Unscented Bayesian Optimization for Safe Robot Grasping

José Nogueira[1] Ruben Martinez-Cantin[2] Alexandre Bernardino[1] Lorenzo Jamone[1]

Abstract— **Safe and robust grasping of unknown objects is a major challenge in robotics, which has no general solution yet. A promising approach relies on haptic exploration, where active optimization strategies can be employed to reduce the number of exploration trials. One critical problem is that certain optimal grasps discoverd by the optimization procedure may be very sensitive to small deviations of the parameters from their nominal values: we call these unsafe grasps because small errors during motor execution may turn optimal grasps into bad grasps. To reduce the risk of grasp failure, safe grasps should be favoured. Therefore, we propose a new algorithm, unscented Bayesian optimization, that performs efficient optimization while considering uncertainty in the input space, leading to the discovery of safe optima. The results highlight how our method outperforms the classical Bayesian optimization both in synthetic problems and in realistic robot grasp simulations, finding robust and safe grasps after a few exploration trials.**

I. Introduction

Learning how to grasp unknown objects can be performed in two different ways. One strategy relies on gathering extensive amounts of data from multiple sensors and learn features that allow grasping generalization. Those features are mapped to optimal grasp configurations and strategies. However, even for a simple gripper, the amount of data required for proper generalization is very large [1]. A more natural alternative is on-the-fly learning by trial and error [2]. This can be achieved with minimal visual or haptic input and it naturally generalize to multiple objects and tasks. However, on-the-fly learning has two problems. First, it can be expensive. Thus, we need an efficient grasping exploration methodology. Second, it can get lucky. We need to guarantee that the optimal grasp can be repeated in the presence of noise with sufficient quality. A brute-force approach would need to test grasps in many different configurations in search for the best grasping point and, for each configuration, repeat the test many times to average out the robot positioning uncertainty. This is clearly unfeasible in practice, and better search strategies must be devised.

In this paper, we follow the trial and error methodology for grasping and directly address the two problems. First, given an object, we must find the configuration that maximize grasp quality with a small budget. Second, how do we incorporate repeatability errors in the grasp configuration such as optimality is preserved without performing multiple trials. In robotics in general and, for the problem of robot grasping in particular, Bayesian optimization has been one of the most successful and efficient trial-and-error techniques [2], [3], [4], [5], even in the presence of mechanical failures [6]. In this setup, repeatability errors correspond to uncertainty in the input space.

Bayesian optimization [7], [8] is a global optimization technique for black-box functions. Because it is designed for sample efficiency, at the cost of extra computation, it is intended for functions that are expensive to evaluate (in terms of cost, energy, time). The beauty of Bayesian optimization is its capability to deal with general black-box functions, therefore being able to address the grasping problem without any extra information, just the results from previous trials. Bayesian optimization relies on a probabilistic surrogate function (e.g. a Gaussian process) that is able to learn about the target function based on previous samples and, therefore, drive future sampling more efficiently. However, to the authors knowledge, the consideration of uncertainty in the input space has been addressed neither in the grasp planning literature nor in the Bayesian optimization literature. There has been previous works that consider input noise in Gaussian process regression [9], however, those methods propagate the input noise to the output space, which may result in unnecessary exploration of the space for the optimization problem. Safe exploration has also been recently addressed within Bayesian optimization, but in that case, the problem is to guarantee that the outcome is above a threshold for each trial [10].

The main contribution of the paper addresses the problem of input noise in Bayesian optimization, which is then applied to robot grasping. For dealing with the input noise, we need a system to propagate the noise distribution from the input query through all the models and decisions of our method. We solve this with the *unscented transformation* [11], [12], a method to estimate the results of applying a nonlinear transformation to a probability distribution.

In this paper, we present the *unscented Bayesian optimization* (UBO) algorithm. It has the advantages of the sample efficiency from Bayesian optimization and the capability of dealing with input noise during function queries. Applied to grasping, this means that the method can find the optimal grasp while considering the input noise for safety. Furthermore, due to the recent popularity of Bayesian optimization in many areas (e.g. autonomous algorithm tuning [13], robot planning [14], [15], control [16], [17], reinforcement learning [18], [5], sensor networks [19], etc.), this method can directly

*This work was partially supported by the EU project LIMOMAN [PIEF-GA-2013-628315], Portuguese project FCT [UID/EEA/50009/2013] and Spanish projects DPI2015-65962-R (MINECO/FEDER, UE) and CUD2013-05 (SIRENA).

