Benchmark for Bimanual Robotic Manipulation of Semi-Deformable Objects

Konstantinos Chatzilygeroudis, Bernardo Fichera, Ilaria Lauzana, Fanjun Bu,
Kunpeng Yao, Farshad Khadivar, and Aude Billard

***Abstract*—We propose a new benchmarking protocol to evaluate algorithms for bimanual robotic manipulation semi-deformable objects. The benchmark is inspired from two real-world applications: (a) watchmaking craftsmanship, and (b) belt assembly in automobile engines. We provide two setups that try to highlight the following challenges: (a) manipulating objects via a tool, (b) placing irregularly shaped objects in the correct groove, (c) handling semi-deformable objects, and (d) bimanual coordination. We provide CAD drawings of the task pieces that can be easily 3D printed to ensure ease of reproduction, and detailed description of tasks and protocol for successful reproduction, as well as meaningful metrics for comparison. We propose four categories of submission in an attempt to make the benchmark accessible to a wide range of related fields spanning from adaptive control, motion planning to learning the tasks through trial-and-error learning.**

***Index Terms*—Performance evaluation and benchmarking, dual arm manipulation, model learning for control, dexterous manipulation.**

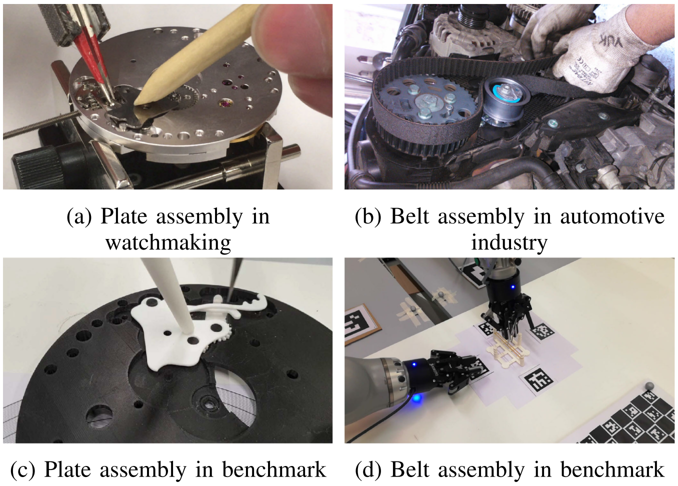
(a) Plate assembly in watchmaking
(b) Belt assembly in automotive industry
(c) Plate assembly in benchmark
(d) Belt assembly in benchmark

Fig. 1. Benchmark tasks: inserting the watch plate (left) and assembling a rubber belt (right), in an real world scenario (top) and in the benchmark setup (bottom).

I. Introduction

A VARIETY of industrial tasks are still performed by humans today, as they require high-level precision and dexterity not yet available in robots. These tasks require the use of prehensile instruments, such as screwdrivers or tweezers, to grasp, insert, and manipulate tiny and deformable objects. Examples of such tasks are common in watchmaking craftsmanship, where both assembling and screwing are the core actions in the whole process, and in pharmaceutical industry, to handle pipettes and vials. There is interest to automatize parts of these tasks [1]. Such precise manipulation can also be found in automotive industry, such as when assembling a fuse box, a task considered for robotization recently, as in [2]. These manipulations often require coordinated motion of two or more end-effectors, where at least one holds the piece in place while the others proceed to execute. The tasks are made particularly difficult as the objects being manipulated may have complex shapes, and accurate models of the piece may not be available. Moreover, these very small objects are often fragile and can easily deform or break. Balanced and precise control of force and position in coordination is hence crucial. In highly precise tasks such as those, perception may suffer from low signal/noise ratio. To achieve high level of dexterity in these tasks often requires extensive human training.

For robots to start tackling these tasks, they should be equipped with similar level of dexterity. We propose a benchmark to assess level of coordinated control of motion in force and position across two robotic arms, and ability to cope with sensori-motor noise. Furthermore, the benchmark assesses performance of learning algorithms to empower robots with such skills, in an attempt to address the need for a generalized framework for research in robotics [3], [4].

