Development of a Bi-level TO-Based Co-design Pipeline for the RePAIR Robotic Platform

Andrea Patrizi, Arturo Laurenzi, Francesco Ruscelli,
Eamon Barrett and Nikolaos Tsagarakis

October 6, 2022

Development of a Bi-level TO-based Co-design Pipeline for the RePAIR Robotic Platform

Andrea Patrizi   Arturo Laurenzi   Francesco Ruscelli   Eamon Barrett   Nikos G. Tsagarakis

***Abstract*—We present a bi-level co-design pipeline employed for the optimization of a set of relevant relevant kinematic parameters of the bi-manual robotic platform under development for the european project RePAIR. In particular, the bottom level consists of a monolithic co-design framework, based on trajectory optimization, which jointly optimizes for design and state variables. The framework employs a kinematic model of the system and accounts for collisions, state bounds, a number of user-defined tasks, while minimizing a suitable performance index. The top level employs a three-step heuristic globalization algorithm which performs several calls to the low level TO to cope with the observed sensitivity of the TO solution w.r.t. the choice of the initial guess.**

***Index Terms*—co-design, trajectory optimization, clustering, manipulability, RePAIR**

I. Introduction

This work applies concurrent-design principles to the final design refinement stage of the bi-manual robotic platform which is under development for the RePAIR European project [1]. This project lies at the intersection of robotics and artificial intelligence and its goal is the development of innovative technologies to automate the physical reconstruction of fragmentary archaeological artefacts. The robotic platform is made of two 7-DOF manipulators (shown in Figure 1) and has an additional translational degree of freedom along the frontal axis (blue arrow in Figure 1).

The objective of this work is to determine the optimal value w.r.t. a suitable performance criteria of three relevant kinematic mounting parameters (shown in Figure 1), whereas the kinematic design of each arm of the platform is fixed beforehand and hence is not subject to optimization.

We choose to tackle the problem of determining the best mounting parameters by employing trajectory optimization (TO). This choice allows us to solve for the design variables while ensuring kinematic feasibility, imposing a variety of constraints (for example collision and task constraints) and, more importantly, minimizing a performance index. Given the use case of the project, we select our performance criteria to avoid large joint excursions and velocities during tasks execution. We will see in Section II-B3 that these specifications can be easily translated into a suitable cost function for the TO and that this cost can be given the interesting interpretation of a non-local manipulability index (as opposed to the local [2]). To make code prototyping faster, we make use of the TO framework Horizon [3].

The authors are with the Humanoid and Human-Centred Mechatronics (HHCM), Istituto Italiano di Tecnologia (IIT), Genova. Please send correspondence to andrea.patrizi@iit.it

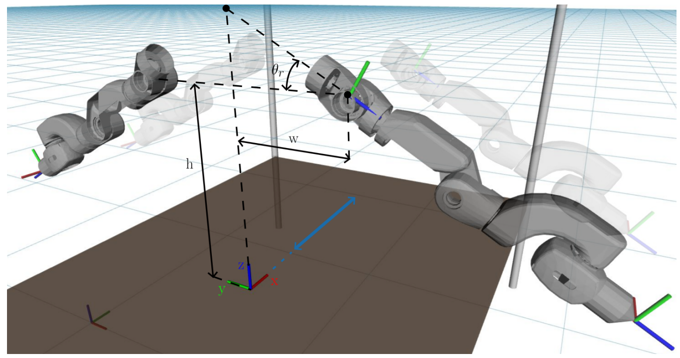

Fig. 1. Design variables, for simplicity shown only for the right arm. h is the mounting height measured from the working plane to the shoulder axis, along the z direction; w is half of the shoulder width (symmetrical between left and right), measured along the y axis; θ_r is the shoulder mounting angle, measured w.r.t. the x direction. The light blue arrow shows the translational degree of freedom along x axis.

