

Optimización bayesiana multisolución para la exploración eficiente de agarres robóticos

Ignacio Herrera Seara, Juan García-Lechuz Sierra, Javier García Barcos, Rubén Martínez Cantín
 Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza
 {756920, 736161, jgbarcos, rmcantin}@unizar.es

Resumen

El uso de brazos robóticos con agarre de tipo mano permite automatizar tareas previamente realizadas por humanos, minimizando la necesidad de acondicionar el entorno al brazo robótico. Esto es debido a la destreza y capacidad de manipulación que otorgan este tipo de agarres. Su aplicación es evidente: desde permitir un mayor grado de automatización en procesos industriales a incrementar la capacidad motriz de personas con discapacidades mediante sistemas asistenciales. No obstante, estas aplicaciones requieren con frecuencia manipular objetos previamente desconocidos, lo que en la práctica conlleva un sobre-coste elevado para reconfigurar y reprogramar el agarre del robot al nuevo objeto. A pesar de la existencia de simuladores que nos permite de manera eficiente evaluar y verificar que la configuración es válida para el agarre, automatizar la identificación del agarre sigue siendo un aspecto desafiante de la robótica. Por otro lado, en función de la tarea a realizar es posible que el agarre óptimo pueda impedir la manipulación, es por tanto necesario que el sistema sea multisolución, es decir, sea capaz de encontrar múltiples agarres alternativos lo suficientemente viables. En este trabajo se abordará la identificación automática de múltiples agarres para un objeto previamente desconocido mediante el uso de aprendizaje automático. Para ello, se evaluará en un simulador robótico una aproximación de prueba y error basada en la optimización bayesiana multisolución, la variante multisolución de un método de optimización global de caja negra.

Palabras clave: robótica, agarre, manipulación, optimización bayesiana.

1 Introducción

La capacidad de agarrar objetos previamente desconocidos para su manipulación es uno de los principales retos a los que se enfrenta la robótica hoy en día, donde, gran parte de este reto se basa en identificar correctamente la posición en la que orientar el brazo y sus actuadores para realizar un agarre correcto y poder llevar a cabo su posterior manipulación. Debido al coste que conlleva operar

un brazo robótico, resulta inviable realizar una aproximación por fuerza bruta probando diferentes agarres. Es por ello que son necesarios métodos que traten de identificar el agarre de manera eficiente. Uno de los métodos más eficientes para encontrar una solución óptima, es la optimización bayesiana, que se usará en este trabajo.

La optimización bayesiana es un método de optimización global, eficiente, que no requiere información sobre el problema a optimizar y que funciona en un régimen secuencial de prueba y error donde evalúa diferentes configuraciones de agarres para identificar la mejor configuración. El hecho de que no necesite información previa del problema facilita que pueda aprender el agarre de cualquier objeto sin necesidad de realizar cambios en el sistema. También, que no dependa de información a priori, generalmente proporcionada por un modelo o un experto, evita que exista un sesgo inicial que impida aprender agarres de objetos completamente diferentes a los incluidos en el modelo o con los que trabaja el experto.

Existen diversos trabajos previos que se basan en la optimización bayesiana para el cálculo de agarres. Uno de los primeros trabajos usaba optimización bayesiana en combinación con *mean-shift* para explorar de manera eficiente un objeto para su agarre, utilizando una cámara externa para evaluar la calidad del agarre [7]. Si bien la mayoría de algoritmos que se basan en optimización bayesiana utilizan un proceso gaussiano como modelo estocástico, el trabajo de Montesano y Lopes emplea procesos beta para representar un agarre como un valor binario [12]. Otra de las ventajas de la optimización bayesiana, es la capacidad de utilizar el modelo estocástico para realizar otro tipo de razonamientos sobre la optimización, como en el caso de Nogueira et al. donde analizan la reproducibilidad del agarre para obtener soluciones seguras y fiables [13]. Este trabajo fue extendido posteriormente para incorporar colisiones y oclusiones [2]. Recientemente, se han realizado diversos trabajos que consideran a la vez la tarea de exploración táctil con optimización bayesiana y percepción del objeto con nubes de puntos [3]. Finalmente, en [4] se puede encontrar un ejemplo de uso de optimización bayesiana con

aprendizaje por refuerzo para tareas de manipulación.

