2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT) | 978-1-6654-1980-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/METROIND4.0IOT51437.2021.9488517

Enhancing Object Detection Performance Through Sensor Pose Definition with Bayesian Optimization

1stLoris Roveda
Istituto Dalle Molle di studi
sull'Intelligenza Artificiale (IDSIA),
6900 Lugano, Switzerland.
loris.roveda@idsia.ch

2nd Marco Maroni
Politecnico di Milano,
Department of Mechanical Engineering,
Milano, Italy

3rd Lorenzo Mazzuchelli
Politecnico di Milano,
Department of Mechanical Engineering,
Milano, Italy

4th Loris Praolini
Politecnico di Milano,
Department of Mechanical Engineering,
Milano, Italy

5th Giuseppe Bucca
Politecnico di Milano,
Department of Mechanical Engineering,
Milano, Italy

6th Dario Piga
Istituto Dalle Molle di studi
sull'Intelligenza Artificiale (IDSIA),
6900 Lugano, Switzerland.

***Abstract*—Robots equipped with vision systems at the end-effector provide a powerful combination in industrial contexts. While much attention is dedicated to machine vision algorithms, the optimization of the vision system pose is not properly addressed (to increase object detection performance). A complete pipeline for such optimization is proposed. To this aim, Bayesian Optimization is employed. A Franka EMIKA Panda robot has been used as a robotic platform, equipped at its end-effector with an Intel© RealSense D400. Achieved results show the high-fidelity reconstruction of the real working environment for the offline optimization (*i.e.*, performed simulations), together with capabilities of the proposed Bayesian Optimization-based approach to defining the sensor pose in a limited number of experimental trials (50 maximum iterations has been considered).**

***Index Terms*—Object detection, camera pose optimization, Bayesian optimization, industrial robots, 3D vision system.**

I. Introduction

Industry 4.0 paradigm has put a huge importance on robotic platforms autonomy inside production plants. The robot is required to self-adapt to its working environment to perform an application, such as assembly [1] and human-robot interaction [2] tasks. Machine vision is enhancing such autonomy [3], providing to the manipulator the capabilities to sense its working environment. In particular, object detection-based tasks (such as parts manipulation and quality inspection applications) are becoming extremely powerful, having a camera mounted at the robot end-effector to be positioned for the localization of complex parts, even in cluttered environments [4]. In such a scenario, the robot kinematics can be exploited to find the most suitable camera pose, maximizing the object detection performance and, therefore, the task success. Despite machine vision algorithms have been improved to work in difficult situations [5], there is still a lack of optimization algorithms to manage the vision system pose in such tasks.

A. Related work

Object detection for robotic applications is currently a hot-research topic, due to the huge variety of applications requiring such capabilities [6]. One critical issue in this field is to enhance the object detection performance optimizing the vision system pose (*i.e.*, to maximize the object detection performance). Such capability, in fact, is of tremendous importance in many real robotic applications. In many real working conditions, in fact, obstacles and occlusions are present, reducing drastically the chances to have a clear view of the part [7]. Therefore, being able to correctly position the vision system inside the working environment is mandatory. With this aim, some works can be identified from the literature review. [8] proposes a methodology to optimize the placement of the camera in order to minimize the detection error in the 3D measurements, treating the issue as an optimization problem. [9] describes the optimal placement of multiple cameras and the selection of the best subset of cameras for a single-object localization in the framework of sensor networks. In [10] a method for automatic sensor placement for model-based object detection is described, involving the determination of the optimal sensor placements and a shortest path through these viewpoints. The problem of camera placement for automated visual inspection tasks is studied under a multi-objective framework in [11], exploiting an evolutionary based technique. [12] proposed an approach for the computation of the next best view for object detection purposes. Firstly, the visibility of the object candidate from a set of candidate views that are reachable by a robot is analyzed. Then, the visibility of the object features by projecting the model of the most likely object into the scene is analyzed. In such a way, the next best view for the object detection purposes is performed. In [13] a framework for the definition of the vision system positioning which uses an analysis of the object stable poses, together with dynamic simulation to predict the probability distribution of initial object poses is proposed. A scalable approach to determine a small number of well-placed camera viewpoints for optical surface inspection planning is proposed in [14]. By defining a set of geometric feature functionals, an adaptive, non-uniform surface sampling (sparse in geometrically low-