[1] Institute for Systems and Robotics, Instituto Superior Tecnico, Universidade de Lisboa, Portugal josemscnogueira@gmail.com, alex@isr.tecnico.ulisboa.pt, ljamone@isr.tecnico.ulisboa.pt

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

impact many other fields, that would greatly benefit from an extension to deal with input noise.

II. Bayesian optimization

Consider the problem of finding the optimum (e.g. minimum) of an unknown real valued function $f : \mathbb{X} \to \mathbb{R}$, where $\mathbb{X}$ is a compact space, $\mathbb{X} \subset \mathbb{R}^d, d \geq 1$, with a maximum budget of N evaluations of the target function f. The Bayesian optimization algorithm selects the best query points at each iteration so that the optimization gap $|y^* - y_n|$ is minimum for the available budget. This is achieved by using two ingredients. First, a probabilistic surrogate model: a distribution over the family of functions $P(f)$, where the target function $f()$ belongs, built incrementally using the sample evaluations. Second, a Bayesian decision process, that uses the information captured in the surrogate model to select the next query point in order to maximize the information about the optimum. Therefore, Bayesian optimization can be seen as an active learning approach to find the optimum. Without loss of generality, in the remainder of the paper we assume that the surrogate model $P(f)$ is a Gaussian process $\mathcal{GP}(\mathbf{x}|\mu, \sigma^2, \boldsymbol{\theta})$ with inputs $\mathbf{x} \in \mathbb{X}$, scalar outputs $y \in \mathbb{R}$ and an associated kernel or covariance function $k(\cdot, \cdot)$ with hyperparameters $\boldsymbol{\theta}$. The hyperparameters are estimated using a Monte Carlo Markov Chain (MCMC) algorithm, i.e.: slice sampling [13], [20], resulting in m samples $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i\}_{i=1}^m$.

Given at step n a dataset of query points $\mathbf{X} = \{\mathbf{x}_{1:n}\}$ and its respective outcomes $\mathbf{y} = \{y_{1:n}\}$, then the prediction of the Gaussian process at a new query point $\mathbf{x}_q$, with kernel k_i conditioned on the i-th hyperparameter sample $k_i = k(\cdot, \cdot|\boldsymbol{\theta}_i)$ is normally distributed, $\widehat{y}(\mathbf{x}_q) \sim \sum_{i=1}^m \mathcal{N}(\mu_i, \sigma_i^2|\mathbf{x}_q)$, where:

$$\begin{aligned}\mu_i(\mathbf{x}_q) &= \mathbf{k}_i(\mathbf{x}_q, \mathbf{X})\mathbf{K}_i^{-1}\mathbf{y}\\ \sigma_i^2(\mathbf{x}_q) &= k_i(\mathbf{x}_q, \mathbf{x}_q) - \mathbf{k}_i(\mathbf{x}_q, \mathbf{X})\mathbf{K}_i^{-1}\mathbf{k}_i(\mathbf{X}, \mathbf{x}_q)\end{aligned} \tag{1}$$

The vector $\mathbf{k}_i(\mathbf{x}_q, \mathbf{X})$ is the cross-correlation of the query point $\mathbf{x}_q$ with respect to the dataset $\mathbf{X}$ and $\mathbf{K}_i = \mathbf{K}_i(\mathbf{X}, \mathbf{X}) + \mathbf{I}\sigma_n^2$ is the Gram matrix corresponding to kernel k_i for the dataset $\mathbf{X}$, with noise variance σ_n^2. The noise term represents the observation noise in stochastic functions [21] or the nugget term for surrogate missmodeling [22]. Note that, because we use a sampling distribution of $\boldsymbol{\theta}$ the predictive distribution at any point $\mathbf{x}$ is a mixture of Gaussians.

To select the next point at each iteration, we use the *expected improvement* criterion [23] as a way to minimize the optimality gap. The expected improvement is the expectation of the improvement function $I(\mathbf{x}) = \max(0, \rho - f(\mathbf{x}))$, where ρ is an incumbent value, usually the *best outcome* until that iteration y_{best} or, for stochastic functions, the best average prediction $\widehat{y}_{best}$. Taking the expectation over the mixture of Gaussians of the predictive distribution, we can compute the expected improvement as:

$$\begin{aligned}EI(\mathbf{x}) &= \mathbb{E}_{p(y|\mathbf{x},\boldsymbol{\theta})}\left[\max(0, \rho - f(\mathbf{x}))\right]\\ &= \sum_{i=1}^m \left[(\rho - \mu_i)\,\Phi(z_i) + \sigma_i \phi(z_i)\right]\end{aligned} \tag{2}$$

where ϕ and Φ are the corresponding Gaussian probability density function (PDF) and cumulative density function (CDF) and $z_i = (\rho - \mu_i)/\sigma_i$. In this case, (μ_i, σ_i^2) is the prediction computed with Equation (1).