No obstante, es necesario destacar que durante el aprendizaje del agarre, es inviable modelar todos los posibles escenarios de manipulación. En la práctica, puede ocurrir que un agarre óptimo impida realizar ciertas tareas y manipulaciones. Por ejemplo, para manipular un objeto y colocarlo sobre su base, un agarre que cubra la base puede fallar en la tarea. Es por ello que resulta imprescindible que el sistema sea capaz de ofrecer múltiples alternativas de agarre. En este trabajo se propone y se evalúa el uso de optimización bayesiana con multisolución.

Las contribuciones de este trabajo son:

- El estudio de diferentes parametrizaciones de agarre para facilitar la exploración táctil de manera eficiente utilizando optimización bayesiana.
- El estudio de diferentes técnicas de optimización con multisolución para obtener, de manera eficiente, una base de datos de agarres óptimos o estables y que permitan agarrar el objeto en el futuro aunque el mejor agarre este oculto o no sea válido para la tarea a realizar.

2 Optimización bayesiana

La optimización bayesiana (BO) es un método que trata de optimizar problemas de caja negra, costosos de evaluar y posiblemente ruidosos de manera eficiente. Para conseguirlo, utiliza 2 ingredientes fundamentales: el *modelo probabilístico subrogado* $p(f)$, que aprende las propiedades y características de la función a partir de los datos previamente evaluados; y la *función de adquisición* $a(x, p(f))$, la cual en base al modelo subrogado nos permite determinar donde evaluar la siguiente muestra de la optimización.

Formalmente, la optimización bayesiana trata de encontrar el óptimo global de una función desconocida y costosa $f : \mathcal{X} \rightarrow \mathbb{R}$ sobre el dominio $\mathcal{X} \subset \mathbb{R}^d$ realizando de manera secuencial evaluaciones sobre f . En la iteración t , todos los valores previamente evaluados $\mathbf{y} = y_{1:t}$ en los puntos $\mathbf{X} = \mathbf{x}_{1:t}$ son utilizados para aprender el modelo probabilístico subrogado $p(f|y_{1:t}, \mathbf{x}_{1:t})$. El siguiente punto a evaluar \mathbf{x}_{t+1} es obtenido maximizando en \mathcal{X} la función de adquisición:

$$\mathbf{x}_{t+1} = \arg \max_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}, p(f | y_{1:t}, \mathbf{x}_{1:t})) \quad (1)$$

Las funciones de adquisición son heurísticas diseñadas de tal forma que balanceen el *trade-off*

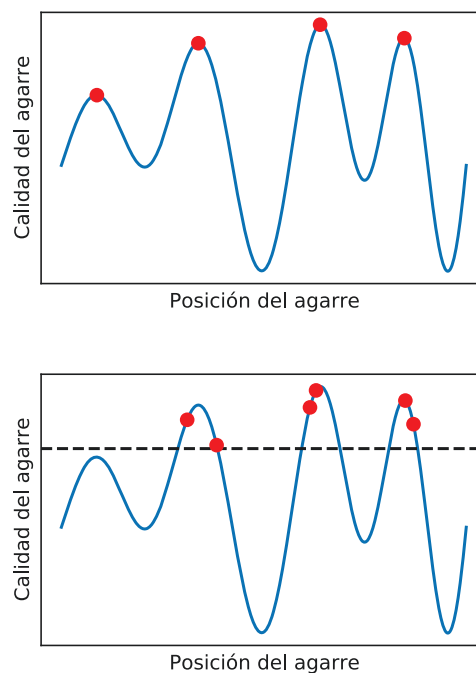


Figura 1: Comparativa de los diferentes métodos para obtener múltiples soluciones. En el primer caso (arriba) tenemos un sistema de multisolución óptima donde el objetivo es encontrar los múltiples agarres que obtienen altos valores de métrica de calidad de agarre. En el segundo caso (abajo) tenemos un sistema de multisolución mediante restricciones, el objetivo es encontrar múltiples agarres que superen la restricción de valor representada por la línea punteada.

