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

Stiff and heavy industrial robots are suitable for repetitive tasks in well-defined environments. Precise position control is one of the main objectives for designing such robots. As a consequence, industrial robots usually operate in separated areas to avoid hurting humans. In contrast, the vision that robots safely collaborate and work together with humans in unstructured environments is highly appreciated. Despite much effort, the transformation from caged industrial robots to human assistants has not happened yet fully. This has several reasons: In general, the houses, cities, streets etc. we live in are very unstructured and change dynamically. Assistive robots therefore need to adapt to new situations by exploring and manipulating their environment autonomously. 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. However, seeking contacts can be dangerous, especially in uncertain environments. Therefore, in order to enable the robot to manipulate its environment robustly, safely and without the risk of damaging anything, suitable actuator and control concepts have to be used that enable compliant and force control in situations that are afflicted with high uncertainties. The controller has to cope with a priori unknown positions of objects in the world, their shapes, weights and stiffness properties. Instead of focusing on the precise execution of preplanned trajectories with joint space control, interaction control tries to establish a dynamic relationship between the robot and the environment. Variable end-effector stiffnesses and interaction force control improve the robust- and safeness during the interaction.

Although such control ideas have been around for years, the majority of robot control is still in the joint space. A possible explanation is that interaction controllers have many parameters that need to be tuned properly in order to achieve satisfactory performance. Control often deals with systems for which a sufficiently precise model is available. In such cases, finding reasonable parameters, for example by optimal control, is well-studied. The situation is, however, completely different for robots that interact with unstructured environments. Here, not only the robot needs to be modeled, but also the environment, which is often impossible to do precisely. Therefore, estimating good parameters for interaction controllers is difficult, as they depend on unknown physical properties (stiffness, mass etc.) of the objects in the environment.

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. They have developed this method to optimize key points in trajectories.

In the present work, we combine this special constrained Bayesian optimization with a class of interaction controllers 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. Our controller framework enables compliant control in multiple task spaces, as well as to limit interaction forces simultaneously. By including a constraint into the optimization process, we can ensure that the optimized parameters lead to successful

Danny Drieß, Peter Englert and Marc Toussaint are with the Machine Learning and Robotics Lab, University of Stuttgart, Germany.
Email: danny.driess@gmx.de,
peter.englert@ipvs.uni-stuttgart.de,
marc.toussaint@ipvs.uni-stuttgart.de

manipulations.

Our paper is organized as follows. Section II reviews related work that deals with automatic tuning of controller parameters and interaction control concepts briefly. 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, respectively. 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. Furthermore, it is desirable to regulate contact forces, for example to limit forces during the interaction in order to improve the safeness. Villani et al. [6] give a comprehensive overview of different force control concepts like hybrid control.

Estimating reasonable parameters for these compliant/force controllers can be challenging [6], 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 optimization in robotics

Learning/tuning controllers has a great history in robotics. Often, this consists of identifying dynamics models [7], [8]. 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. [9] applied (unconstrained) Bayesian optimization 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. [10] 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. [11] adhere 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.