Finally, in order 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 [24].

III. Unscented Bayesian optimization

In this paper, we propose to consider the input noise during the decision process to explore and select the regions that are safe. That is, the regions that guarantee good results even if the experiment/trial is repeated several times. In this respect, our contribution is twofold: we present the *unscented expected improvement* (Sec. III-B) and the *unscented optimum incumbent* (Sec. III-C). Both methods are based on the *unscented transformation* (Sec. III-A), initially developed for tracking and filtering applications [25], [11].

A. Unscented transformation

The unscented transformation is a method to propagate probability distributions through nonlinear transformations with a trade off of computational cost vs accuracy. It is based on the principle that *it is easier to approximate a probability distribution than to approximate an arbitrary nonlinear function* [11]. The unscented transformation uses a set of deterministically selected samples from the original distribution (called *sigma points*) and transform them through the nonlinear function $f(\cdot)$. Then, the transformed distribution is computed based on the weighted combination of the transformed sigma points.

The advantage of the unscented transformation is that the mean and covariance estimates of the new distribution are accurate to the third order of the Taylor series expansions of $f(\cdot)$ provided that the original distribution is a Gaussian prior, or up to the second order of the expansion for any other prior. Fig. 1 highlights the differences between approximating the distribution using sigma points or using standard first-order Taylor linearization. The distribution from the UT is closer to the real distribution. Because the prior and posterior distributions are both Gaussians, the unscented transformation is a linearization method. However, because the linearization is based on the statistics of the distribution, it is often found in the literature as *statistical linearization*.

Another advantage of the unscented transformation is its computational cost. For a d-dimensional input space, the unscented transformation requires a set of $2d + 1$ sigma points. Thus, the computational cost is negligible compared to other alternatives to Bayesian approximation such as Monte Carlo, which requires a large number of samples, or numerical integration such as *Gaussian quadrature*, which has an exponential cost on d. Van der Merwe [12] proved that the unscented transformation is part of the more general *sigma point filters*, which achieve similar performance results.

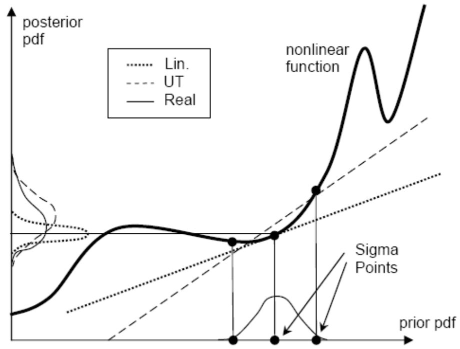


Fig. 1: Propagation of a normal distribution through a nonlinear function. The first order Taylor expansion (dotted) only uses information of the function at the mean point to compute the linear approximation, while the UT (dashed) approaches the function with a linear regression of several sigma points. The actual distribution is the solid one. (Adapted from [12])

1) Computing the unscented transformation: Assuming that the prior distribution is a Gaussian distribution $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \Sigma_{\mathbf{x}})$, then the $2d+1$ sigma points of the unscented transformation are computed by

$$\begin{aligned} \mathbf{x}^0 &= \bar{\mathbf{x}} \\ \mathbf{x}_+^{(i)} &= \bar{\mathbf{x}} + \left(\sqrt{(d+k)\Sigma_{\mathbf{x}}}\right)_i \qquad \forall\, i = 1 \dots d \\ \mathbf{x}_-^{(i)} &= \bar{\mathbf{x}} - \left(\sqrt{(d+k)\Sigma_{\mathbf{x}}}\right)_i \qquad \forall\, i = 1 \dots d \end{aligned} \tag{3}$$

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 optimal values of k, see [11].For these sigma points, the weights are defined as:

$$\begin{aligned} \omega^0 &= \frac{k}{d+k} \\ \omega_+^{(i)} &= \frac{1}{2(d+k)} \qquad \forall\, i = 1 \dots d \\ \omega_-^{(i)} &= \frac{1}{2(d+k)} \qquad \forall\, i = 1 \dots d \end{aligned} \tag{4}$$

