Finding safe 3D robot grasps through efficient haptic exploration with unscented Bayesian optimization and collision penalty

João Castanheira[1], Pedro Vicente[1], Ruben Martinez-Cantin[2,3], Lorenzo Jamone[4,1], Alexandre Bernardino[1]

***Abstract*— Robust grasping is a major, and still unsolved, problem in robotics. Information about the 3D shape of an object can be obtained either from prior knowledge (*e.g.*, accurate models of known objects or approximate models of familiar objects) or real-time sensing (*e.g.*, partial point clouds of unknown objects) and can be used to identify good potential grasps. However, due to modeling and sensing inaccuracies, local exploration is often needed to refine such grasps and successfully apply them in the real world. The recently proposed unscented Bayesian optimization technique can make such exploration safer by selecting grasps that are robust to uncertainty in the input space (*e.g.*, inaccuracies in the grasp execution). Extending our previous work on 2D optimization, in this paper we propose a 3D haptic exploration strategy that combines unscented Bayesian optimization with a novel collision penalty heuristic to find safe grasps in a very efficient way: while by augmenting the search-space to 3D we are able to find better grasps, the collision penalty heuristic allows us to do so without increasing the number of exploration steps.**

I. Introduction

Robotic grasping and manipulation has been a major research area for many years now [1], [2], [3], [4]. However, the robust grasping of arbitrary objects is still an open problem. According to Bohg et al [4], an object can be (i) known, (ii) familiar or (iii) unknown. In the first category, accurate models of the object are available and can be exploited for transforming a grasping problem into a pose estimation one [5], [6], [7], since the optimum grasp could be learned *a priori* and retrieved from a database of possible grasp poses. In the second case, the object shares some features (*e.g.*, visual 3D geometric features) with previous known objects and can fall back on the previous case by adjusting the final target pose with online learning methods [8]. Finally, when grasping unknown objects we should resort to real-time sensing (*e.g.*, partial point clouds) using exploration for retrieving the optimum grasp pose.

Even with known objects, it is hard to achieve a robust grasp since small errors in object sensing or motor execution may turn optimal grasps into bad grasps. Indeed, many object grasping controllers rely on a complete open-loop approach to achieve the pre-computed optimal grasp [9], [10], [11], [12], [13], where the robot looks once to the scene, computes the pose of the object, and then drives the arm to the grasping pose without sensory feedback. Due to modeling inaccuracies and sensing limitations, such open-loop approach does not permit robust grasping performance. As a matter of fact, very few grasping pipelines exploit tactile or visual feedback to achieve a robust grasp execution [4]. In general, some form of local adjustment is needed to adapt the desired grasps the objects in the real world. For instance, positioning errors can be reduced by means of visual servoing [14], or by closing the control-loop with force and torque sensors [15], or by learning the optimum grasping position with a trial-and-error approach [16].

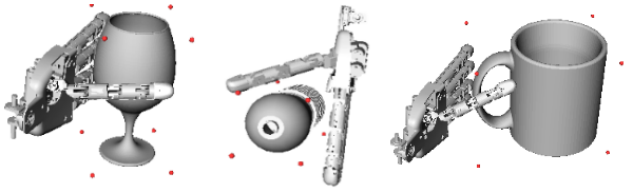


Fig. 1: Examples of objects to perform grasp optimization on simulation. Initial pose for each test object.

In this work, we follow the trial-and-error approach based on tactile feedback to reduce the effect of uncertainty in the input space (*i.e.*, the hand positioning in 3D) and eventually obtain robust and safe grasps. We apply Bayesian optimization methods to find such grasps. This approach will minimize the number of required exploration actions needed to achieve the optimum grasp.

In our previous work [17] we have introduced the unscented Bayesian optimization (UBO) algorithm to search for safe grasps in a 2D space. In this paper, we scale to 3D and prove that better grasps can be found by UBO against Bayesian optimization (BO). Notably, the superior outcome of the 3D UBO method was possible due to a new Collision Penalty (CP) strategy proposed to increases the convergence speed.

The rest of the paper is organized as follows: we present the BO framework (Sec. II), followed by the recently proposed UBO (Sec. III) and we introduce the CP heuristic in Sec. IV. Then, we present experimental results on grasping simulations in Sec. VI, using the objects presented in Fig. 1, and conclude our paper in Sec. VII.

[1] Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal. joao.castanheira@tecnico.ulisboa.pt, {pvicente,alex}@isr.tecnico.ulisboa.pt

[2] Centro Universitario de la Defensa, Zaragoza, Spain. rmcantin@unizar.es

[3] SigOpt Inc. San Francisco, CA

[4] ARQ (Advanced Robotics at Queen Mary), School of Electronic Engineering and Computer Science, Queen Mary University of London, UK l.jamone@qmul.ac.uk

This work was partially supported by FCT (UID/EEA/50009/2013 and PD/BD/135115/2017), and by EPSRC UK (project NCNR, National Centre for Nuclear Robotics, EP/R02572X/1). We acknowledge the support of NVIDIA with the donation of a GPU.