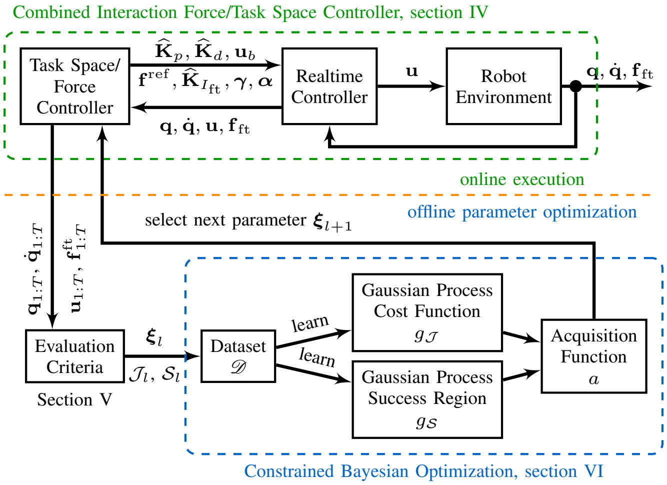

Combined Interaction Force/Task Space Controller, section IV
Task Space/ Force Controller
$\widehat{\mathbf{K}}_p, \widehat{\mathbf{K}}_d, \mathbf{u}_b$
$\mathbf{f}^{\mathrm{ref}}, \widehat{\mathbf{K}}_{I_{\mathrm{ft}}}, \boldsymbol{\gamma}, \boldsymbol{\alpha}$
Realtime Controller
$\mathbf{u}$
Robot Environment
$\mathbf{q}, \dot{\mathbf{q}}, \mathbf{f}_{\mathrm{ft}}$
$\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}, \mathbf{f}_{\mathrm{ft}}$
online execution
offline parameter optimization
select next parameter $\boldsymbol{\xi}_{l+1}$
$\mathbf{q}_{1:T}, \dot{\mathbf{q}}_{1:T}$
$\mathbf{u}_{1:T}, \mathbf{f}^{\mathrm{ft}}_{1:T}$
Evaluation Criteria
Section V
$\boldsymbol{\xi}_l$
$\mathcal{J}_l, \mathcal{S}_l$
Dataset $\mathscr{D}$
learn
learn
Gaussian Process Cost Function $g_{\mathcal{J}}$
Gaussian Process Success Region $g_{\mathcal{S}}$
Acquisition Function a
Constrained Bayesian Optimization, section VI

Fig. 1. Overview: Autonomic optimization of parameters for combined interaction force/task space controllers with constrained Bayesian optimization.

III. Overview: Constrained Bayesian Optimization for 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 $\boldsymbol{\xi} \in \mathcal{P} \subset \mathbb{R}^p$ 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 $\mathcal{S}$ is specified that determines whether an execution was successful or not (for example, manipulation success). Every specific choice of a parameter $\boldsymbol{\xi}$ yields a certain cost $\mathcal{J}(\boldsymbol{\xi}) \in \mathbb{R}$ and success $\mathcal{S}(\boldsymbol{\xi}) \in \{0, 1\}$, observable only, when executed in real world. The parameter tuning process therefore consists of the *constrained* optimization problem

$$\min_{\boldsymbol{\xi} \in \mathcal{P}} \mathcal{J}(\boldsymbol{\xi}) \quad \text{subject to} \quad \mathcal{S}(\boldsymbol{\xi}) = 1. \tag{1}$$

Unfortunately, standard methods to solve (1) are not applicable, because no structural information like gradients about $\mathcal{J}$ and $\mathcal{S}$ is available. Instead, only (probably noisy) samples of $\mathcal{J}$ and $\mathcal{S}$ can be drawn for specific parameter choices $\boldsymbol{\xi}$. In order to overcome this, we utilize the method from Englert et al. [3]. This constrained Bayesian optimization (blue box) iteratively selects parameters $\boldsymbol{\xi}_l$ 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 $\mathscr{D} = \{(\boldsymbol{\xi}_l, \mathcal{J}_l, \mathcal{S}_l)\}_{l=1}^{w}$ with $\mathcal{J}_l = \mathcal{J}(\boldsymbol{\xi}_l)$ and $\mathcal{S}_l = \mathcal{S}(\boldsymbol{\xi}_l)$. Based upon this dataset, we learn two Gaussian processes (GP), one regression $g_{\mathcal{J}}$ that is a surrogate model for the cost function $\mathcal{J}$ and a classifier $g_{\mathcal{S}}$ 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 $\boldsymbol{\xi}_{w+1}$ should be rolled out next. The decision is made based upon the goal of both exploring the success region without sampling too much 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

$$\mathbf{u} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}), \tag{2}$$

with the mass/inertia matrix $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$, the vector $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ representing Coriolis, centripetal and gravity forces, $\mathbf{u} \in \mathbb{R}^n$ are the motor commands and $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ 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* $\phi : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}^d$, $\mathbf{y} = \phi(\mathbf{q})$, which are (differentiable) functions from the robot configuration $\mathbf{q}$ to a d-dimensional space. The most 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, which is specified by positive definite stiffness $\mathbf{K}_p \in \mathbb{R}^{d \times d}$ and damping $\mathbf{K}_d \in \mathbb{R}^{d \times d}$ matrices that realize a PD behavior around the desired references $\mathbf{y}^{\text{ref}}, \dot{\mathbf{y}}^{\text{ref}}, \ddot{\mathbf{y}}^{\text{ref}} \in \mathbb{R}^d$ in that task space. More formally, in m task spaces $\phi_i : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}^{d_i}$, $i = 1, \ldots, m$, we want to impose certain acceleration laws