The main contribution of this work is a bi-level hybrid gradient/sample-based co-design pipeline. In particular, the bottom level provides raw optimized data to the top level, which was developed to cope with the observed sensitivity of the TO solution w.r.t. to the employed initial guess. The co-design pipeline is made of the following main components (summarized in Figure 2):

- A low level monolithic TO for computing optimal design (and state) variables, while accounting for joint limits, collisions and project-specific tasks.
- A genetic-inspired heuristic globalization pipeline built on top of the TO level to navigate the set of local minima for the underlying TO problem.

II. Formulation

A. Use case and requirements

The employed formulation was developed accounting for the most relevant characteristics associated with the use case of the RePAIR platform. In particular, the following observations can be made:

- The project involves manipulation of objects in the proximity of the working plane; consequently, tasks must be chosen in accordance with this characteristic.
- The project requires operation in the relatively narrow space of the working cage. Hence, large joint excursions should be avoided as much as possible to minimize the risk of colliding with the working surface and the cage.

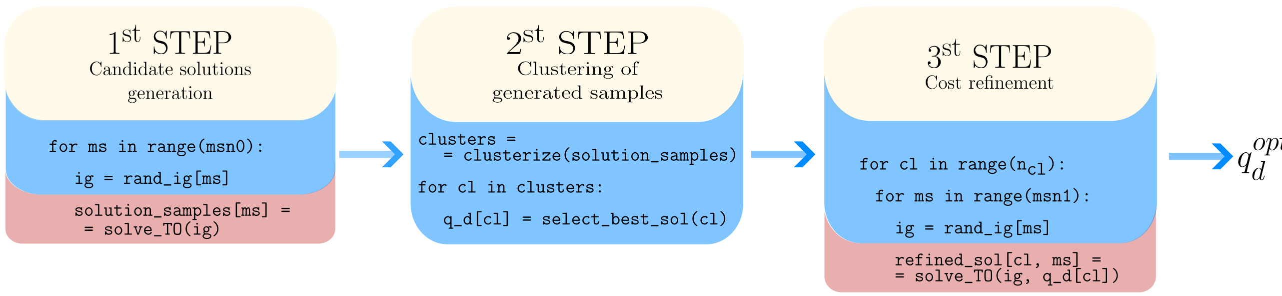

Fig. 2. Co-design pipeline description. The pipeline is made of a lower level which houses a TO (pseudo-code in red areas) and a high level heuristic globalization algorithm (pseudo-code in light blue areas). First, several candidate solutions are retrieved by solving many instances of the low level TO for different initial guesses. We call this procedure 'multi-starting' the TO. Then, a clustering step (via Mini-Batch K-Means algorithm [4]) is applied to group 'close' solutions and, for each cluster, only the best candidate (w.r.t. the optimal cost) is selected. The final phase consists of a cost refinement step: the value of the true cost of each candidate is updated to the best value between the one coming from the first step and the ones obtained by multi-starting once again the TO for each cluster candidate (design paramters are fixed to the value associated to each cluster candidate). The final design parameters are extracted from the candidate with the lowest refined cost.

- The platform will have to grasp and reassemble archaeological fragments; this will involve frequent and low-speed single arm, bi-manual and reorientation maneuvers.
- At the current stage of the project, the chosen end-effector is the SoftHand [5], a tendon-driven anthropomorphic robotic hand designed for grasping and soft manipulation (shown in Figure 3). It is therefore advisable to account for the capabilities of this end-effector in the choice of task constraints.

B. Low level monolithic TO

1) Simplifications: Instead of employing the full kinematics of the anthropomorphic hand, we simplify the problem by considering a point end-effector (visible in Figure 1); the real hand is only employed for visualization purposes. This simplification allows for a more end-effector agnostic and less expensive analysis; the capabilities of the hand are then incorporated by specifying relevant and sensible tasks.

Furthermore, given the presence of the sliding degree of freedom along the x-axis, it is sufficient to specify task constraints on any plane parallel to the yz one.

2) Tasks: Given the scope of the project and the expected capabilities of the hand, we devise a series of relevant tasks which are then translated into pose constraints for the low-level TO. Tasks are replicated along the y axis to explore the full workspace of the platform.