II. Bayesian optimization

In this article, we follow the BO notation presented in [17], [18]. Formally, the problem is based on finding the optimum (maximum) of an unknown real valued function $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is a compact space, $\mathcal{X} \subset \mathbb{R}^d$, and $d \geq 1$ its dimension, with a maximum budget of N evaluations of the target function f. In our problem f is a grasp quality criterion described in Sec. IV, and $\mathcal{X}$ is the space of relative hand–object poses. For each query point $\mathbf{x_i} \in \mathcal{X}$ a grasp trial has an outcome $y_i = f(\mathbf{x_i}) + \eta$, where η is zero-mean noise with variance σ_η^2.

The BO consists of two stages. First, given a query point $\mathbf{x_i}$ and outcome y_i we update a probabilistic surrogate model of f, a distribution over the family of functions $P(f)$, where the target function f belongs. A very popular choice is a Gaussian process (GP), built incrementally by sampling over the input-space. Second, a Bayesian decision process, where an acquisition function uses the information gathered in the GP, is used to decide on the best point $\mathbf{x}$ to query (*i.e.*, to sample) next. The goal is to guide the search to the optimum, while balancing the exploration *vs* exploitation trade-off.

A. Surrogate model estimations

The Gaussian process $\mathcal{GP}(\mathbf{x}|\mu, \sigma^2, \theta)$ has inputs $\mathbf{x} \in \mathcal{X}$, scalar outputs $\mathrm{y} \in \mathbb{R}$ and an associated kernel function $k(\cdot, \cdot)$ with hyperparameters $\boldsymbol{\theta}$. The hyperparameters are optimized during the process using the slice sampling method [19], [20], resulting in m samples $\boldsymbol{\Theta} = [\boldsymbol{\theta}_i]_{i=1}^m$.

From the GP we can get an estimate, $\hat{y}$, of f by conditioning the distribution over functions to what is already known. At a step n we have a dataset of observations $\mathcal{D}_n = (\mathbf{X}, \mathbf{y})$, represented by all the queries until that step, $\mathbf{X} = (\mathbf{x}_{1:n})$, and their respective outcomes, $\mathbf{y} = (y_{1:n})$. The prediction, $y_{n+1} = \hat{y}(\mathbf{x}_{n+1})$, at an arbitrary new query point $\mathbf{x}_{n+1}$, with kernel k_i conditioned on the i-th hyperparameter sample $k_i = k(\cdot, \cdot|\boldsymbol{\theta}_i)$, is normally distributed and given by:

$$\begin{aligned} \hat{y}(\mathbf{x}_{n+1}) &\sim \sum_{i=1}^{m} \mathcal{N}(\mu_i(\mathbf{x}_{n+1}), \sigma_i^2(\mathbf{x}_{n+1})) \\ \mu_i(\mathbf{x}_{n+1}) &= \mathbf{k}_i^T \mathbf{K}_i^{-1} \mathbf{y} \\ \sigma_i^2(\mathbf{x}_{n+1}) &= k_i(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}) - \mathbf{k}_i^T \mathbf{K}_i^{-1} \mathbf{k}_i. \end{aligned} \tag{1}$$

The vector $\mathbf{k}_i$ is a kernel i evaluated at the arbitrary query point $\mathbf{x}_{n+1}$ with respect to the dataset $\mathbf{X}$, and $\mathbf{K}_i = k_i(\mathbf{X}, \mathbf{X}) + \mathbf{I} \cdot \sigma_\eta^2$ is the Gram matrix corresponding to kernel k_i for dataset $\mathbf{X}$, with noise variance σ_η^2. Note that, because we use a sampling distribution of $\boldsymbol{\Theta}$, the predictive distribution at any point $\mathbf{x}$ is a mixture of Gaussians.

B. Decision using acquisition function

To select the next query point at each iteration, we use the expected improvement criterion (EI) as the acquisition function. This function takes into consideration the predictive distribution for each point in $\mathcal{X}$, whose mean and variance are as in Eq. (1), to decide the next query point. The EI is the expectation of the improvement function $I(\mathbf{x}) = \max(0, \hat{y}(\mathbf{x}) - \mathrm{y}_n^{bopt})$, where the incumbent is the best outcome found until now (iteration n):

$$\mathrm{y}_n^{bopt} = \max(y_{1:n}). \tag{2}$$

Then, the optimum value corresponds to its associated query in the dataset and is denoted as $\mathbf{x}_n^{bopt}$. Taking the expectation over the mixture of Gaussians of the predictive distribution, the EI can be computed as

$$\begin{aligned} \mathrm{EI}(\mathbf{x}) &= \mathbb{E}_{(\hat{y}|\mathcal{D}_n, \boldsymbol{\theta}, \mathbf{x})}[\max(\mathbf{0}, \hat{y}(\mathbf{x}) - \mathrm{y}_n^{bopt}] \\ &= \sum_{i=1}^{m} [(\mu_i(\mathbf{x}) - \mathrm{y}_n^{bopt})\Phi(z_i) + \sigma_i(\mathbf{x})\phi(z_i)], \end{aligned} \tag{3}$$

where $z_i = (\mu_i(\mathbf{x}) - \mathrm{y}_n^{bopt})/\sigma_i(\mathbf{x})$, ϕ corresponds to the Gaussian probability density function (PDF) and Φ to the cumulative density function (CDF). Also, the pair (μ_i, σ_i^2) are the predictions computed in Eq. (1). Then, the new query point is selected by maximizing the EI:

$$\mathbf{x}_{n+1} = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}}\, EI(\mathbf{x}). \tag{4}$$

Lastly, to reduce initialization bias and improve global optimality, we rely on an initial design of p points based on *Latin Hypercube Sampling* (LHS), as suggested in [21].