978-1-6654-1980-2/21/$31.00 ©2021 IEEE

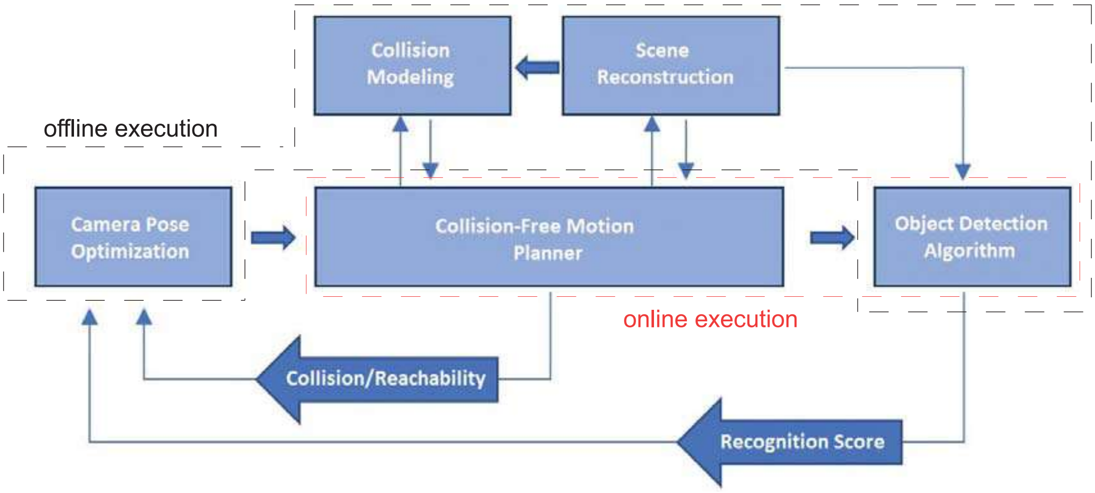

Fig. 1: The complete workflow of the proposed framework is shown, highlighting the connections between all the defined blocks. The offline part of the methodology (including the scene reconstruction, the collision modeling, the object detection algorithm, and the camera pose optimization) is highlighted, together with the online part (including the object detection algorithm and the collision-free motion planner).

complexity areas, and dense in regions of higher complexity) is performed.

The main drawback of the above described approaches is that occlusions are not considered. If the object is not (at least) partially visible, the proposed algorithms are not able to compute the vision system pose. In addition, cluttered environments are not considered, having the robot possibly colliding with other objects in the operating scene. Moreover, the vision system positioning criteria is not always related to the maximization of the object detection performance.

B. Paper contribution

Within the proposed context, the development of an approach capable of optimizing the robot end-effector mounted vision system pose is the main objective of this paper. A complete pipeline for such optimization is proposed, composed by the following main components: working scene reconstruction, robot-environment collisions modeling, object detection, sensor pose optimization, and collision-free robot motion planning. To validate the proposed approach, experimental tests have been executed in a real-like industrial environment, in which a target part has to be recognized. The target part (*i.e.*, a driller) is positioned in a toolbox with many other components occluding its visibility. A Franka EMIKA Panda robot has been employed as a robotic platform, equipped at its end-effector with an Intel© RealSense D400. Achieved results show the high-fidelity reconstruction of the working environment for the offline optimization (*i.e.*, performed simulations), making it possible to model the camera occlusions and collisions between the robotic system and the environment. The proposed approach is capable to optimize the sensor pose in a reduced amount of optimization iterations (*i.e.*, reducing the processing time). 50 iterations are considered for the optimization process.