Then, the transformed distribution is $\mathbf{x}' \sim \mathcal{N}(\bar{\mathbf{x}'}, \Sigma'_{\mathbf{x}})$, where:

$$\bar{\mathbf{x}'} = \sum_{i=0}^{2d} \omega^{(i)} f(\mathbf{x}^{(i)}) \tag{5}$$

B. Unscented expected improvement

Bayesian optimization is about selecting the most interesting point at each iteration. Usually, this is achieved by a greedy criterion, such as the *expected improvement*, the *upper confidence bound* or the *predictive entropy*. These criteria, also denoted *acquisition functions*, select the query point that has the higher potential to become the optimum, assuming that the query is deterministic. However, in our case, the query is a probability distribution due to input noise. Thus, instead of analysing the outcome of the criterion, we are going to analyse the resulting posterior distribution of transforming the query distribution through the acquisition function.

For the purpose of safe Bayesian optimization, we will use the expected value of the transformed distribution as the acquisition function. Therefore, we define the unscented expected improvement as:

$$UEI(\mathbf{x}) = \sum_{i=0}^{2d} \omega^{(i)} EI(\mathbf{x}^{(i)}) \tag{6}$$

where $\mathbf{x}^{(i)}$ and $\omega^{(i)}$ are computed according to equations (3) and (4) respectively. The expected value of the transformed distribution $\bar{\mathbf{x}'} = UEI(\mathbf{x})$ is enough to take a decision considering the risk on the input noise. Anyway, the value of $\Sigma'_{\mathbf{x}}$ represents the output uncertainty and can also be used as meta-analysis tool.

C. Unscented optimal incumbent

The unscented expected improvement can be used to drive the search procedure towards safe regions. However, because the target function is unknown by definition, the sampling procedure can still query good outcomes in unsafe areas. Furthermore, in Bayesian optimization there is a final decision that is independent of the acquisition function employed. Once the optimization process is stopped after sampling N queries, we still need to decide which point is the *best*. Moreover, after every iteration, we need to say which point is the incumbent. If the final decision about the incumbent selects the sample with best outcome $\mathbf{x}^*$ such that $y_{best} = f(\mathbf{x}^*)$ we may select an unsafe query. Instead, we propose to apply the unscented transformation also to the select the optimal incumbent $\mathbf{x}^*$, based on the function outcome $f()$ at the sigma points. This would require additional evaluations of $f()$, but the main idea of Bayesian optimization is to reduce the number of evaluations on $f()$. Instead of evaluating $f()$ at the sigma points, we evaluate the sigma points at the GP surrogate average prediction $\mu()$.

Therefore, we define the unscented outcome (UO) as:

$$UO(\mathbf{x}) = \sum_{i=0}^{2d} \omega^{(i)} \sum_{j=1}^{m} \mu_j(\mathbf{x}^{(i)}) \tag{7}$$

where $\sum_{j=1}^{m} \mu_j(\mathbf{x}^{(i)})$ is the prediction of the GP according to equation (1) integrated over the kernel hyperparameters and at the sigma points of equation (3). Under these conditions, the incumbent of the optimal solution $\mathbf{x}^*$ corresponds to:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} UO(\mathbf{x}) \tag{8}$$

As an illustrative example of the unscented Bayesian optimization process, take the RKHS function in Fig. 2. In this case, the maximum of the function is at $x \approx 0.87$. However, this maximum is very risky, that is, small variations in x result in large deviations from the optimal outcome. On the other hand, the local maximum at $x \approx 0.07$ is safer. Even if there is noise in x, repeated queries will produce similar outcomes. In this case, if we assume input noise of $\sigma_x = 0.05$ and compute the unscented transformation