III. Unscented Bayesian optimization

When selecting the most interesting point to query next, acquisition functions like the EI assume that the query is deterministic. However, with input noise, our query is in fact a probability distribution. Indeed, if we take the query's vicinity into consideration, a better notion and estimation of the expected outcome can be achieved. The size of the vicinity to be considered depends on the input noise power.

Thus, instead of analyzing the outcome of the EI to select the next query, we are going to analyze the posterior distribution that results from propagating the query distribution through the acquisition function. This can be done using the unscented transformation, a method used to propagate probability distributions through nonlinear functions with reduced computational cost and good accuracy. The unscented transform uses selected samples from the prior distribution designated sigma points, $\mathbf{x}^{(i)}$, and calculates the value of the nonlinear function g at each of these points. Then, the output distribution is computed based on the weighted combination of the transformed sigma points.

For a d-dimensional input space, the unscented transformation only requires a set of $2d+1$ sigma points. If the input distribution is a Gaussian, then the transformed distribution is simply $\mathbf{x}' \sim \mathcal{N}(\sum_{i=0}^{2d} w^{(i)} g(\mathbf{x}^{(i)}), \Sigma'_{\mathbf{x}})$, where $w^{(i)}$ is the weight corresponding to the i^{th} sigma point.

The unscented transformation provides mean and covariance estimates of the new distribution that are accurate to the third order of the Taylor series expansions of g, provided that the original distribution is a Gaussian prior. Another advantage of the unscented transformation is its computational cost. The $2d+1$ sigma points make the computational cost almost negligible compared to other alternatives to distribution approximation.

A. Unscented expected improvement

Considering that our prior distribution is Gaussian $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{I}\sigma_x)$, then the set of $2d+1$ sigma points of the unscented transform are computed as

$$\mathbf{x}^0 = \bar{\mathbf{x}},\ \mathbf{x}_{\pm}^{(i)} = \bar{\mathbf{x}} \pm \left(\sqrt{(d+k)\ \sigma_x}\right)_i,\ \forall i = 1...d \qquad (5)$$

where $(\sqrt{(\cdot)})_i$ is the i-th row or column of the corresponding matrix square root. In this case, k is a free parameter that can be used to tune the scale of the sigma points. For more information on choosing the optimal values for k, refer to [22]. For these sigma points, the corresponding weights are

$$w^0 = \frac{k}{d+k},\ w_{\pm}^{(i)} = \frac{1}{2(d+k)},\ \forall i = 1...d \qquad (6)$$

Considering the EI as the nonlinear function g, then we are making a decision on the next query considering that there is input noise. This new decision can be interpreted as a new acquisition function, the unscented expected improvement (UEI). It corresponds to the expected value of the EI with respect to the input noise distribution:

$$UEI(\mathbf{x}) = \sum_{i=0}^{2d} w^{(i)} EI(\mathbf{x}^{(i)}), \qquad \mathbf{x} \in \mathcal{X}. \qquad (7)$$

This strategy, reduces the chance that the next query point is located in an unsafe region, *i.e.*, where a small change on the input (induced by noise) implies a bad outcome.

B. Unscented optimal incumbent

In BO, the final decision for what we consider the optimum is independent of the acquisition function. We defined the incumbent for BO in Eq. (2), as the best observation outcome until the current iteration. However, with UEI, each query point is evaluated considering its vicinity, so as we incrementally obtain more observations and get a better GP fit, we might observe that our optimum is actually located in an unsafe region. Thus, instead of considering the best observation outcome as the incumbent, we also apply the unscented transformation to select the incumbent at each iteration, based on the outcome at the sigma points of each query that belongs to the dataset of observations ($\mathcal{D}_n$). Obviously, we do not want to perform any additional evaluations of f because that would defeat the purpose of BO. Alternatively, we evaluate the sigma points with our estimation $\hat{y}$, which is the GP surrogate average prediction μ. Therefore, we define the unscented outcome as:

$$u(\mathbf{x}) = \sum_{i=0}^{2d} w^{(i)} \sum_{j=1}^{m} \mu_j(\mathbf{x}^{(i)}), \qquad \mathbf{x} \in \mathbf{X} \qquad (8)$$

where $\sum_{j=1}^{m} \mu_j(\mathbf{x}^{(i)})$ is the prediction of the GP according to Eq. (1), integrated over the kernel hyperparameters and at the sigma points of Eq. (5). Under these conditions, the incumbent for the UBO is defined as $\mathrm{y}_n^{\mathrm{ub}opt} = u(\mathbf{x}_n^{\mathrm{ub}opt})$, where $\mathbf{x}_n^{\mathrm{ub}opt} = \mathrm{argmax}_{\mathbf{x}_i \in \mathbf{x}_{1:n}} u(\mathbf{x}_i)$ is the optimal query until that iteration according to the unscented outcome. For further information on the performance of the UBO on synthetic functions, refer to [17].