entre *exploración* y *explotación*. La *exploración* trata de evaluar en zonas donde la incerteza del modelo subrogado es elevada, principalmente por falta de datos en la región. La *explotación* se centra en evaluar zonas donde el modelo subrogado predice valores de métrica elevados. Este equilibrio entre ambas es clave para encontrar la solución óptima de manera eficiente. Una introducción más completa a la optimización bayesiana y sus posibles usos puede encontrarse en [1].

3 Optimización bayesiana multisolución

Es posible que el agarre óptimo obtenido en un entorno aislado no sea viable para ciertas tareas y manipulaciones del objeto, debido a posibles oclusiones o limitaciones de movimiento del robot. Dado que tener en cuenta todas las posibles aplicaciones del agarre es inviable, es más eficiente que, en lugar de una solución, el sistema sea capaz de devolver múltiples soluciones, es decir, una serie

de agarres viables y suficientemente variados para que el robot disponga de alternativas en caso de fallo en la manipulación. En la figura 1 se muestra un ejemplo de multisolución idóneo.

Por defecto, la optimización bayesiana solo encuentra una única solución. Además, todo el proceso de optimización se centra en encontrar una única optimización, por lo que utilizar los N mejores evaluaciones durante la optimización no garantiza suficiente diversidad. Tampoco sirve seleccionar N evaluaciones en diferentes puntos del espacio, ya que no garantizan una métrica de agarre de calidad. Es necesario usar una variante de optimización bayesiana que tenga en cuenta la multisolución. Se han identificado 2 variantes las cuales permitirían seleccionar múltiples soluciones:

BO multisolución: Se balancea de manera activa el encontrar soluciones con valores de métrica elevados y el evaluar puntos suficientemente diversos. Esta búsqueda activa de diversidad se asemeja a la realizada en algunas variantes de BO con paralelismo, donde se introduce una repulsión local alrededor de una solución ya elegida para forzar que la siguiente solución sea distinta. Esto se puede conseguir sustituyendo el valor de una solución ya elegida por un valor constante bajo [5] o calculando una penalización alrededor de soluciones ya elegidas a partir de la *constante de Lipschitz* [6]. Esta variante requiere seleccionar la cantidad de soluciones de antemano y es importante destacar que elegir un número elevado de soluciones requerirá un mayor número de evaluaciones de agarre para encontrar la multisolución más óptima mediante BO. En la figura 1 se representa una multisolución óptima en un problema de ejemplo.

Búsqueda activa con restricciones (constrained active search): Se utilizan restricciones (*constraints*) en el valor de la métrica durante la optimización bayesiana, de tal forma que cuando la optimización bayesiana decida el siguiente punto, se centre en evaluar puntos que satisfagan la restricción de la métrica [8]. Esta aproximación requiere seleccionar un valor adecuado para la restricción: si es muy elevado apenas encontrará soluciones y si es un valor bajo encontrará demasiadas soluciones. En la figura 1 se representa algunas soluciones encontradas que satisfacen la restricción en un problema de ejemplo.

4 Agarre de Robot Simulado

Para el sistema propuesto, las simulaciones han sido realizadas mediante el software Simox [18]. Importante destacar que las contribuciones de este trabajo son agnósticas a la elección de simulador.