We test the co-design pipeline on a couple of particularly representative cases which, from here on, will be referenced to as the 'handover' or 'flipping' task and the 'bi-manual pick' task. Specifically, the handover task is made of the following main phases:

1) The first arm approaches a chosen pick position on the working surface and picks the object from above.
2) The object is passed to the other arm using a maneuver we refer to as 'handover'. This maneuver was designed to comply with the expected manipulation capabilities of [5].
3) The second arm performs a flipping maneuver and proceeds to place the object back at the original pick position.

Differently, the bi-manual pick is structured in the following way:

1) Both arms approach a chosen pick position and lift the object keeping a bi-manual configuration which is chosen to comply with the manipulation capabilities of [5].
2) The object is lifted up to a reference height while keeping the bi-manual configuration.

Examples of the aforementioned tasks performed by the platform are shown in the accompanying video.

3) TO formulation: The problem is formulated as the following continuous optimization program:

$$\begin{aligned} \min_{q,p} \quad & J(q,p) = \int_0^T \|\dot{q}\|^2 \, dt \\ \text{s.t.} \quad & \dot{q} = u \\ & \text{task constraints} \\ & \text{state bounds} \\ & \text{collision constraints} \end{aligned} \tag{1}$$

where q is the state vector of the system and p is the vector of design parameters. We neglect any dynamic effects (given the expected slow motions of the use case) and employ a kinematic model ($\dot{q} = u$) instead of the full-fledged rigid-body dynamics. This simplification makes the optimization lighter and avoids unnecessary complexity. Moreover, we employ the simple running cost $\|\dot{q}\|^2$ on the basis of the following considerations:

- Manipulation of fragile archaeological artifacts requires low joint velocities.
- Minimizing joint velocities corresponds to avoiding large joint excursions, which is one of the requirements stated in Section II-A.
- From a purely mathematical point of view, such a cost works as an input-regularization term.

The continuous TO (1) is then discretized using a uniform time grid of N nodes. Constraints (tasks, collisions and state bounds) and costs are distributed over the defined nodes with the criteria explained in Figure 4. Collision constraints are enforced using the simplified collision model shown in Figure 3.

Interestingly, the cost J can be given a suggestive interpretation. Let us define the following performance index:

$$\eta_p = \frac{1}{\sqrt{\frac{\sum_{i=0}^{N-1} \dot{q}_i^2}{N}}} \tag{2}$$

where $\dot{q}_i$ is the value of $\dot{q}$ assumed at node i. This index is inversely proportional to the square root of the average running cost of the discretized TO. The higher (2), the lower are the average joint velocities needed to perform the tasks. On this line of thought, (2) can be interpreted as a non-local or distributed manipulability measure (as opposed to classical manipulability measures [2], which are intrinsically local) indicating how easily, in terms of joint velocities (or, equivalently, joint excursions between TO nodes) an end-effector trajectory can be performed by the platform.

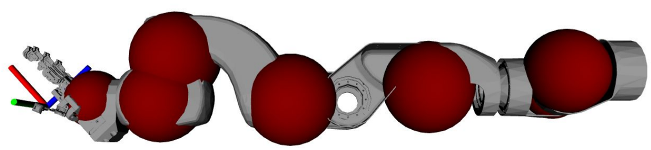

Fig. 3. Simplified collision model employed for the collision constraint: each link is modeled with a sphere.