IV. Grasp quality metric and collision penalty

In our problem the target function f is the quality of a given relative hand–object pose. Different grasp quality metrics have been defined in the literature [23], usually based on the contact points, torques and forces applied to the object. In our work, we use the Grasp Wrench Volume metric, introduced in [24] and defined as $f = \mathrm{Volume}(\mathcal{P})$, where $\mathcal{P}$ is the convex hull of the *Grasp Wrench Space* [25], [26]. In our context, starting the hand with an initial pose $\mathbf{x}$ (as in Fig. 1), the fingers are closed until touching the object, and the grasp quality metric is then computed.

We assume that approximate information about the object size and location is available, and is used to limit the exploration space. However, there are configurations that result in unfeasible grasps where the robot's hand collides with the object, even before attempting to close the hand. This indicates that the problem has constraints. Although there has been some recent work on BO with constraints [27], [28], we opted for the simpler approach of adding a penalty as described in [29]. This approach means that the input space remains unconstrained, improving the performance of the acquisition function's optimization. Additionally, due to kernel smoothing, we also get a safety area around the collision query where the function is only partly penalized.

Other research works on robotic grasping optimization had different approaches to deal with collisions. Some skip the collision query (*e.g.*, [30]) and others give it a grasp quality of zero (*e.g.*, [17]). In the first case, by ignoring the query we are losing information about the target function, hence not reducing uncertainty. In the second case, although uncertainty is reduced, the zero value is usually associated to a pose where the hand does not touch the object at all. Therefore, by modeling a collision with a zero value, one cannot distinguish a collision from a non-contact. Instead, we propose a penalization factor that will drive the search away from collision locations, ensuing a reduction of explored area and consequently leading to faster convergence.

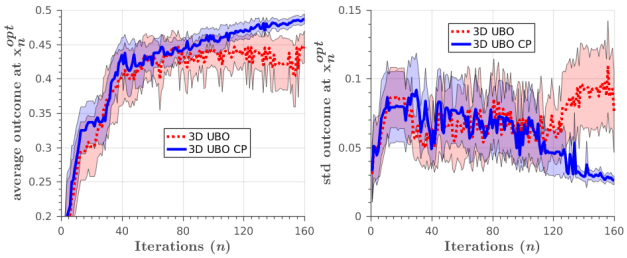
We propose a Collision Penalty (CP) that accounts for the level of penetration of the hand in the object. The CP is calculated by finding the number of joints in the robot's hand that collide with the object, $n_j \in \mathbb{N}$, which indicates a measure of penetration in the object:

$$\mathrm{CP}(n_j) = 1 - \mathrm{e}^{-\lambda n_j}, \qquad (9)$$

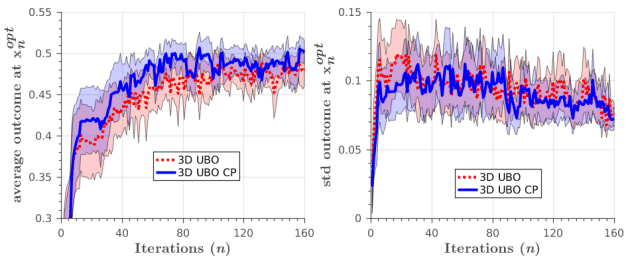
where λ is a tuning parameter used to smooth the penalty. The CP is an heuristic used only to improve the convergence speed. Therefore, during the optimization we redefine the target function as $f' = f + \mathrm{CP}$. Note that, in the evaluation process, we still resort to the original f.

V. Implementation

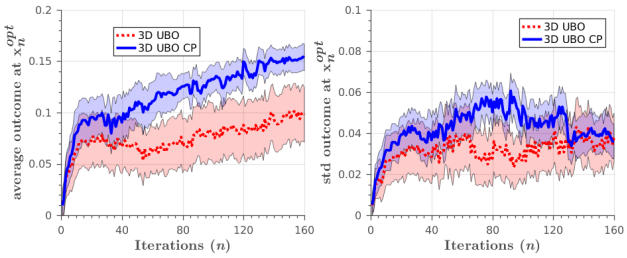
All the results were obtained from simulations using the Simox toolbox [31]. This toolbox simulates the iCub's hand grasping task arbitrary objects. At the beginning of each experiment, we define an initial pose for the hand and a motion trajectory for the finger joints. The hand is placed parallel to one of the object's facets with the thumb aligned



(a) **Glass: CP vs $\overline{\text{CP}}$** (UBO 3D).