Con este software se ha simulado la tarea de agarre de objetos arbitrarios con el robot *iCub*, cuya herramienta (*end-effector*) es una mano robótica con 5 actuadores. El objetivo es encontrar la posición del Punto Central de la Herramienta (TCP) que maximice la métrica de calidad *radio* (Sección 4.1), mediante técnicas de optimización. Para este propósito se han implementado tres planificadores de agarre (Sección 4.2) basados en la optimización del TCP mediante coordenadas cartesianas o esféricas.

4.1 Métrica de calidad del agarre

En el problema que se plantea la función objetivo es la calidad de una determinada posición relativa mano-objeto. Existen numerosas métricas de calidad del agarre definidas en la literatura [14], basadas normalmente en los puntos de contacto, torques y fuerzas aplicadas al objeto.

En este trabajo se ha utilizado una métrica basada en la envolvente convexa del *Wrench space* del agarre, concretamente la métrica *radio* (ϵ) de la mayor bola 6D centrada en el origen, que se puede encerrar en dicha envolvente, presentada en [10]. El agarre se considera más eficiente cuanto más cerca esté ϵ de 1.

Se asume que la información relativa al tamaño y localización del objeto está disponible, y se utiliza para delimitar el espacio de búsqueda. Además, en caso de colisión con el objeto o no exista contacto con su superficie la métrica ϵ tomará el valor 0.

4.2 Planificadores de agarre

Remarcar que el agarre se lleva a cabo cuando el *end-effector* está situado en la posición final deseada. Una vez posicionado se cierran sus 5 actuadores de manera síncrona y, cada uno de ellos se detendrá cuando entre en contacto con la superficie del objeto o cuando llegue a su límite articular. Cuando el movimiento de los actuadores ha finalizado se calcula la métrica de calidad del agarre. Finalmente, destacar que este proceso es igual para todos los planificadores.

4.2.1 *GraspPlanner*

El planificador *GraspPlanner* (GP) solo hace uso del *end-effector*, dejando fuera el resto del modelo del robot *iCub*. Además, tampoco es compatible con la inclusión de objetos y elementos en el entorno de simulación más allá del propio objeto a agarrar.

El planificador acepta como entrada la posición (x, y, z) y orientación (*roll, pitch, yaw*) del TCP en coordenadas globales. Cada vez que se recibe

una nueva posición objetivo, se calcula la posición relativa del *end-effector* y éste se sitúa en la nueva localización, posteriormente se comprueba que no existan colisiones y en tal caso se procede a agarrar el objeto.

4.2.2 *GraspPlannerSpherical*

En este planificador (GPS) se sustituyen las entradas en coordenadas cartesianas por coordenadas esféricas, lo que permite definir una posición y orientación del TCP utilizando únicamente 3 parámetros (θ , ϕ , ρ). La posición del origen de coordenadas esféricas se sitúa dependiendo del tamaño y forma del objeto a agarrar, para abarcar la mayor superficie posible en la optimización. La orientación del eje X de la herramienta se establece en sentido opuesto al rayo cuya orientación está definida por θ y ϕ , con una corrección de 45 grados para que la palma de la mano se sitúe frente al objeto y evitar la colisión con los dedos. Las coordenadas esféricas resultantes se transforman en coordenadas globales para evaluar el agarre como en el caso anterior.

Con el objetivo de mejorar la optimización se han desarrollado dos tipos de parametrización más, basados en coordenadas esféricas, que se describen a continuación.

Distancia ρ a partir de la intersección entre rayo y objeto (GPS_i): La siguiente parametrización supone una geometría del objeto conocida, para eliminar un parámetro de la optimización. Se calcula el valor de ρ según la intersección entre el modelo tridimensional del objeto y el rayo con la dirección establecida por θ y ϕ , añadiendo un valor de compensación en función del tamaño de la herramienta. Esta intersección se calcula para la malla de triángulos que forma el objeto mediante el algoritmo Möller-Trumbore presentado en [11].