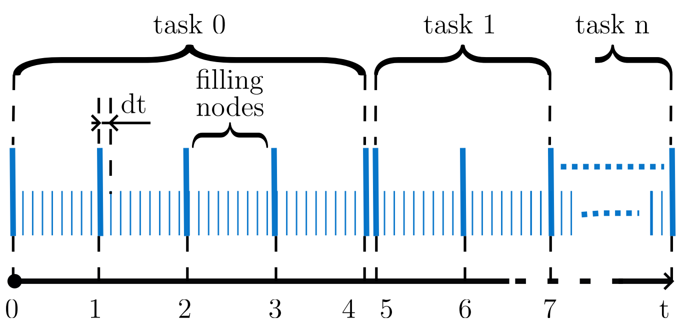

Fig. 4. Problem formulation: the TO is discretized using a uniform time grid of N nodes, shown by the vertical blue lines in the picture. A black horizontal arrow indicates the time axis. Multiple tasks (of the same or different kind) are distributed among the nodes of the problem. Tasks are enforced using suitable pose constraints at specific nodes, which we refer to as 'task base nodes' (indicated in the picture by the integer numbers on the time axis). Aside from task base nodes, a number of filling nodes (thin vertical blue lines) are added to the problem to be able to solve for collision-free trajectories. Furthermore, the cost is relaxed in between the final and initial base node of two successive tasks, so that the transition between them and their ordering in time do not influence the optimization.

C. Top level heuristic globalization algorithm

We observed a high sensitivity of the optimal solution of (1) w.r.t. the employed initial guess; as a consequence, inspired by genetic algorithms, we devised a globalization pipeline (briefly described in Figure 2) built on top of the low level TO.

The pipeline is made of the following main phases:

1) Samples generation step: The first step consists of finding several candidate solutions by solving many instances of the low level TO for different initial guesses; these guesses are generated randomly, starting from a given seed. From now on, we will refer to this kind of approach as 'multi-starting' the low level TO. The resulting locally optimal design parameters, costs and trajectories are then stored and passed to the following step.

2) Clustering step: We observe that solutions resulting from the first step span almost the whole design space, but they are not evenly distributed across it. Instead, they tend to form agglomerates. For this reason, we employ a clustering algorithm, specifically Mini-Batch K-Means [4], to group 'close' solutions. We then select the best candidate for each cluster and feed these promising solutions to the last step of the pipeline.

3) Cost refinement step/true fitness evaluation: The last step of the pipeline is similar to the first and consists of multi-starting the low level TO for each cluster candidate, with the design variables fixed to their optimal value. This allows to explore several locally optimal joint trajectories and better characterize each candidate solution. The fitness of each cluster candidate (i.e. their optimal cost) is updated to the lowest value between the original cost and the new ones coming from the multi-start procedure. We refer to this value as the 'refined' cost of the cluster candidate.

4) Final solution selection: The final optimal solution is chosen among the set of cluster candidates by picking the one attaining the lowest refined cost.

III. Conclusions and Open Challenges

In this work we presented a bi-level co-design pipeline developed to address the final design refinement stage of the RePAIR project [1]. In particular, we provided a concise description of the main components of both pipeline levels and outlined the driving reasons behind their development.

Future work will include a more in depth analysis of the top level globalization algorithm and improvements in the selection of the final solution by combining more complex and robust selection criteria (e.g. computing a confidence index indicating how much a cluster candidate refined cost can be trusted). Other possible improvements include, but are not limited to, the use of a more accurate collision models, the inclusion of the full end-effector kinematics in the low level TO and the testing of additional representative tasks. Moreover, the formal relationship between the proposed non-local manipulability index (2) and the classical [2], if any, deserves investigation. Finally, it would also be interesting to test and adapt the proposed pipeline to different and more complex problems with larger design space and, possibly, discrete variables.

References

[1] "RePAIR." https://www.repairproject.eu/, 2022. [Online; accessed 18-August-2022].

[2] T. Yoshikawa, “Manipulability and redundancy control of robotic mechanisms,” in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1004–1009, 1985.

[3] F. Ruscelli, A. Laurenzi, N. G. Tsagarakis, and E. M. Hoffman, “Horizon: a trajectory optimization framework for robotic systems,” *Frontiers in Robotics and AI*, vol. 9, 2022.

[4] “MiniBatchKMeans.” https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html, 2022. [Online; accessed 5-September-2022].

[5] C. D. Santina, C. Piazza, G. Grioli, M. G. Catalano, and A. Bicchi, “Toward dexterous manipulation with augmented adaptive synergies: The pisa/iit softhand 2,” *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1141–1156, 2018.