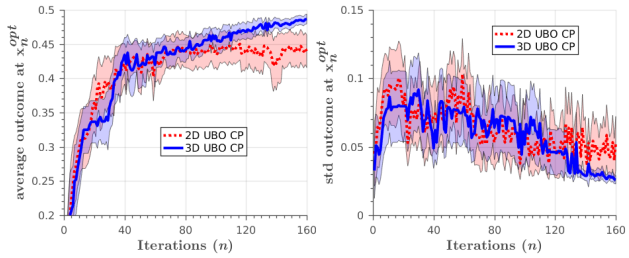


(b) **Bottle: CP vs $\overline{\text{CP}}$** (UBO 3D).

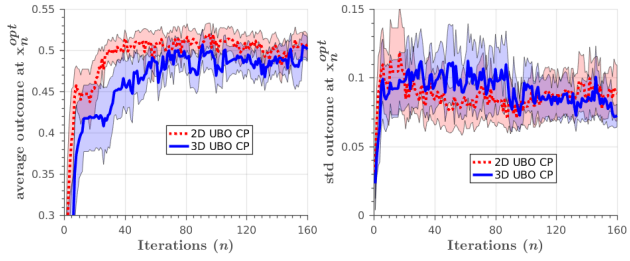


(c) **Mug: CP vs $\overline{\text{CP}}$** (UBO 3D).

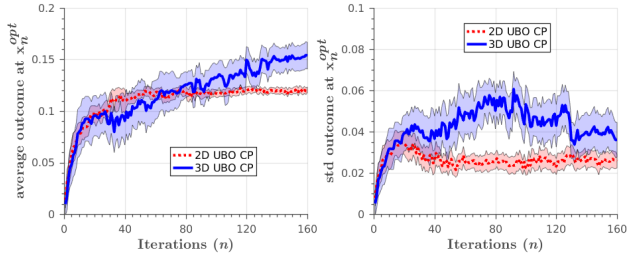
Fig. 2: **CP vs $\overline{\text{CP}}$** (UBO 3D). Left: expected outcome of current optimum $\bar{\text{y}}_{\text{mc}}(\mathbf{x}_n^{\text{ub}opt})$, Right: Variability of the outcome std$\left(\text{y}_{\text{mc}}(\mathbf{x}_n^{\text{ub}opt})\right)$. Best seen in color.



(a) **Glass: 2D vs 3D** (UBO CP).



(b) **Bottle: 2D vs 3D** (UBO CP)



(c) **Mug: 2D vs 3D** (UBO CP).

Fig. 3: **2D vs 3D** (UBO CP). Left: expected outcome of current optimum $\bar{\text{y}}_{\text{mc}}(\mathbf{x}_n^{\text{ub}opt})$, Right: Variability of the outcome std$\left(\text{y}_{\text{mc}}(\mathbf{x}_n^{\text{ub}opt})\right)$. Best seen in color.

with one of the neighbor facets. This defines the canonical pose of the hand with respect to the object. Then, at each trial (optimization step) a new hand pose is defined with respect to the canonical pose by incremental translations: $(\delta_x, \delta_y, \delta_z)$. All other parameters (*e.g.*, hand orientation, finger joints trajectories) are identical for all trials. In 2D optimization, we optimize (δ_x, δ_y), while δ_z is kept fixed. In 3D, we optimize $(\delta_x, \delta_y, \delta_z)$. For dimensions x and y, the bounds are set to the object's dimensions, as for z, which is the approach direction, the bounds extend from the surface of the object's facet to the plane where the hand is no longer able to touch the object.

At each trial, after the hand is placed in a new pose, the fingers joints move according to predefined motion trajectories until they touch the object. The trajectories are set so that the hand performs a power grasp on the object, in which all fingers are closed at the same time, following a movement synergy defined in previous work [32]. When the fingers motion is finished, the grasp quality metric defined in Sec. IV is computed. If a collision between the hand (either palm or fingers) and the object is detected when positioning the hand in the new pose, the CP is applied, and the grasping motion will not be executed. For the computation of the CP, the tuning parameter is set to $\lambda = 0.1$.

The *BayesOpt* library [20] is used to perform both BO and UBO. In the GP we use the Matérn kernel with $\upsilon = 5/2$. The input noise at each query point is assumed to be white Gaussian, $\mathcal{N}(0, I\sigma_{\mathbf{x}})$, with $\sigma_{\mathbf{x}} = 0.03$ (note that the input space was normalized in advance to the unit hypercube $[0, 1]^d$). We assume the grasp quality metric to be stochastic, due to small simulation errors and inconsistencies, thus we set $\sigma_\eta = 10^{-4}$.

VI. Experimental results

In this section, we start by describing the experiments performed to evaluate the benefits of CP (Sec. VI-A); then we compare UBO in 2D and 3D search problems (Sec. VI-B); and lastly we compare UBO with BO in the 3D case (Sec. VI-C).