II. Methodology

The proposed pipeline for the optimization of the (robot end-effector mounted) camera pose is shown in Figure 1. The object detection performance are maximized while guaranteeing the pose reachability w.r.t. both robot kinematics and collisions constraints within the reconstructed operating environment. Five main blocks are identified in the pipeline. A *scene reconstruction* block is defined, to acquire the point cloud for the operating environment modeling. Such a point cloud will be used to perform the object detection inside the operating scene. A *collision modeling* block is defined, to define the environmental constraints. Such environmental constraints will be used to compute both the collisions-free optimized camera pose and the collisions-free robot motion. An *object detection algorithm* block is defined, to perform the recognition of the target object inside the working scene. On the basis of the target part CAD (in .stl format) file and exploiting the point cloud available from the *scene reconstruction* block, the adopted algorithm gives as an output an index quantifying the performance of the object detection procedure. Such an index is used to optimize the camera pose inside the operating scene. A *camera pose optimization* block is defined, to optimize the camera positioning inside the working scene. Bayesian Optimization [15] is exploited for such a purpose, being able to minimize the required experiments (*i.e.*, the application time and resources usage). The object detection index from the *object detection algorithm* block is employed in the Bayesian Optimization cost function (together with penalties related to collisions and unreachable poses). A *collision-free motion planner* block is defined, to plan and execute the robot motion, positioning the camera in the optimized pose. The optimized camera pose from the *object detection algorithm* block is exploited, together with the environmental constraints from the *collision modeling* block.

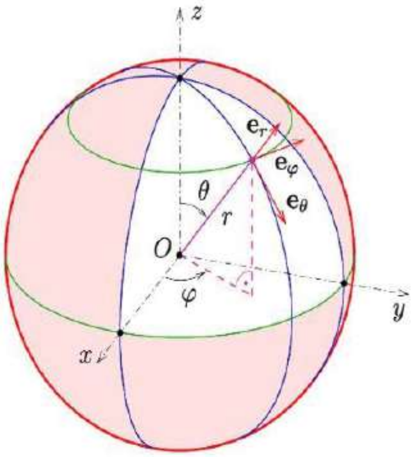

Fig. 2: Admissible camera poses lay on the reference sphere with radius r maximizing the field of view of the sensor and angles ϕ and θ defining the specific position pointing to the sphere center.

III. Camera pose optimization exploiting Bayesian Optimization

A. Camera pose definition in the operating scene

The camera pose (*i.e.*, the robot end-effector pose) affects the performance of the object detection algorithm. The camera has to point to the target object in order to be visible and perform its recognition. Commonly, the position of the target object is (roughly) known in industrial applications [7]). Therefore, it is possible to define the camera orientation to point directly to the object. In addition, on the basis of the adopted vision system, the specific sensor field of view has to be considered, so that the target part is positioned in its range of visibility (*i.e.*, not too close, not too far from the sensor). On the basis of this considerations, admissible camera poses are generated in the reconstructed operating scene (exploiting the *scene reconstruction* block) in order to lay on the surface of a sphere centered in the (expected) target object position, with a radius r (*i.e.*, the distance between the sensor and the target part) maximizing the performance of the adopted sensor. The orientation of each pose is defined to have the camera pointing to the center of the sphere (*i.e.*, pointing to the target part). In such a way, two angles ϕ and θ (polar and azimuth angles, respectively) can be used as independent variables in order to described the position on the given sphere surface (polar coordinates, Figure 2). The angles ϕ and θ are then the optimization variables to be optimized by the Bayesian optimization in order to maximize the object detection performance.

B. Cost function

In order to optimize the camera pose on the proposed nominal sphere described in Section III-A (*i.e.*, optimize the polar and azimuth angles ϕ and θ) to maximize the part detection performance (*i.e.*, the matching score), a cost function J guiding the optimization has to be defined. In this paper, the following cost function is proposed:

$$\begin{aligned} J = & K_s(S-1) + K_r(S-\min(S,S_r)) \\ & - K_k K - K_c C - K_p(\max(S,S_p) - S). \end{aligned} \tag{1}$$