Optimización del origen de coordenadas esféricas (GPS_o): Uno de los inconvenientes de la parametrización anterior resulta de la influencia del valor de compensación en ρ sobre la posición del agarre óptimo. Para reducir esta influencia y favorecer la exploración se optimiza la posición del origen de coordenadas esféricas, en este caso optimizando su posición a lo largo del eje vertical del objeto.

4.2.3 *GraspPlannerIK*

Los dos planificadores anteriores sólo tienen en cuenta el *end-effector* y omiten información relevante y necesaria para su ejecución en un entorno real. Por tanto, con el objetivo de obtener resul-

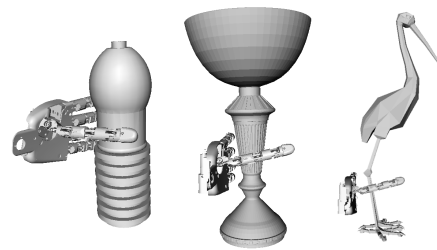


Figura 2: Objetos de prueba utilizados en simulación para la evaluación de los agarres.

tados en un entorno de simulación más realista se ha implementado el planificador *GraspPlannerIK* (GPIK).

Este nuevo planificador hace uso del modelo completo del robot y permite añadir objetos a su entorno. Con esto se pretende que cuando reciba la posición deseada del TCP, en coordenadas cartesianas como el GP, se deba planificar y ejecutar una trayectoria completa de las distintas articulaciones que conforman la cadena cinemática del robot desde la posición inicial hasta alcanzar el objetivo. En caso de que no se pueda obtener una trayectoria el agarre se considera inválido y la métrica será 0. Además, la planificación tendrá en cuenta las posibles colisiones que pueda haber entre el robot y su entorno, y entre las distintas articulaciones del propio robot.

La trayectoria se obtiene mediante el planificador *Bidirectional Rapidly-exploring Random Trees* (BiRRT), incluido en Simox. El BiRRT se basa en el muestreo del *C-space* y un problema común en este tipo de planificadores es que la trayectoria resultante no es óptima. Por tanto, se ha decidido aplicar un post-procesado sobre la solución del BiRRT con el objetivo de optimizar la trayectoria. Dicho post-procesado se basa en la búsqueda de atajos sin colisiones en el *C-space* a partir de la solución dada. Además, debido a la dificultad de encontrar una solución que permita alcanzar de manera 100% precisa la posición deseada del TCP, se ha establecido un error máximo de 2 milímetros en posición y de 0.04 radianes en orientación.

5 Experimentos

En este trabajo se han empleado dos plataformas de optimización para la obtención de resultados, por un lado la librería de optimización bayesiana BayesOpt [9] y, por otro lado, la plataforma de optimización como servicio (OaaS) SigOpt [15]. Con la librería BayesOpt se ha configurado un primer método de optimización bayesiana (BO) como el mencionado en la sección 2. Mientras que

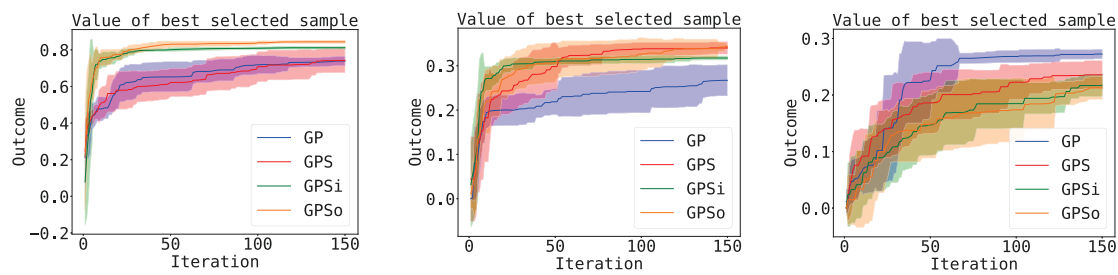


Figura 3: Evolución de la calidad de los agarres obtenidos utilizando las diferentes parametrizaciones de agarre. De izquierda a derecha: botella, jarrón, escultura.