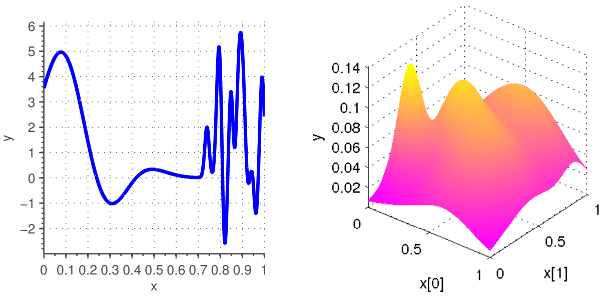


Fig. 2: Left: RKHS function from [26]. Right: Gaussian mixture (GM) function

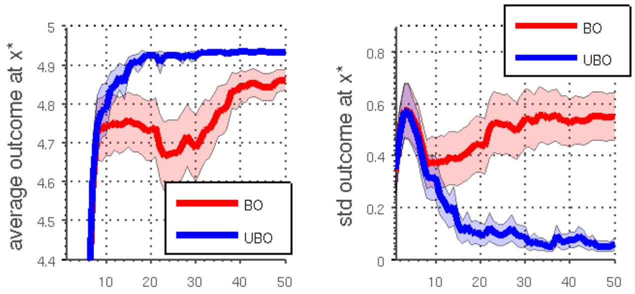


Fig. 3: RKHS Results. Left: Expected outcome at the current optimum $\bar{y}_{mc}\left(\mathbf{x}^{*}\right)$, Right: Variability of the outcome $std\left(y_{mc}\left(\mathbf{x}^{*}\right)\right)$

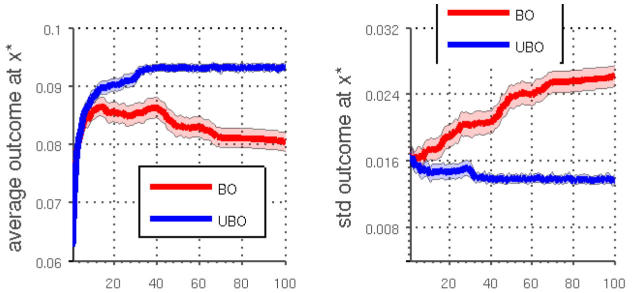


Fig. 4: Gaussian mixture Results. Left: Expected outcome at the current optimum $\bar{y}_{mc}\left(\mathbf{x}^{*}\right)$, Right: Variability of the outcome $std\left(y_{mc}\left(\mathbf{x}^{*}\right)\right)$

of that noise through the function, we can see that the sigma points centered at the leftmost maximum would have higher outcome that the sigma points centered at the global maximum. Therefore, the expected posterior value of the local smooth maximum would be larger than the value at the global narrow maximum.

IV. Results

In this section we describe the experiments performed to evaluate the benefits of the unscented Bayesian optimization (UBO) with respect to the classical expected Bayesian optimization (BO). The main goal is to demonstrate that, by using the UBO, we minimize the risk of choosing unsafe global optima. We first illustrate the method applied to synthetic functions, to clearly visualize the importance of selecting safe optima. Then, we show the results of doing autonomous exploration of daily life objects with a dexterous robot hand using realistic simulations, reproducing the conditions of a real robot setup.

In this work we have used and extended the BayesOpt software [20] with the proposed methods. For the GP kernel, we used the standard choice of the Matérn kernel with $\nu = 5/2$. Without loss of generality, we assume that the input noise is white Gaussian, isotropic and stationary, i.e. $\mathcal{N}(0, \mathbf{I}\sigma_x)$, although the method can be applied to anisotropic, nonstationary and even non-Gaussian noise.

To reproduce the effect of the input noise, we queried the result of each method using Monte Carlo samples according the input noise distribution at the incumbent point at each iteration $\{y_{mc}^{(i)}(\mathbf{x}^*)\}$. By analysing the outcome of the samples we can estimate the expected outcome from the current optimum $\bar{y}_{mc}\left(\mathbf{x}^{*}\right)$ and the variability of outcomes $std\left(y_{mc}\left(\mathbf{x}^{*}\right)\right)$. As we can see in the results, our method is able to provide equal or better expected outcomes while reducing the variability of those same outcomes.

A. Synthetic Functions

In this section, we use two synthetic functions with distinct regions in terms of risk: the 1D RKHS from [26] and a 2D Gaussian mixture model, see Fig. 2. Both have a global maximum at a narrow peak, which represents a region of high risk.

We performed 100 runs of each optimization procedure (BO and UBO) for each function (RKHS, GM). We used 100 Monte Carlo samples for $\{y_{mc}^{(i)}(\mathbf{x}^*)\}$. For RKHS each run has 5 initial samples with LHS and the optimization performs 45 iterations. The input noise is set as $\sigma_x = 0.01$. For GM each run has 30 initial samples and the optimization performs 90 iterations. The input noise is set as $\sigma_x = 0.1$. In Fig. 3 and Fig. 4 we show the statistics over the different runs for the evaluation criteria, as a function of the number of iterations. The shaded region represents the 95% confidence interval. Being deterministic functions, we used $\sigma_n = 10^{-6}$ as nugget in both cases.