The cost function (to be maximized, with values in the range $[-\infty,\ 0]$) is composed by reward terms and penalty terms. W.r.t. reward terms, the following components can be identified: $K_s(S-1)$ and $K_r(S-\min(S,S_r))^2$. S is the matching score (*i.e.*, the output of the *object detection algorithm* block) having values between 0 (no match) and 1 (perfect match), K_s is the reward gain, S_r defines a threshold for the matching score S to achieve an additional reward, and K_r is the additional reward gain. $K_s(S-1)$ is an absolute reward based on the matching score S. $K_r(S-\min(S,S_r))$ is a relative reward that is enabled if $S-\min(S,S_r) > 0$, *i.e.*, $S > S_r$, that is the matching score S is over the defined threshold S_r. W.r.t. penalty terms, the following components can be identified: $K_k K$, $K_c C$, and $K_p(\max(S,S_p) - S)$. K is the reachability index defining if the target position on the sphere is kinematically reachable having binary values 0 (target position reachable) or 1 (target position not reachable), K_k is the reachability penalty gain, C is the collisions penalty index defining if collisions are shown in the target position on the sphere having binary values 0 (no collisions are present) or 1 (collisions are present), K_c is the collisions penalty gain, S_p defines a threshold for the matching score S to produce an additional penalty, and K_p is the penalty gain associated to low matching scores. $K_k K$ is an absolute penalty related to the reachability of the target pose on the sphere that exploits the *collision-free motion planner* block. $K_c C$ is an absolute penalty related to the presence of collisions in the target pose on the sphere that exploits the *collision modeling* block. $K_p(\max(S,S_p) - S)$ is a relative penalty that is enabled if $\max(S,S_p) - S > 0$, *i.e.*, $S_p > S$, that is the matching score S is lower than the defined threshold S_p. The cost function J in (1) is therefore capable to guide the optimization in order to achieve a reachable collisions-free camera pose maximizing the matching score, *i.e.*, the object detection performance.

C. Bayesian Optimization

The cost function J (1) in Section III-B defines a metric to tune the camera pose parameters ϕ and θ. Let us collect all the design parameters in a vector $\boldsymbol{\theta}$. The tuning task thus reduces to the minimization of the cost $J(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, within a space of admissible values $\boldsymbol{\Theta}$. However, a closed-form expression of the cost J as a function of the design parameter vector $\boldsymbol{\theta}$ is not available. Furthermore, this cost cannot be evaluated through numerical simulations as the robot dynamics are assumed to be partially unknown. Instead, it is possible to perform experiments on the robot and measure the cost J_i achieved for a given controller parameter vector $\boldsymbol{\theta}_i$, and thus run an optimization algorithm driven by measurements of J. This peculiar nature of the optimization problem at hand restricts the class of applicable optimization algorithms. Indeed, (*i*) the measured cost J_i consists in a noisy observation of the ”true” cost function, namely $J_i = J(\boldsymbol{\theta}_i) + n_i$, with n_i denoting measurement noise and possibly intrinsic process variability;

(*ii*) no derivative information is available;

(*iii*) there is no guarantee that the function $J(\boldsymbol{\theta})$ is convex;

(*iv*) function evaluations may require possibly costly and time-consuming experiments on the robot.

Features (*i*), (*ii*) and (*iii*) rule out classical gradient-based algorithms and restrict us to the class of gradient-free, global optimization algorithms. Within this class of algorithms, *Bayesian optimization* (BO) is generally the most efficient in terms of the number of function evaluations [16] and it is thus the most promising approach to deal with (*iv*).

In BO, the cost J is simultaneously learned and optimized by sequentially performing experiments on the robot. Specifically, at each iteration i of the algorithm, an experiment is performed for a given controller parameter $\boldsymbol{\theta}_i$ and the corresponding cost J_i is measured. Then, all the past parameter-cost observations $\mathcal{D}_i = \{(\boldsymbol{\theta}_1, J_1), (\boldsymbol{\theta}_2, J_2), \dots, (\boldsymbol{\theta}_i, J_i)\}$ are processed and a new parameter $\boldsymbol{\theta}_{i+1}$ to be tested at the next experiment is computed according to the approach discussed in the following. Additional details related to the *surrogate model*, *acquisition function*, and *algorithm outline* used for Bayesian Optimization can be found in [17].

IV. Results

In order to prove the effectiveness of the proposed framework in real conditions, an experimental test has been executed to consider a realistic environment (*i.e.*, high level of occlusions, high number of obstacles, presence of random objects in the operating scene).

A. Operating scenario

The target part to be detected is a driller, as shown in Figure 3. The proposed real-like environment validation scenario is shown in Figure 4. The part is inside a toolbox, with many other random parts positioned around. Upper toolbox drawers are open, so that a high-level of occlusion is achieved (*i.e.*, only a small portion of the nominal sphere defining the admissible camera poses allows to properly detect the target part). Therefore, the optimization of the camera pose is highly important to execute the target object detection task.