$$\ddot{\phi}_i = \ddot{\mathbf{y}}_i^* = \ddot{\mathbf{y}}_i^{\text{ref}} + \mathbf{K}_{p_i}\left(\mathbf{y}_i^{\text{ref}} - \mathbf{y}_i\right) + \mathbf{K}_{d_i}\left(\dot{\mathbf{y}}_i^{\text{ref}} - \dot{\mathbf{y}}_i\right). \tag{3}$$

With suitable tuned gain matrices $\mathbf{K}_{p_i}$ and $\mathbf{K}_{d_i}$, 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 $\mathbf{u}$ such that the robot behaves like these virtual mass-spring-damper-systems. Peters et al. [12] 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 ϕ with respect to time twice, i.e. $\mathbf{J}_{\phi_i}\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{\phi_i}\dot{\mathbf{q}} = \ddot{\mathbf{y}}_i^*$ with the Jacobian $\mathbf{J}_{\phi_i}$ of ϕ_i. In the formulation of [12], 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

$$\min_{\ddot{\mathbf{q}}} \|\underbrace{\mathbf{M}\ddot{\mathbf{q}} + \mathbf{F}}_{\mathbf{u}} - \mathbf{u}_0\|_{\mathbf{H}}^2 + \sum_{i=1}^{m} \left\|\mathbf{J}_{\phi_i}\ddot{\mathbf{q}} - \ddot{\mathbf{y}}_i^* + \dot{\mathbf{J}}_{\phi_i}\dot{\mathbf{q}}\right\|_{\mathbf{C}_i}^2 \tag{4}$$

by relaxing the constraints to become a squared penalty. Similar to [12], the motor commands can be weighted with the positive definite $\mathbf{H} \in \mathbb{R}^{n \times n}$. 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 $\mathbf{C}_i \in \mathbb{R}^{d_i \times d_i}$. Those with higher eigenvalues correspond to more important tasks. In the limit $\mathbf{C} \to \infty$ in any norm, this controller converges for one task to the same framework presented in [12]. With $\mathbf{u}_0 \in \mathbb{R}^n$ a nullcost control law can be realized, for example to stabilize in the joint space or to not penalize gravity compensation control costs. The solution of (4), given by $\ddot{\mathbf{q}}^* = \mathbf{A}^{-1}\left(\mathbf{M}\mathbf{H}\left(\mathbf{u}_0 - \mathbf{F}\right) + \sum_{i=1}^m \mathbf{J}_{\phi_i}^T \mathbf{C}_i \left(\ddot{\mathbf{y}}_i^* - \dot{\mathbf{J}}_{\phi_i}\dot{\mathbf{q}}\right)\right)$ with the symmetric positive definite generalized inverse $\mathbf{A} = \mathbf{M}\mathbf{H}\mathbf{M} + \sum_{i=1}^m \mathbf{J}_{\phi_i}^T \mathbf{C}_i \mathbf{J}_{\phi_i}$, 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 $\mathbf{q}_0$ with $\mathbf{y}_i \approx \phi_i(\mathbf{q}_0) + \mathbf{J}_{\phi_i}\left(\mathbf{q} - \mathbf{q}_0\right)$. This yields an affine control law

$$\mathbf{u}^* = \mathbf{u}_b - \widehat{\mathbf{K}}_p \mathbf{q} - \widehat{\mathbf{K}}_d \dot{\mathbf{q}} \tag{5}$$

of the joint state with the *projected* stiffness/damping matrices

$$\widehat{\mathbf{K}}_p = \mathbf{M}\mathbf{A}^{-1} \sum_{i=1}^{m} \mathbf{J}_{\phi_i}^T \mathbf{C}_i \mathbf{K}_{p_i} \mathbf{J}_{\phi_i} \in \mathbb{R}^{n \times n} \tag{6}$$

$$\widehat{\mathbf{K}}_d = \mathbf{M}\mathbf{A}^{-1} \sum_{i=1}^{m} \mathbf{J}_{\phi_i}^T \mathbf{C}_i \left(\mathbf{K}_{d_i} \mathbf{J}_{\phi_i} + \dot{\mathbf{J}}_{\phi_i}\right) \in \mathbb{R}^{n \times n} \tag{7}$$