To reproduce the effect of the input noise, we obtain Monte Carlo samples at the optimum in each iteration, $y_{\text{mc}}(\mathbf{x}_n^{opt})$, according to the input noise distribution $\mathcal{N}(0, I\sigma_{\mathbf{x}})$. Remember that $\mathbf{x}_n^{opt}$ corresponds to $\mathbf{x}_n^{\text{b}opt}$ when performing BO, and to $\mathbf{x}_n^{\text{ub}opt}$ for UBO. By analyzing the outcome of the samples we can estimate the expected outcome from the current optimum

$\bar{y}_{mc}(\mathbf{x}_n^{opt})$, and the variability of the outcome $\text{std}\left(y_{mc}(\mathbf{x}_n^{opt})\right)$. These metrics allow us to assess if the optimum belongs to a safe region. Indeed, if $\bar{y}_{mc}(\mathbf{x}_n^{opt})$ decreases over time (which cannot occur in classical BO) it should be correlated to the fact that the optimum ($\mathbf{x}_n^{bopt}$) is inside an unsafe area and is not a robust grasp.

For each experiment, we performed 20 runs of the robotic grasp simulation for all test objects. The robot hand posture for each object is initialized as shown in Fig. 1. Every time a new optimum is found, we collect 10 Monte Carlo samples at its location to get $y_{mc}(\mathbf{x}_n^{opt})$. Each run starts with 20 initial iterations using LHS, followed by 140 iterations of optimization. The shaded region in each plot represents a 95% confidence interval. All the quantitative results from each experiment, at its last iteration, are presented in Table I.

A. Benefits of CP

To assess the benefits of CP, we performed two types of experiments for each object, a 3D UBO with and without CP. As we can see in Fig. 2a and 2b, the addition of CP to the optimization process provides a boost in convergence speed for both the glass and the bottle. Also, by penalizing collisions we are reducing the regions that are worth exploring, meaning that the robot is actually able to find a better grasp at the end, both in terms of mean and variance.

The mug is the most challenging object to learn, since the optimization is performed on the mug's facet that includes the handle. In 3D, the handle is inside the search-space, leading to a large number of configurations that result in collisions, consequently undermining the convergence to the optimum. This is a situation where the CP really thrives. By penalizing these collisions, we are driving our search away from the inside of the handle and finding a safer grasp outside the handle. The results in Fig. 2c show how dramatic the improvement is, achieving higher mean values with great confidence level (*i.e.*, a smaller shaded region).

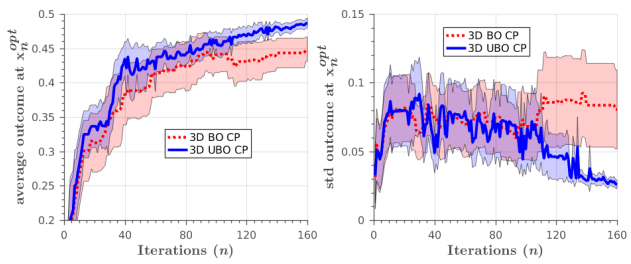
B. Generalization to 3D

We performed UBO with CP in both 2D and 3D to provide evidence that UBO generalizes well into a higher dimension space, *i.e.*, in 3D we only need a few extra evaluations to reach the same results obtained in 2D.

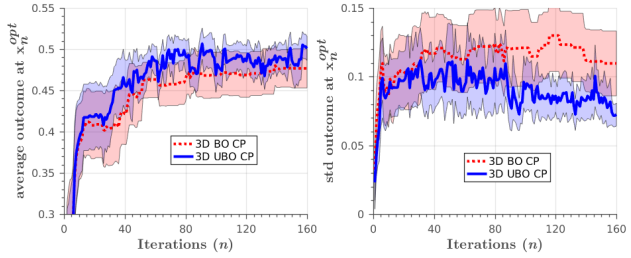
In Fig. 3a, we observe that for the glass, even though 2D reaches better mean values right after the learning starts (iteration 20), 3D is able to reach the same level around iteration 40 and proceeds to surpass it achieving better results. As for the bottle, in Fig. 3b, the mean value of the 3D case trails almost the whole process, only edging out the 2D results close to the end of the budget.

In the mug, Fig. 3c, the 3D optimization only manages to reach similar mean values at around iteration 65. As explained in Sec. VI-A, this is due to the high amount of queries that result in collisions when we are optimizing in a 3D search-space.

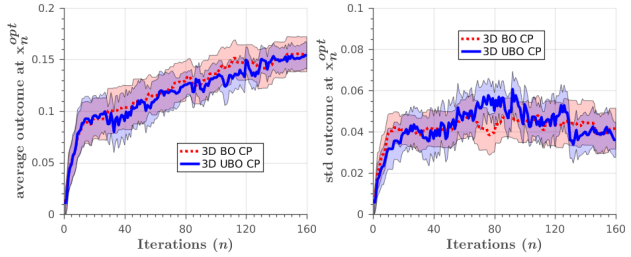
We must point out that the z coordinate in the 2D optimization was chosen to ensure a fair comparison with the 3D, setting it to the parallel plane where the optimal grasp should be. However, the better results obtained for both glass and mug in 3D indicate that the optimum z was somewhere else. As we can observe in Fig. 5a and Fig. 5c, the visual difference between the best grasps in 2D and 3D is not noticeable, even though 3D still achieves better results. Therefore, the generalization to the 3D search-space is arguably needed since a better grasp point was found during the 3D optimization. On the bottle object, the initial z coordinate was set closer to the optimum.