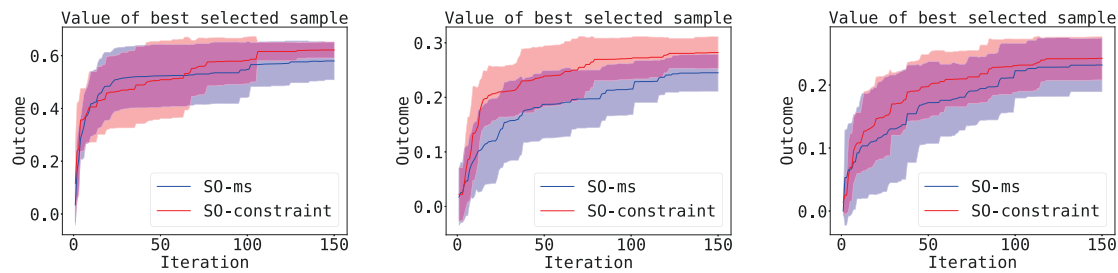


Figura 4: Evolución de la calidad de los agarres obtenidos en GPIK utilizando los diferentes algoritmos de soluciones múltiples: SigOpt multisolution (SO-ms) y SigOpt with metric constraints (SO-cnt). De izquierda a derecha: botella, jarrón, escultura.

con SigOpt se han utilizado modelos disponibles en su plataforma para optimización con múltiples soluciones, como se ha descrito en la sección 3: optimización con restricciones (SO-cnt) [16], cuyo *threshold* se ha establecido en un 80% sobre el valor máximo obtenido con BO; y optimización multisolución (SO-ms) [17], que se ha configurado para que devuelva 4 soluciones. Cada método de optimización realizará un total de 150 iteraciones o evaluaciones de agarres.

En la sección 5.1 se presenta primero una comparación entre las diferentes parametrizaciones de agarre; seguido de una comparación entre los métodos que ofrecen múltiples soluciones (SO-ms y SO-cnt) en la sección 5.2.

Se han utilizado objetos de diferente forma y tamaño para la evaluación de los agarres, concretamente una botella, un jarrón y una escultura de un animal. Estos objetos pueden verse junto con la mano izquierda del *iCub* en la Figura 2. Cada experimento se ha ejecutado un total de 10 repeticiones para cada uno de los objetos de prueba.

Tanto para el GP como el GPIK se optimiza la posición del TCP en los tres ejes de coordenadas para los dos primeros objetos, mientras que la orientación se mantendrá siempre igual con la palma de la mano apuntando y en paralelo al objeto. Por otro lado, para el objeto escultura se optimiza la posición en los ejes de altura (*Z*) y aproximación

a la superficie (*Y*), y la orientación en *roll*, manteniendo la posición en el eje *X* centrada en el objeto.

Los límites del espacio de búsqueda para los ejes *X* y *Z* se fijan en las dimensiones del objeto, anchura y altura respectivamente. Para el eje *Y* los límites se extienden desde la superficie del objeto hasta el plano en el que no exista contacto entre el *end-effector* y el objeto. Por último, los límites del espacio en *roll* van desde la orientación de la mano en paralelo a la superficie del objeto hasta la orientación completamente en perpendicular a dicha superficie. En el caso del planificador con coordenadas esféricas el valor de θ y ϕ se ha limitado entre 0 y 180 grados y los límites de ρ dependen de las dimensiones del objeto y están comprendidos entre 20 y 150. En la optimización del origen de coordenadas se utiliza su posición absoluta en el eje *Z* de coordenadas cartesianas, con los límites de optimización en función de las dimensiones del objeto.