and the bias vector $\mathbf{u}_b \in \mathbb{R}^n$ with

$$\begin{aligned} \mathbf{u}_b = \mathbf{F} + \mathbf{M}\mathbf{A}^{-1} \big[\mathbf{M}\mathbf{H}\left(\mathbf{u}_0 - \mathbf{F}\right) + \\ \sum_{i=1}^{m} \mathbf{J}_{\phi_i}^T \mathbf{C}_i \left(\ddot{\mathbf{y}}_i^{\text{ref}} + \mathbf{K}_{p_i}\left(\mathbf{y}_i^{\text{ref}} - \phi_i(\mathbf{q}_0) + \mathbf{J}_{\phi_i}\mathbf{q}_0\right) + \mathbf{K}_{d_i}\dot{\mathbf{y}}_i^{\text{ref}}\right)\Big] \end{aligned} \tag{8}$$

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 ϕ to a desired reference $\mathbf{f}^{\text{ref}} \in \mathbb{R}^d$. The force/torque measurements $\mathbf{f}_{\text{ft}} \in \mathbb{R}^6$ of the sensors located at the wrists of our PR2 can be projected to the actual force $\mathbf{f}$ in the task space via a pseudo-inverse of the Jacobian from the sensor frame to the task space. Many force control concepts [6] realize the force reference by the feed-forward term $\mathbf{u}_f = \mathbf{J}_\phi^T \mathbf{f}^{\mathrm{ref}}$ 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 controller term $\mathbf{e}^{\mathrm{ft}} \in \mathbb{R}^d$, that is initialized by $\mathbf{0}$ and updated in every loop with

$$\mathbf{e}^{\mathrm{ft}}_j \leftarrow \boldsymbol{\gamma}_j \mathbf{e}^{\mathrm{ft}}_j + \left[|\mathbf{f}_j| > \left|\mathbf{f}^{\mathrm{ref}}_j\right|\right] \left(\mathbf{f}^{\mathrm{ref}}_j - \mathbf{f}_j\right) \quad \forall_{j=1,\ldots,d}. \tag{9}$$

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 $\boldsymbol{\gamma} \in (0, 1]^d$ ensures that the force integral term vanishes exponentially, if no force control is necessary anymore. The motor commands are

$$\mathbf{u}^*_f = \mathbf{J}_\phi^T \boldsymbol{\alpha} \mathbf{e}^{\mathrm{ft}} = \widehat{\mathbf{K}}_{I_{\mathrm{ft}}} \mathbf{e}^{\mathrm{ft}}. \tag{10}$$

With the positive definite matrix $\boldsymbol{\alpha} \in \mathbb{R}^{d \times d}$, 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 [6], 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. Doing this, simultaneous force and position control is possible. To actually realize a certain force reference $\mathbf{f}^{\mathrm{ref}}$ in the task space ϕ, we set a *velocity* reference $\dot{\mathbf{y}}_{\mathrm{ref}}$ towards the environmental constraint. By proper tuning of the velocity gain $\mathbf{K}_d$ and the force control coefficients $\boldsymbol{\alpha}, \boldsymbol{\gamma}$, the force reference is realized, as the constant velocity towards the constraint increases the force, depending on $\mathbf{K}_d$ and $\dot{\mathbf{y}}_{\mathrm{ref}}$, while the limit force controller intervenes to stabilize the force reference.

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 the performance are cost functions that penalize deviations from a desired reference to the actual value or control costs that penalize 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 stiffnesses/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 $\widehat{\mathbf{K}}^p_{1:n_T}$ of consecutive projected stiffness matrices by $\frac{1}{n_T}\sum_{t=1}^{n_T} \mathrm{trace}\left(\widehat{\mathbf{K}}^p_t\right)$ for the static stiffness, or with $\widehat{\mathbf{K}}_d$ for a dynamic compliance. This definition is not limited to the specific control framework presented here. For instance, the symmetric matrix $\mathbf{J}_\phi^T \mathbf{K}_p \mathbf{J}_\phi$ corresponds to general projected task space stiffnesses. We belief 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 $\mathbf{f}^{\mathrm{ref}}$ 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 $\mathbf{f}_{\mathrm{os}}$ at the moment of contact. An alternative is to quantify the smoothness of the force signal, which was done in the experiments by

$$\mathcal{J} = \tfrac{1}{|T|}\int_T \left(\left|\tfrac{\mathrm{d}}{\mathrm{d}t}\mathbf{f}(t)\right| + \left|\tfrac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{f}(t)\right| + \left|\tfrac{\mathrm{d}^3}{\mathrm{d}t^3}\mathbf{f}(t)\right| \right) \mathrm{d}t. \tag{11}$$