(a) **Glass: BO vs UBO** (3D CP).



(b) **Bottle: BO vs UBO** (3D CP).



(c) **Mug: BO vs UBO** (3D CP).

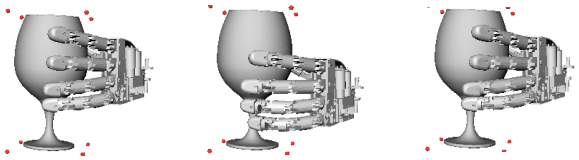
Fig. 4: **BO vs UBO** (3D CP). Left: expected outcome of current optimum $\bar{y}_{mc}(\mathbf{x}_n^{opt})$, Right: Variability of the outcome $\text{std}\left(y_{mc}(\mathbf{x}_n^{opt})\right)$. Best seen in color.

	Glass		Bottle		Mug	
	$\bar{y}_{mc}(\mathbf{x}_n^{opt})$	$\text{std}\left(y_{mc}(\mathbf{x}_n^{opt})\right)$	$\bar{y}_{mc}(\mathbf{x}_n^{opt})$	$\text{std}\left(y_{mc}(\mathbf{x}_n^{opt})\right)$	$\bar{y}_{mc}(\mathbf{x}_n^{opt})$	$\text{std}\left(y_{mc}(\mathbf{x}_n^{opt})\right)$
2D UBO CP	0.4396	0.0536	**0.5026**	0.0887	0.1205	**0.0256**
3D UBO $\overline{\text{CP}}$	0.4462	0.0761	0.4810	0.0754	0.0979	0.0378
3D UBO CP	**0.4867**	**0.0260**	0.5011	**0.0725**	**0.1567**	0.0361
3D BO CP	0.4489	0.0805	0.4767	0.1097	0.1551	0.0415

TABLE I: Results at the last iteration ($n = 160$) of the optimization process (means over all runs).

C. Advantages of UBO over BO

Here we compare UBO against BO in 3D using CP, to conclude if the advantages of UBO described in [17] in 2D, generalize to 3D.



(a) UBO 2D CP (b) BO 3D CP (c) UBO 3D CP

Fig. 5: Best grasps in one of the runs. The best grasp of UBO **2D** CP (a) is similar to the UBO **3D** CP (c). The best grasp achieved by **BO** is in an unsafe zone (b). The **UBO**'s best grasp is more robust to input noise (c). Check Table II for the grasp metrics in these configurations.

	UBO 2D CP	**BO 3D CP**	**UBO 3D CP**
y_n^{bopt} and y_n^{ubopt}	0.467	0.649	0.497
$\bar{\mathrm{y}}_{\mathrm{mc}}(\mathbf{x}_n^{opt})$	0.4424	0.383	0.472
$\mathrm{std}\left(\mathrm{y}_{\mathrm{mc}}(\mathbf{x}_n^{opt})\right)$	0.0471	0.207	0.0203

TABLE II: The grasp metric corresponding to the Monte Carlo sampled hand configurations shown in Fig. 5.

The results collected from both methods show that we are still able to learn safer grasps using UBO in 3D. The advantage is clear for both the glass, Fig. 4a, and bottle, Fig. 4b, where the UBO achieves higher mean values and lower variance. In the mug, Fig. 4c, we get competitive mean values using BO, but UBO finds an optimum with lower variance. The visual comparison between the two optimization strategies (BO and UBO) is displayed in Fig. 5b and 5c, where we can observe the best grasps achieved in one of the 20 runs.

VII. Conclusions

This work has validated the application of Unscented Bayesian Optimization to 3D grasp optimization. We show that it outperforms the classical Bayesian Optimization in this problem and generalizes well from the existing results in 2D search to a more challenging 3D search, without compromising the optimization budget. We propose a collision penalty function to force the search algorithm away from potential collision configurations, thus speeding up the convergence of the method. In future work we will study how to extend the method to the full 6D (translation+rotation) optimization and the application of the method in a real robotic anthropomorphic hand with 3D force sensors in the finger's phalanges [33].

References

[1] M. T. Mason and J. K. Salisbury, *Robot Hands and the Mechanics of Manipulation*. The MIT Press, Cambridge, MA, 1985, ch. Manipulator grasping and pushing operations.

[2] K. Shimoga, "Robot grasp synthesis algorithms: A survey," *Int. J. Rob. Res.*, vol. 15, no. 3, 1996.

[3] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Rob. Auton. Syst.*, vol. 60, no. 3, 2012.

[4] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - a survey," *IEEE Trans. Robot*, vol. 30, no. 2, April 2014.

[5] K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic, and R. Dillmann, "Grasping known objects with humanoid robots: A box-based approach," in *IEEE ICAR*, 2009.