5.1 Resultados con optimización bayesiana

La Figura 3 muestra la media y desviación típica del mejor agarre conseguido durante todo el proceso para los diferentes objetos y planificadores. En ella se puede observar como tanto en la botella como en el jarrón, los planificadores basados en coordenadas esféricas muestran un mejor rendimiento

Tabla 1: Distribución de los agarres a lo largo del objeto. Se muestra la varianza de los agarres clasificados como óptimos en las diferentes coordenadas.

		Botella		Jarrón		Escultura	
		SO-ms	SO-cnt	SO-ms	SO-cnt	SO-ms	SO-cnt
GPIK	σ_x^2	14.17	0.26	103.4	16.5	-	-
	σ_y^2	28.72	0.08	382.28	9.46	403.17	4.5
	σ_z^2	558.87	65.16	2349.56	7.34	3827.79	756.01
	σ_{metric}^2	$1.3 \cdot 10^{-3}$	$1.3 \cdot 10^{-4}$	$9.3 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$	$1.08 \cdot 10^{-3}$	$3.74 \cdot 10^{-5}$

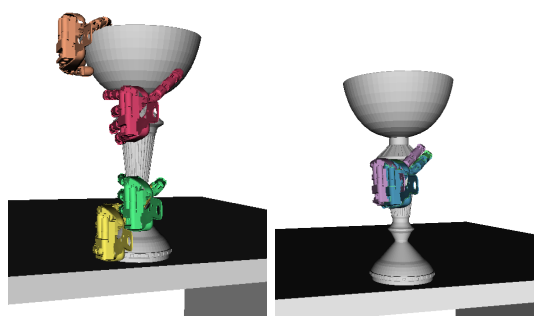


Figura 5: Comparativa de resultados multisolución entre SO-ms y SO-cnt en el problema jarrón.

y precisión. Sin embargo, en el caso de la escultura sucede al contrario. Esto puede tener relación con la forma y simetría del objeto a agarrar. En objetos más irregulares, como el caso de la escultura, la orientación definida por las coordenadas esféricas parece no ser suficiente para obtener agarres precisos. Además, la desviación típica de los resultados muestra una diferencia de estabilidad significativa que se corresponde con el comportamiento descrito anteriormente.

5.2 Resultados con optimización multisolución

En este caso se puede observar en la Figura 4 que ambos métodos de optimización multisolución obtienen un agarre óptimo muy similar, aunque el conseguido por SO-cnt es ligeramente superior. Además, se ve que la desviación típica de SO-cnt es levemente inferior durante todo el proceso y obteniendo, por tanto, resultados más estables.

En la Tabla 1 se puede ver la media de la variación de agarres devueltos como solución para SO-ms y SO-cnt. Aquí se aprecia una diferencia significativa entre ambos métodos, mientras que SO-ms ofrece una mayor variación en las soluciones en cada experimento, SO-cnt devuelve por lo general agarres que están muy cerca unos de otros. Como se puede ver en la Figura 5. Además, la variación de la métrica ϵ es muy pequeña y, por tanto, la calidad de los agarres óptimos es prácticamente la misma, tanto para SO-ms como para SO-cnt.

Con esto se ve lo explicado en la Sección 3, mientras que uno explota el encontrar soluciones diversas (SO-ms), el otro trata de encontrar soluciones que satisfagan ciertas restricciones sin importar su variedad (SO-cnt). En ambos casos intentando maximizar la métrica.

6 Conclusiones

Este trabajo muestra el uso de la optimización bayesiana para la exploración táctil de un objeto para obtener los agarres más estables. También se ha demostrado que la parametrización del agarre es fundamental para obtener una exploración eficiente. Por otro lado, se han diseñado y evaluado extensiones del algoritmo optimización bayesiana para obtener múltiples soluciones que permitan realizar agarres fiables y robustos del objeto, aunque el objeto esté parcialmente ocluido o requiera ser agarrado por una zona específica para ser manipulado o realizar la tarea. Se han evaluado dos métodos: *multisolution* que permite obtener soluciones diversas según las multimodalidad de la función de resultado y el método basado en restricciones que permite cubrir el espacio de soluciones mientras garantiza un valor mínimo en resultado. Los experimentos se han evaluado en simulación con varios objetos y una mano robótica humanoide.