VI. Constrained Bayesian Optimization

Finally, we solve the optimization problem (1), i.e. finding a controller parameter $\boldsymbol{\xi}$ that minimizes the costs $\mathcal{J}(\boldsymbol{\xi})$ under the success constraint $\mathcal{S}(\boldsymbol{\xi}) = 1$, with the method from Englert et al. [3], which will be summarized shortly, for details refer to [3]. The initial parameter $\boldsymbol{\xi}_1$ is specified by hand such that the execution is successful. The algorithm then iteratively selects parameters $\boldsymbol{\xi}_l$ that should be tested on the real robot. The outcome of each experiment yields a certain cost $\mathcal{J}_l = \mathcal{J}(\boldsymbol{\xi}_l)$ and success indicator $\mathcal{S}_l = \mathcal{S}(\boldsymbol{\xi}_l)$, which are collected in a dataset $\mathscr{D} = \{(\boldsymbol{\xi}_l, \mathcal{J}_l, \mathcal{S}_l)\}_{l=1}^{w}$. From this dataset, two GPs are learned in every iteration, one regression GP $g_{\mathcal{J}}$ for the cost function and one binary classifier g_S for the success region.

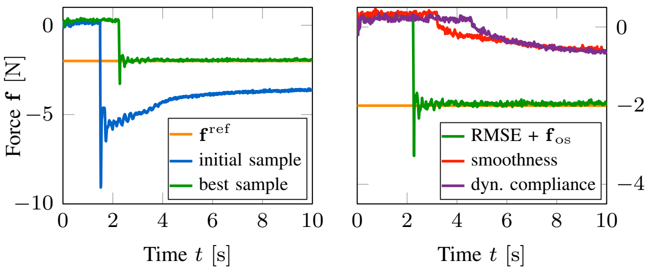
In the regression case, a GP models the probability of the model $g_\mathcal{J}$ conditioned on the dataset $\mathscr{D}$ as a Gaussian distribution $P(g_\mathcal{J}(\boldsymbol{\xi})|\mathscr{D}) = \mathcal{N}\left(g_\mathcal{J}(\boldsymbol{\xi})\,|\mu(\boldsymbol{\xi}), \mathbb{V}(\boldsymbol{\xi})\right)$ around the mean model function $\mu(\boldsymbol{\xi})$ and the model variance $\mathbb{V}(\boldsymbol{\xi})$. For the binary classifier $S(\boldsymbol{\xi}) \in \{0,1\}$, the GP defines a discriminative function $g_\mathcal{S}$, representing class probabilities $P(\mathcal{S}(\boldsymbol{\xi}) = 1)$. 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. [13] 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 $\boldsymbol{\xi}_{w+1} = \mathrm{argmax}_{\boldsymbol{\xi}\in\mathcal{P}}\, a(\boldsymbol{\xi}, g_\mathcal{J}, g_\mathcal{S}, \mathscr{D})$. Englert et al. [3] proposed a novel acquisition function

$$a(\boldsymbol{\xi}) = [g_\mathcal{S}(\boldsymbol{\xi}) \geq 0] \cdot \mathrm{PI}_{g_\mathcal{J}}(\boldsymbol{\xi}) + [g_\mathcal{S}(\boldsymbol{\xi}) = 0] \cdot \mathbb{V}_{g_\mathcal{S}}(\boldsymbol{\xi}). \quad (12)$$