[6] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka, "Rigid 3d geometry matching for grasping of known objects in cluttered scenes," *Int. J. Rob. Res.*, vol. 31, no. 4, Apr. 2012.

[7] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Şucan, *Towards Reliable Grasping and Manipulation in Household Environments*. Springer Berlin Heidelberg, 2014.

[8] L. Montesano and M. Lopes, "Active learning of visual descriptors for grasping using non-parametric smoothed beta distributions," *Rob. Auton. Syst.*, vol. 60, no. 3, 2012.

[9] R. Figueiredo, A. Shukla, D. Aragao, P. Moreno, A. Bernardino, J. Santos-Victor, and A. Billard, "Reaching and grasping kitchenware objects," in *Proc. Int. Symp. Syst. Integration*, 2012.

[10] A. S. Seungsu Kim and A. Billard, "Catching objects in flight," *IEEE Trans. Robot*, 2014.

[11] J. Stückler, M. Schwarz, M. Schadler, A. Topalidou-Kyniazopoulou, and S. Behnke, "Nimbro explorer: Semiautonomous exploration and mobile manipulation in rough terrain," *J. Field Robotics*, vol. 33, no. 4, 2015.

[12] D. Leidner, W. Bejjani, A. Albu-Schaeffer, and M. Beetz, "Robotic agents representing, reasoning, and executing wiping tasks for daily household chores," in *AAMAS*, 2016.

[13] D. Leidner, A. Dietrich, M. Beetz, and A. Albu-Schäffer, "Knowledge-enabled parameterization of whole-body control strategies for compliant service robots," *Auton Robots*, vol. 40, no. 3, 2016.

[14] P. Vicente, L. Jamone, and A. Bernardino, "Towards markerless visual servoing of grasping tasks for humanoid robots," in *IEEE ICRA*, 2017.

[15] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, "Online movement adaptation based on previous sensor experiences," in *IEEE/RSJ IROS*, 2011.

[16] O. Kroemer, R. Detry, J. Piater, and J. Peters, "Active learning using mean shift optimization for robot grasping," in *IEEE/RSJ IROS*, 2009.

[17] J. Nogueira, R. Martinez-Cantin, A. Bernardino, and L. Jamone, "Unscented Bayesian optimization for safe robot grasping," in *IEEE/RSJ IROS*, 2016.

[18] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *CoRR*, vol. abs/1012.2599, 2010.

[19] H. L. J. Snoek and R. Adams, "Practical Bayesian optimization of machine learning algorithms," *NIPS*, 2012.

[20] R. Martinez-Cantin, "BayesOpt: A Bayesian optimization library for nonlinear optimization, experimental design and bandits," *J. Mach. Learn. Res.*, vol. 15, 2014.

[21] A. D. Bull, "Convergence rates of efficient global optimization algorithms," *J. Mach. Learn. Res.*, vol. 12, 2011.

[22] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, March 2004.

[23] M. Roa and R. Suárez, "Grasp quality measures: review and performance," *Auton Robots*, vol. 38, 2015.

[24] A. T. Miller and P. K. Allen, "Examples of 3D grasp quality computations," in *IEEE ICRA*, 1999.

[25] N. S. Pollard, "Synthesizing grasps from generalized prototypes," in *IEEE ICRA*, 1996.

[26] C. Borst, M. Fischer, and G. Hirzinger, "a fast and robust grasp planner for arbitrary 3d objects," in *IEEE ICRA*.

[27] J. Gardner, M. Kusner, Zhixiang, K. Weinberger, and J. Cunningham, "Bayesian optimization with inequality constraints," in *ICML*, 2014.

[28] J. S. Michael A. Gelbart and R. P. Adams, "Bayesian optimization with unknown constraints," in *Uncertain Artif Intell*, 2014.

[29] R. Martinez-Cantin, "Funneled bayesian optimization for design, tuning and control of autonomous systems," *Trans. Cybern.*, 2018.

[30] P. Allen, M. Ciocarlie, and C. Goldfeder, "Grasp planning using low dimensional subspaces," in *The Human Hand as an Inspiration for Robot Hand Development*, V. S. R. Balasubramanian, Ed. Springer, Cham, 2014.

[31] N. Vahrenkamp, M. Kröhnert, S. Ulbrich, T. Asfour, G. Metta, R. Dillmann, and G. Sandini, *Simox: A Robotics Toolbox for Simulation, Motion and Grasp Planning*. Intelligent Autonomous Systems 12. Advances in Intelligent Systems and Computing, 2013.

[32] A. Bernardino, M. Henriques, N. Hendrich, and J. Zhang, "Precision grasp synergies for dexterous robotic hands," in *IEEE ROBIO*, 2013.

[33] T. Paulino, P. Ribeiro, M. Neto, S. Cardoso, A. Schmitz, J. Santos-Victor, A. Bernardino, and L. Jamone, "Low-cost 3-axis soft tactile sensors for the human-friendly robot Vizzy," in *IEEE ICRA*, 2017.