For both functions, we can observe that UBO quickly overcomes the results of BO. UBO computes less risky solutions, as demonstrated by the higher expected return value and lower standard deviation. In Table I we show the numeric results obtained at the last iteration, as well as the values of the worst sample of the Monte Carlo runs. The worst case for UBO is always more favourable than the worst case for BO by a large margin.

B. Robot Grasp Simulations

We use the Simox simulation toolbox for robot grasping [27]. This toolbox simulates the iCub robot hand grasping arbitrary objects. Given an initial pose for the robot hand and a finger joint trajectory, the simulator runs until the fingers are in contact with the object surface and computes a grasp

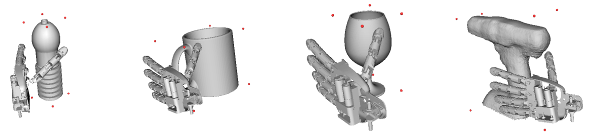


Fig. 5: Objects used in the simulations with corresponding initial robot hand configuration. Left to right: bottle, mug, glass and drill.

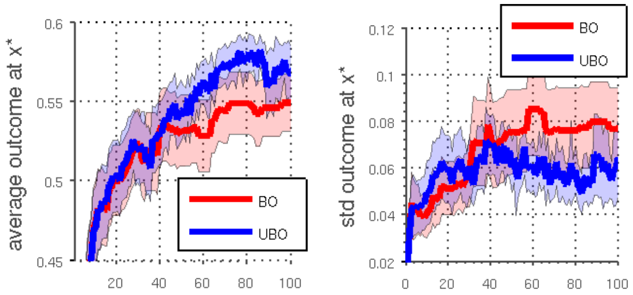


Fig. 6: Bottle. Left: Expected outcome at the current optimum $\bar{y}_{mc}(\mathbf{x}^*)$, Right: Variability of the outcome $std(y_{mc}(\mathbf{x}^*))$

quality metric based on wrench space analysis. We use a representation of the iCub left hand which can move freely in space and a few static objects (see Fig. 5). The robot hand is initially placed with the palm facing parallel to one of the facets, at a fixed distance of the object bounding box, and the thumb aligned with one of the neighbour facets. The hand pose is then defined with respect to the initial pose by incremental translations and rotations: $(\delta_x, \delta_y, \delta_z, \theta_x, \theta_y, \theta_z)$. In the reported experiments we optimize the translation parallel to the facet (δ_x, δ_y), while the other values are fixed. The translation variables are bound to the limits of the bounding box. A power grasp synergy was adopted for the hand closure.

We performed 30 runs of the robotic grasp simulation for each object and each optimization criterion. The robot hand posture with respect to the objects was initialized as shown in Fig. 5. We used 20 samples to compute $\{y_{mc}^{(i)}(\mathbf{x}^*)\}$.

Each run starts with 40 initial samples with LHS and proceeds with 60 iterations of optimization. We assume the grasp quality metric to be stochastic, due to small simulation errors and inconsistencies, and we set $\sigma_n = 10^{-4}$. Also, we assume an input noise $\sigma_x = 0.03$ (note that the input space was normalized in advance to the unit hypercube $[0,1]^d$). The results are shown in Fig. 6, 7, 8 and 9.

It can be seen that, for the bottle and glass, the UBO method has clear advantages over BO. UBO obtains higher mean values and lower standard deviations. For the drill, UBO eventually overcomes BO after few iterations, which might imply that the unsafe optimum is difficult to find, but still exists. Looking at the quantitative results shown in Table I, we can see that, at the end of the optimization, UBO is better than BO in all criteria, except for the mean output value for the mug. For the mug, the 100 trials are not enough to obtain better mean values. We can see that the mug and drill objects are more challenging due to their non-rotational symmetry. Since the optimization is only done in translation parameters, the method is missing exploration in the rotation degrees of freedom. Furthermore, in the mug case, the facet chosen was the one that contains the handle. Trying to learn a grasp in this setting is much harder than the other cases since, for the same input space volume, the number of configurations which return a good metric is much smaller. This deteriorates GP regression and hinders Bayesian optimization performance in general. An alternative in this case would be to use a nonstationary kernel [28].