The first term focuses on *exploiting* the inner region, where success is expected. This is done by the probability of improvement $\mathrm{PI}_{g_\mathcal{J}}(\boldsymbol{\xi})$, which is defined as the probability that the costs are lower at a specific point $\boldsymbol{\xi}$ inside the learned success region than the current best, successful point $\boldsymbol{\xi}^*$ in the dataset. The second term *explores* the success region boundary at locations where the model has a high variance $\mathbb{V}_{g_\mathcal{S}}(\boldsymbol{\xi})$. 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 from Willow Garage. A typical benchmark problem for interaction controllers is the task of establishing and maintaining the 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 $\mathbf{f}^{\mathrm{ref}} = -2$ 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 $\mathbf{f}_{\mathrm{os}}$ at the moment of contact. The execution was considered to be successful ($\mathcal{S} = 1$), if the contact was established *and* maintained during the execution. For the prior mean function of $g_\mathcal{J}$, we chose $m = 2$, which corresponds to the costs of a virtual execution in which the robot does not move at all. For $g_\mathcal{S}$ 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 $\dot{\mathbf{y}}_{\mathrm{ref}}$ towards the surface, the corresponding gain $\mathbf{K}_d$ and its force control decay coefficient $\boldsymbol{\alpha}$. γ was fixed. The other tasks were set such that the end-effector is oriented in the direction of the constraint and positioned at a specific height. These tasks were parameterized in a way that they did not interfere with the force control objective. The exact position of the surface is not important, 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) Improvement from the initial to the best learned parameter.

(b) Best samples determined by different cost functions.

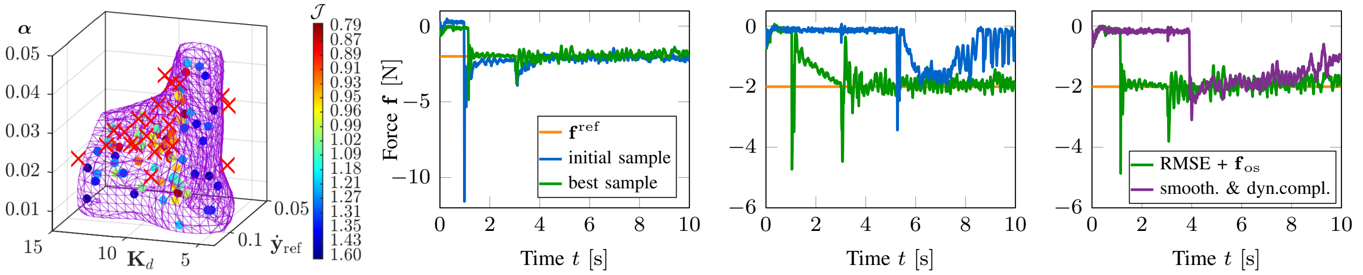
Fig. 2. Measured force profiles of the contact establishment experiment.

A. Results: Contact establishment

The initial parameters were $\dot{\mathbf{y}}_{\mathrm{ref}} = 0.1$ m/s, $\mathbf{K}_d = 9$ and $\boldsymbol{\alpha} = 0.001$. 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 $\mathbf{K}_d$ 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 $\dot{\mathbf{y}}_{\mathrm{ref}} = 0.096$ m/s, $\mathbf{K}_d = 6.74$, $\boldsymbol{\alpha} = 0.0078$ 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 $\mathbf{K}_d = 5.77$, $\boldsymbol{\alpha} = 0.006$, $\dot{\mathbf{y}}_{\mathrm{ref}} = 0.1$ m/s) and with the highest dynamic compliance (parameters $\mathbf{K}_d = 5.68$, $\boldsymbol{\alpha} = 0.0032$, $\dot{\mathbf{y}}_{\mathrm{ref}} = 0.1$ 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 $\boldsymbol{\alpha}$ of the constrained Bayesian optimization algorithm was chosen such that a broader range of $\boldsymbol{\alpha}$ values are explored earlier. Higher $\boldsymbol{\alpha}$ 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 $\boldsymbol{\alpha}$, if the stiffness of the environment is unknown. The initial sample had the parameters $\mathbf{K}_d = 9$, $\boldsymbol{\alpha} = 0.01$, $\dot{\mathbf{y}}_{\mathrm{ref}} = 0.09$ m/s with costs of $\mathcal{J} = 1.2$. It turned out that a



(a) Samples (dots), failures (crosses) and success region (pruple).

