S} \), representing class probabilities \( P(S(\xi) = 1) \). These GPs are specified by a kernel and a prior mean function \( m \). For both the regression and classification GP, we use a Gaussian kernel. We refer to Rasmussen et al. [15] for details about GPs. Based upon the information encoded in these GPs about the “true” cost function and success region, the parameter that should be tested next is determined by maximizing an acquisition function \( \xi_{u+1} = \argmax_{\xi \in \mathcal{D}} a(\xi) \). Englert et al. [3] proposed a novel acquisition function

\[
a(\xi) = g_S(\xi) \geq 0 \cdot P_{1_g_{\mathcal{D}}}(\xi) + g_S(\xi) = 0 \cdot \Sigma g_S(\xi). \quad (12)
\]

The first term focuses on exploiting the inner region, where success is expected. This is done by the probability of improvement \( P_{1_g_{\mathcal{D}}}(\xi) \), which is defined as the probability that the costs are lower at a specific point \( \xi \) inside the learned success region than the current best, successful point \( \xi^* \) in the dataset. The second term explores the success region boundary at locations where the model has a high variance \( \Sigma g_S(\xi) \). This ensures a tradeoff between both exploitation and exploration. The hyperparameters of this algorithm (e.g., kernel widths) allow to control this tradeoff and to model the uncertainty about the cost function/success region.

### VII. Experiments

We show the effectiveness of our approach in two experiments on a real robot platform, the PR2. A typical benchmark problem for interaction controllers is the task of establishing contact while sliding on a surface. In the first experiment, we focused on establishing the contact between the end-effector of the robot and the surface of a cabinet. The second experiment additionally included a slide, after the contact has been established. For both experiments, a force reference \( f_{\text{ref}} = -2 \text{ N} \) was specified. The parameter learning was performed with a cost function that evaluates the RMSE between the measured and the actual force and the force peak \( f_{\text{os}} \) at the moment of contact. The execution was considered to be successful (\( S = 1 \)), if the contact was established and maintained during the execution. For the prior mean function of \( g_{\mathcal{D}} \), we chose \( m = 2 \), which corresponds to the costs of a virtual execution in which the robot does not move at all. For \( g_{S} \) the prior mean function was \( m = -7 \), ensuring that regions that have not been explored yet are considered to be unsuccessful. The learned parameters were the ones of a 1D end-effector position/force task, more specifically the velocity reference \( \gamma_{\text{ref}} \) towards the surface, the corresponding gain \( K_{d} \) and its force control decay coefficient \( \alpha \). \( \gamma \) was fixed. Other tasks were the orientation of the end-effector in the direction of the constraint, position at a specific height and joint limits. For the sliding, an additional position task orthogonal to the end-effector orientation was set. Knowing the exact position of the surface is not necessary, as the robot steers towards the surface to establish the contact and the limit force controller regulates the force, if necessary. Neither switching between control laws, nor inferring the contact was required.

#### A. Results: Contact establishment

The initial parameters were \( \gamma_{\text{ref}} = 0.1 \text{ m/s}, K_{d} = 9 \) and \( \alpha = 0.001 \). These were specified by hand such that the execution was successful. The discounting factor was \( \gamma = 0.9995 \). The costs for these initial parameters were \( \mathcal{J} = 2.56 \). All in all, there were 10 failures out of 127 samples. The failures mainly occurred because the robot did not touch the surface at all. This can be explained by the fact that the dynamics model of the robot is not perfect, especially friction was not modeled, hence it requires a minimal \( K_{d} \) to actually move the robot. Fig. 2a compares the measured force profiles of the initial sample with the best, successful sample after 127 trials. This best sample had the parameters \( \gamma_{\text{ref}} = 0.096 \text{ m/s}, K_{d} = 6.74, \alpha = 0.0078 \) with costs of \( \mathcal{J} = 1.22 \). The costs are low, because not only the force error was low, but also the force peak at the moment of contact was low as well. Compared to grid-search for a similar experiment, we observed 25 failures out of 45 trials without significant improvement of the costs. Furthermore, we evaluated the samples with two other cost functions, the smoothness of the force signal (11) and the dynamic compliance criteria, see section V-A. In Fig. 2b, the (successful) sample with the smoothest force signal (parameters \( K_{d} = 5.77, \alpha = 0.006, \gamma_{\text{ref}} = 0.1 \text{ m/s} \)) and with the highest dynamic compliance (parameters \( K_{d} = 5.68, \alpha = 0.0032, \gamma_{\text{ref}} = 0.1 \text{ m/s} \)) are shown. Interestingly, the overall sample with the highest compliance was a failure. This emphasizes the necessity for the success constraint.

#### B. Results: Sliding on a surface while maintaining contact

The hyperparameter for \( \alpha \) of the constrained Bayesian optimization algorithm was chosen such that a broader range of \( \alpha \) values are explored earlier. Higher \( \alpha \) allows faster convergence of the force error, under the risk of loosing the contact, as the end-effector can bump-off the surface for a too high \( \alpha \), if the stiffness of the environment is unknown. The initial sample had the parameters \( K_{d} = 9, \alpha = 0.01, \gamma_{\text{ref}} = 0.09 \text{ m/s} \) with costs of \( \mathcal{J} = 1.2 \). It turned out that a lower discounting factor of \( \gamma = 0.999 \) was favorable for the sliding experiment. Fig. 3b compares the force profiles of the initial with the best sample after 100 trials. The
parameters \( K_d = 7.86 \), \( \alpha = 0.031 \), \( \dot{y}_{ref} = 0.083 \) m/s led to the lowest costs of \( \mathcal{J} = 0.79 \). The sampled successful data points (dots), failures (red crosses) and the learned success region (purple net) after 100 samples are shown in Fig. 3a. Two of the 21 failures out of 100 samples in total are shown in Fig. 3c. One of those lost the contact during the slide, the other one at the moment the slide began. The successful sample that had the lowest costs based on the smoothness criteria was also the one with the lowest dynamic compliance measure with parameters \( K_d = 4.98 \), \( \alpha = 0.016 \), \( \dot{y}_{ref} = 0.095 \) m/s. Again, the importance of the success constraint can be seen here, as a failure sample had the lowest costs measured with the dynamic compliance or smoothness criteria. As a last experiment, the generalization capabilities of the best learned parameters were explored by changing the angle of the surface, as visualized in Fig. 4a, leading to additional movement in the \( x \)-direction (Fig. 4b), while sliding. As can be seen in Fig. 4c, the force profile of the generalization experiment was very similar to the one during learning and also had low costs of \( \mathcal{J} = 0.81 \).

VIII. CONCLUSION

In the present work, we have proposed to use constrained Bayesian optimization to improve parameters of combined interaction force/task space controllers in an active and sample efficient manner, while ensuring manipulation success. An important step here was to define proper criteria to measure the performance of such controllers. We evaluated our approach on a real robot platform for a task that consisted of establishing and maintaining contact while sliding on a surface. We have shown that constrained Bayesian optimization is a suitable method to tune controller parameters for such applications sample efficiently and successfully.

For future research we plan to investigate more complex, sequential manipulation tasks, where each phase may require different parameters to be optimized.

REFERENCES