In Fig. 10 we illustrate four grasps at the water bottle

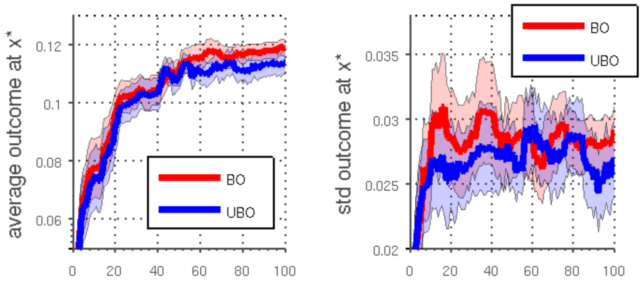


Fig. 7: Mug. Left: Expected outcome at the current optimum $\bar{y}_{mc}(\mathbf{x}^*)$, Right: Variability of the outcome $std(y_{mc}(\mathbf{x}^*))$

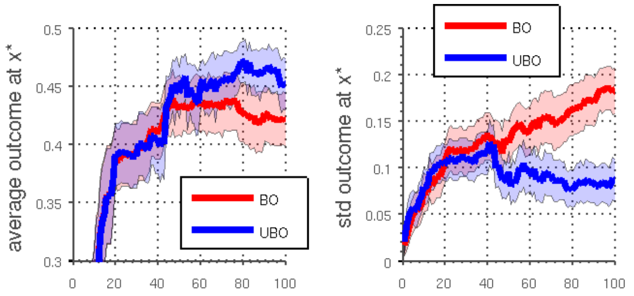


Fig. 8: Glass. Left: Expected outcome at the current optimum $\bar{y}_{mc}(\mathbf{x}^*)$, Right: Variability of the outcome $std(y_{mc}(\mathbf{x}^*))$

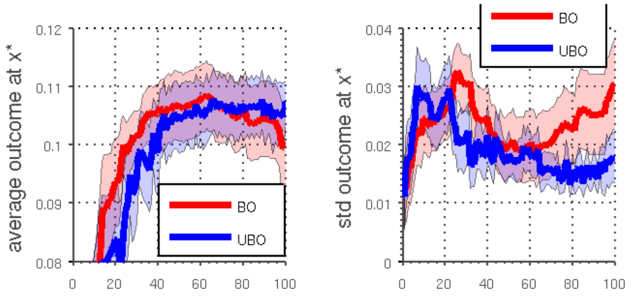
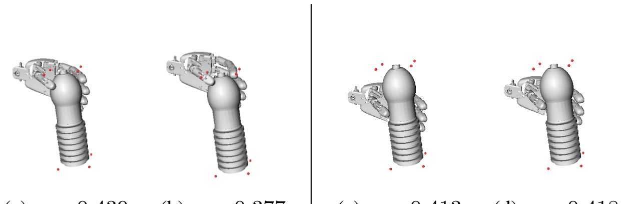


Fig. 9: Drill. Left: Expected outcome at the current optimum $\bar{y}_{mc}(\mathbf{x}^*)$, Right: Variability of the outcome $std(y_{mc}(\mathbf{x}^*))$



(a) $y = 0.439$ (b) $y = 0.377$ (c) $y = 0.413$ (d) $y = 0.418$

Fig. 10: Grasp safety. In this example the best grasp is at an unsafe zone (a), because some bad grasps can be found in its vicinity (b). The unscented Bayesian optimization chooses grasps with lower risk at a safe zone (c) and (d), where performance is robust to input noise.

	$\bar{y}_{mc}(\mathbf{x}^*)$		worst $y_{mc}(\mathbf{x}^*)$		$std(y_{mc}(\mathbf{x}^*))$	
Exp.	BO	UBO	BO	UBO	BO	UBO
RKHS	4.863	**4.934**	2.881	**4.657**	0.554	**0.065**
GM	0.080	**0.093**	0.023	**0.053**	0.027	**0.014**
Bottle	0.550	**0.567**	0.390	**0.430**	0.077	**0.065**
Mug	**0.119**	0.114	0.051	**0.059**	0.029	**0.027**
Glass	0.421	**0.452**	0.080	**0.252**	0.184	**0.087**
Drill	0.101	**0.108**	0.050	**0.068**	0.030	**0.018**

TABLE I: Results at the last iteration of the Bayesian optimization process (means over all runs). In this case, *worst* represents the sample with worst outcome y_{mc}.

explored during the experiments. Two of the grasps are performed in a safe region while the two other are explored at a unsafe region. Although the unsafe zone has one observation with the highest value, it has also higher risk of getting a low value observation in its vicinity.