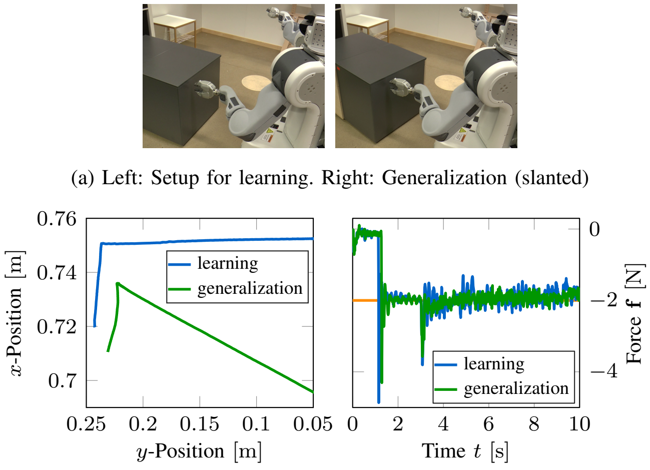
(b) Improvement from the initial to the best learned parameter.

(c) Two failure samples that have lost contact.

(d) Best samples determined by different cost functions.

Fig. 3. Results from the sliding experiment. First, the contact between the end-effector of the robot and the environment was established, then, the robot slided along the surface, while maintaining contact and exerting a reference force of $\mathbf{f}^{\text{ref}} = -2$ N. (b), (c) and (d) show force profiles.

lower discounting factor of $\boldsymbol{\gamma} = 0.999$ was favorable for the sliding experiment. Fig. 3b compares the force profiles of the initial with the best sample after 100 trials. The parameters $\mathbf{K}_d = 7.86$, $\boldsymbol{\alpha} = 0.031$, $\dot{\mathbf{y}}_{\text{ref}} = 0.083$ 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 $\mathbf{K}_d = 4.98$, $\boldsymbol{\alpha} = 0.016$, $\dot{\mathbf{y}}_{\text{ref}} = 0.095$ 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 $\mathcal{J} = 0.81$.



(a) Left: Setup for learning. Right: Generalization (slanted)

(b) End-effector trajectories.

(c) Force profiles.

Fig. 4. Generalization of the learned parameters for the sliding experiment.

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.

References

[1] M. Toussaint, N. Ratliff, J. Bohg, L. Righetti, P. Englert, and S. Schaal, “Dual execution of optimized contact interaction trajectories,” in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[2] R. Deimel, C. Eppner, J. lvarez-Ruiz, M. Maertens, and O. Brock, “Exploitation of environmental constraints in human and robotic grasping,” in *International Symposium on Robotics Research (ISRR)*, 12 2013.

[3] P. Englert and M. Toussaint, “Combined Optimization and Reinforcement Learning for Manipulations Skills,” in *Proc. of Robotics: Science and Systems*, 2016.

[4] N. Hogan, “Impedance control: An approach to manipulation,” *Journal of Dynamic Systems, Measurement and Control*, 1985.

[5] O. Khatib, “A unified approach for motion and force control of robot manipulators: The operational space formulation,” *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 1, pp. 43–53, Feb. 1987.

[6] L. Villani and J. De Schutter, *Force Control*. Springer, 2008, ch. 7.

[7] J. Schreiter, P. Englert, D. Nguyen-Tuong, and M. Toussaint, “Sparse gaussian process regression for compliant, real-time robot control,” in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[8] D. Nguyen-Tuong and J. Peters, “Model learning for robot control: a survey,” *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.

[9] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe, “Automatic LQR tuning based on gaussian process optimization: Early experimental results,” in *Machine Learning in Planning and Control of Robot Motion Workshop at the Int. Conf. on Intelligent Robots and Systems*, 2015.

[10] J. Schreiter, D. Nguyen-Tuong, M. Eberts, B. Bischoff, H. Markert, and M. Toussaint, “Safe exploration for active learning with gaussian processes,” in *Proc. of the Conf. on Machine Learning (ECML)*, 2015.

[11] F. Berkenkamp, A. P. Schoellig, and A. Krause, “Safe controller optimization for quadrotors with gaussian processes,” in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, May 2016, pp. 491–496.

[12] J. Peters, M. Mistry, F. E. Udwadia, R. Cory, J. Nakanishi, and S. Schaal, “A unifying framework for the control of robotics systems,” in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005.

[13] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.