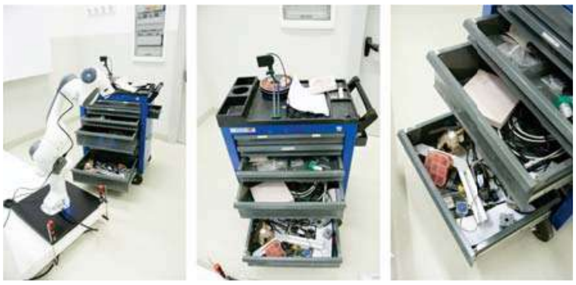

Fig. 4: Real environment validation scenario.

B. Results

The achieved results are shown in the video available at https://www.youtube.com/watch?v=B9KXQ2wixrY&t=12s. In the presented video, the complete procedure described in Section II is shown, together with the task execution.

The proposed framework allows to model the operating scene, reconstructing the collision objects, exploiting such information in order to (offline) optimize the camera pose for the object detection task and to (online) execute the collision-free robot motion to that goal position.

Figure 5 shows the robot executing the object detection in the real-like operating environment, exploiting the optimized camera pose (based on the BO approach), being capable to recognize the part.

To show the increased performance achieved by the object detection algorithm exploiting the proposed camera pose optimization methodology, the a-type uncertainty on the estimated object pose (*i.e.*, on both translational and rotational degrees of freedom - DOFs) has been computed, comparing the obtained results with non-optimized camera pose-based object detection. Table I shows the achieve results, highlighting that the optimized camera pose allows to improve the performance related to the estimation of the target object pose (extremely important in industrial applications, such as inspection or grasping tasks).

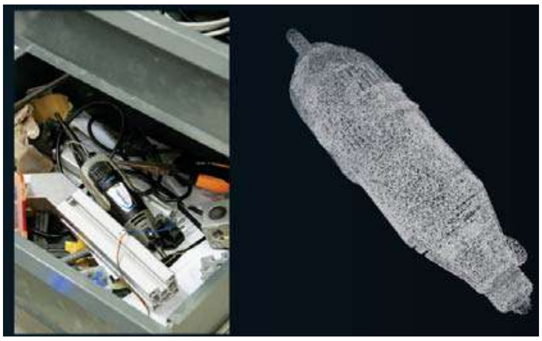

Fig. 3: Real part (driller) to be recognized in the proposed real-like environment.

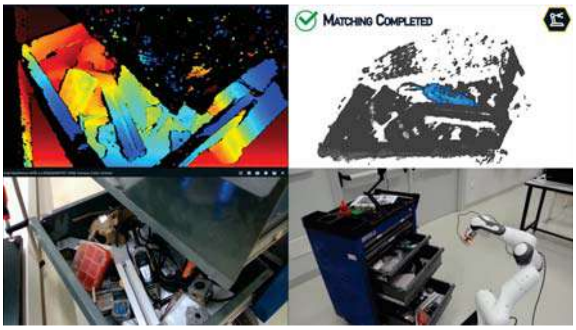

Fig. 5: Object detection task in the real-like operating environment. In the top-right side of the Figure, the correct detection of the part is highlighted.

Camera Pose	Translational Uncertainties	Rotational Uncertainties
	[mm]	[°]
BO-based	0.86	0.088
Not Optimized	6.18	38.02

TABLE I: A-type uncertainty for the estimated target object pose exploiting the BO-based camera pose and a not optimized camera pose. The a-type uncertainty is computed exploiting 20 values for the estimated object pose (for both translations and rotations) for both the two camera poses.

Remark 1. The developed open source software in ROS, including all the components described in Section II, is available at the GitHub repository https://github.com/Loris-Praolini/best_view_algorithm.

V. Conclusions

In this paper, a complete pipeline for the optimization of the (robot end-effector mounted) camera pose for object detection purposes is proposed. The provided approach relies on the following main components: working scene reconstruction, robot-environment collisions modeling, object detection, sensor pose optimization (exploiting Bayesian Optimization), and collision-free robot motion planning. The proposed components have been integrated in ROS to define a flexible application structure, capable to work with real manipulators in industrial tasks. A Franka EMIKA Panda robot has been employed as a robotic platform for the experimental validation, equipped at its end-effector with an Intel$^{\copyright}$ RealSense D400. A real-like use-case has been presented to demonstrate the applicability of the proposed pipeline.