V. Conclusion

The contribution of this paper is twofold. On the one hand, we present a method for *robust and safe grasping* of unknown objects by haptic exploration; because the process is based on general black-box optimization, the method is capable to deal with arbitrary objects, effectors and environmental conditions. On the other hand, we have developed a novel technique for *Bayesian optimization in the presence of input noise*, that we have called unscented Bayesian optimization. The potential interest of this method goes beyond grasping or even robotics. Bayesian optimization is currently being used in many applications: engineering, computer sciences, economics, simulations, experimental design, biology, artificial intelligence, etc. In all those fields, there are many situations where input noise or uncertainty may arise, and in which safe optimization is therefore fundamental. For example, in other areas of robotics, it might be used for navigation, planing or sensor placement, as the robot/sensor location is uncertain.

References

[1] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Rob. Res.*, vol. 34, no. 4-5, pp. 705–724, 2015.

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

[3] ——, "Combining active learning and reactive control for robot grasping," *Robotics and Autonomous systems*, vol. 58, no. 9, pp. 1105–1116, 2010.

[4] A. Boularias, J. A. Bagnell, and A. Stentz, "Efficient optimization for autonomous robotic manipulation of natural objects." in *AAAI*, 2014, pp. 2520–2526.

[5] C. Daniel, O. Kroemer, M. Viering, J. Metz, and J. Peters, "Active reward learning with a novel acquisition function," *Autonomous Robots*, vol. 39, no. 3, pp. 389–405, 2015.

[6] A. Cully, J. Clune, D. Tarapore, and J. B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, p. 503507, 2015.

[7] D. Jones, M. Schonlau, and W. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, December 1998.

[8] E. Brochu, V. Cora, and N. Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," 2010, arXiv:1012.2599v1.

[9] J. McHutchon and C. Rasmussen, "Gaussian process training with input noise," in *Advances in Neural Information Processing Systems 24*, 2011, pp. 1341–1349.

[10] F. Berkenkamp, A. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2016, pp. 491–496.

[11] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, March 2004.

[12] R. van der Merwe, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," Ph.D. dissertation, OGI School of Science & Engineering, Oregon Health & Science University, 2004.

[13] J. Snoek, H. Larochelle, and R. Adams, "Practical Bayesian optimization of machine learning algorithms," in *NIPS*, 2012, pp. 2960–2968.

[14] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6136–6143.

[15] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet, "A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot." *Autonomous Robots*, vol. 27, no. 3, pp. 93–103, 2009.

[16] M. Tesch, J. Schneider, and H. Choset, "Adapting control policies for expensive systems to changing environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[17] R. Calandra, A. Seyfarth, J. Peters, and M. Deisenroth, "Bayesian optimization for learning gaits under uncertainty," *Annals of Mathematics and Artificial Intelligence (AMAI)*, vol. 1 1, pp. 1–19 1–19, 2015.

[18] R. Martinez-Cantin, N. de Freitas, A. Doucet, and J. A. Castellanos, "Active policy learning for robot planning and exploration under uncertainty," in *Robotics: Science and Systems*, 2007.

[19] R. Garnett, M. A. Osborne, and S. J. Roberts, "Bayesian optimization for sensor set selection," in *Information Processing in Sensor Networks (IPSN 2010)*, 2010.

[20] R. Martinez-Cantin, "BayesOpt: A Bayesian optimization library for nonlinear optimization, experimental design and bandits," *Journal of Machine Learning Research*, vol. 15, pp. 3735–3739, 2014.

[21] D. Huang, T. T. Allen, W. I. Notz, and N. Zheng, "Global optimization of stochastic black-box systems via sequential kriging meta-models," *Journal of Global Optimization*, vol. 34, no. 3, pp. 441– 466, 2006.

[22] R. B. Gramacy and H. K. H. Lee, "Cases for the nugget in modeling computer experiments," *Statistics and Computing*, vol. 22, pp. 713–722, 2012.

[23] J. Mockus, *Bayesian Approach to Global Optimization*, ser. Mathematics and Its Applications. Kluwer Academic Publishers, 1989, vol. 37.

[24] A. D. Bull, "Convergence rates of efficient global optimization algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2879–2904, 2011.

[25] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000.

[26] J. Assel, Z. Wang, B. Shahriari, and N. Freitas, "Heteroscedastic treed Bayesian optimizaion," 2015, arXiv:1410.7172v2.

[27] V. Vahrenkamp, "Simox: A lightweight simulation and motion planning toolbox for C++," http://simox.sourceforge.net/.

[28] R. Martinez-Cantin, "Locally-biased Bayesian optimization using nonstationary Gaussian processes," in *Neural Information Processing Systems (NIPS) workshop on Bayesian Optimization*, 2015.