Referencias

- [1] Brochu, E., Cora, V.M., and de Freitas, N. (2010) "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.
- [2] Castanheira, J. Vicente, P., Martinez-Cantin, R., Jamone, L. and Bernardino, A. "Finding safe 3D robot grasps through efficient haptic exploration with unscented Bayesian optimization and collision penalty ", IEEE/RSJ IROS, 2018.
- [3] de Farias, C., Marturi, N., Stolkin, R., Bekiroglu, Y. (2021) "Simultaneous Tactile

- Exploration and Grasp Refinement for Unknown Objects”, IEEE Robotics and Automation Letters, 2021
- [4] Englert, P. and Toussaint, M. (2016) “Combined Optimization and Reinforcement Learning for Manipulation Skills”, 10.15607/RSS.2016.XII.033.
- [5] Ginsbourger, D., Le Riche, R. and Carraro, L (2007) “A multi-points criterion for deterministic parallel global optimization based on kriging”, NCP07.
- [6] González, J. I., Dai, Z., Hennig, P. and Lawrence, N. D. (2016). Batch Bayesian Optimization via Local Penalization. AISTATS.
- [7] Kroemer, O., Detry, R., Piater, J., and Peters, J. (2009) “Active learning using mean shift optimization for robot grasping”, IEEE/RSJ IROS, 2009.
- [8] Malkomes, G., Cheng, B., Lee, E.H., and McCourt, M. (2020) “Beyond the Pareto Efficient Frontier: Constraint Active Search for Multiobjective Experimental Design”, *Neurips* 2020).
- [9] Martinez-Cantin, R. (2014) “BayesOpt: A Bayesian Optimization Library for Nonlinear Optimization, Experimental Design and Bandits”, *Journal of Machine Learning Research*, 15(Nov):3735–3739, 2014.
- [10] Miller, A.T. and Allen, P.K. (1999) “Examples of 3D grasp quality computations”, *IEEE ICRA*, 1999.
- [11] Möller, T., and Trumbore, B. (1997) “Fast, minimum storage ray-triangle intersection”, *Journal of graphics tools*, 1997.
- [12] Montesano, L. and Lopes, M. (2012) “Active learning of visual descriptors for grasping using non-parametric smoothed beta distributions”, *Rob. Auton. Syst.*, vol. 60, no. 3, 2012.
- [13] Nogueira, J., Martinez-Cantin, R., Bernardino, A. and Jamone, L. (2016) “Unscented Bayesian optimization for safe robot grasping”, *IEEE/RSJ IROS*, 2016.
- [14] Roa, M., and Suárez, R. (2015) “Grasp quality measures: review and performance”, *Auton Robots*, vol. 38, 2015.
- [15] SigOpt: A model development platform that makes it easy to track runs, visualize training, and scale hyperparameter optimization. <https://sigopt.com/>. Accedido: 30-06-2022.
- [16] SigOpt: All Constraints Experiment. https://docs.sigopt.com/experiments/all_constraints. Accedido: 30-06-2022.
- [17] SigOpt: Multisolution Experiment. https://docs.sigopt.com/advanced_experimentation/multi_solution. Accedido: 30-06-2022.
- [18] Vahrenkamp, N., Kröhnert, M., Ulbrich, S., Asfour, T., Metta, G., Dillmann, R. and Sandini, G. (2012) “*Simox: A Robotics Toolbox for Simulation, Motion and Grasp Planning.*” *Advances in Intelligent Systems and Computing*. 193.



© 2022 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC-BY-NC 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).