Future work is devoted to include transfer learning capabilities into the developed pipeline to maximize the knowledge from already available data.

Acknowledgment

The work has been developed within the project ASSASSINN, funded from H2020 CleanSky 2 under grant agreement n. 886977.

References

[1] L. Roveda, M. Magni, M. Cantoni, D. Piga, and G. Bucca, "Human-robot collaboration in sensorless assembly task learning enhanced by uncertainties adaptation via bayesian optimization," *Robotics and Autonomous Systems*, p. 103711, 2020.

[2] L. Roveda, J. Maskani, P. Franceschi, A. Abdi, F. Braghin, L. M. Tosatti, and N. Pedrocchi, "Model-based reinforcement learning variable impedance control for human-robot collaboration," *Journal of Intelligent & Robotic Systems*, pp. 1–17, 2020.

[3] L. Roveda, N. Castaman, S. Ghidoni, P. Franceschi, N. Boscolo, E. Pagello, and N. Pedrocchi, "Human-robot cooperative interaction control for the installation of heavy and bulky components," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018, pp. 339–344.

[4] C. Militaru, A.-D. Mezei, and L. Tamas, "Object handling in cluttered indoor environment with a mobile manipulator," in *2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*. IEEE, 2016, pp. 1–6.

[5] Y. Sumi, Y. Kawai, T. Yoshimi, and F. Tomita, "3d object recognition in cluttered environments by segment-based stereo vision," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 5–23, 2002.

[6] J. Chen, L. Zhang, Y. Liu, and C. Xu, "Survey on 6d pose estimation of rigid object," in *2020 39th Chinese Control Conference (CCC)*. IEEE, 2020, pp. 7440–7445.

[7] F. Vicentini, N. Pedrocchi, M. Beschi, M. Giussani, N. Iannacci, P. Magnoni, S. Pellegrinelli, L. Roveda, E. Villagrossi, M. Askarpour *et al.*, "Piros: Cooperative, safe and reconfigurable robotic companion for cnc pallets load/unload stations," in *Bringing Innovative Robotic Technologies from Research Labs to Industrial End-users*. Springer, 2020, pp. 57–96.

[8] G. Olague and R. Mohr, "Optimal camera placement for accurate reconstruction," *Pattern recognition*, vol. 35, no. 4, pp. 927–944, 2002.

[9] A. O. Ercan, D. B. Yang, A. El Gamal, and L. J. Guibas, "Optimal placement and selection of camera network nodes for target localization," in *International Conference on Distributed Computing in Sensor Systems*. Springer, 2006, pp. 389–404.

[10] S. Chen and Y. Li, "Automatic sensor placement for model-based robot vision," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 393–408, 2004.

[11] E. Dunn and G. Olague, "Pareto optimal camera placement for automated visual inspection," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3821–3826.

[12] C. McGreavy, L. Kunze, and N. Hawes, "Next best view planning for object recognition in mobile robotics." CEUR Workshop Proceedings, 2017.

[13] T. M. Iversen and D. Kraft, "Optimizing sensor placement: A mixture model framework using stable poses and sparsely precomputed pose uncertainty predictions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6652–6659.

[14] D. Mosbach, P. Gospodnetić, M. Rauhut, B. Hamann, and H. Hagen, "Feature-driven viewpoint placement for model-based surface inspection," *Machine Vision and Applications*, vol. 32, no. 1, pp. 1–21, 2020.

[15] M. Pelikan, D. E. Goldberg, E. Cantú-Paz *et al.*, "Boa: The bayesian optimization algorithm," in *Proceedings of the genetic and evolutionary computation conference GECCO-99*, vol. 1. Citeseer, 1999, pp. 525–532.

[16] 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," *arXiv preprint arXiv:1012.2599*, 2010.

[17] L. Roveda, M. Forgione, and D. Piga, "Robot control parameters auto-tuning in trajectory tracking applications," *Control Engineering Practice*, vol. 101, p. 104488, 2020.