Taking this into consideration, we propose a benchmark for bimanual manipulation and propose two tasks: 1) a realistic replication of a watchmaking insertion step, (see Fig. 1(a)); 2) a replication of manipulation of rubber bands around gears (see Fig. 1(b)). For the robotic implementation of the benchmarks first task, we offer to use a watch mechanism and its winding

Manuscript received August 15, 2019; accepted December 14, 2019. Date of publication February 10, 2020; date of current version February 20, 2020. This letter was recommended for publication by Associate Editor A. Dollar and Editor H. Ding upon evaluation of the reviewers' comments. This work was supported by the European Research Council under the European Unions Horizon 2020 research and innovation programme (SAHR Project, ERC Advanced under Grant 741945). F. Bu participated in the EPFL Excellence in Engineering Summer Internship. *(Corresponding author: Konstantinos Chatzilygeroudis.)*

Konstantinos Chatzilygeroudis, Bernardo Fichera, Ilaria Lauzana, Kunpeng Yao, Farshad Khadivar, and Aude Billard are with the Learning Algorithms and Systems Laboratory, Ecole Polytechnique Fédérale de Lausanne, 1015, Lausanne, Switzerland (e-mail: konstanti.ch; bernardo.fichera@epfl.ch; ilaria.lauzana@alumni.epfl.ch; kunpeng.yao@epfl.ch; farshad.khadivar@epfl.ch; aude.billard@epfl.ch).

Fanjun Bu is with the Learning Algorithms and Systems Laboratory, Ecole Polytechnique Fédérale de Lausanne, 1015, Lausanne, Switzerland, and also with the Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: fbu2@jhu.edu).

This letter has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/LRA.2020.2972837

components in two scaled-up versions. For the sake of easy accessibility to the benchmark setup, we provide a CAD model of the objects considered for simulation, which is reproducible with 3D printing technologies. As object of choice for the first task, we consider an eccentric shaped object with a semi-deformable part, namely the *watch plate*: a metal plate with an irregular shape and an elongated, bendable leg that needs to be clipped in place as a second step of the assembly. In the second task, we use a rubber band which undergoes elastic deformations during placement around the support's pegs.

The complexity of the tasks lies in the need to determine the correct orientation prior to insertion (in the first task), the right sequence of actions for bimanual control, and the right amount of force for final insertion/or during manipulation (for the second task). While these may be determined by hand after careful tuning, uncertainties still exist due to poor visual tracking (especially due to occlusion when holding the piece or the rubber band in the robot's end-effector) and poor control of force. Such partial knowledge of the environment and imprecise perception may be overcome with machine learning algorithms, which provides a framework that uses previously acquired experience and possibly takes advantage of noisy expert demonstrations. Hence, this makes the task an interesting example where learning can be applied to determine these parameters automatically. Therefore, the benchmark also serves as a test-bed to compare different machine learning approaches to achieve multi-arm manipulation of semi-deformable objects.

II. Related Work

Previous large-scale testing of assembly and bimanual coordination algorithms were organized as challenges, such as the ARM-S project of DARPA described in [5], where teams had to perform grasping and manipulation tasks, or the RoboCup@Work [6], a competition on advanced manipulations in industrial applications. Nonetheless, the number of available benchmarks that the community is using is limited [4]. Another example is given by Fan *et al.* in [7], which proposes a framework for manipulation (bimanual peg-in-hole, nut-and-peg assembly, etc.) and for Reinforcement Learning (RL) algorithms. However, the benchmark is only in simulation, and the objects considered are all rigid bodies.

Here, we propose a real-world benchmark that attempts to replicate main challenges faced in two real-world applications.

A. Manipulation Under Uncertainty

Manipulation and assembly of objects has been widely studied, more notably in the cases of Peg-in-Hole, which may already be tackled in setups considering rigid bodies [4], [8]. Typical approaches to this problem handle uncertainties by either exploiting compliance [9]–[11] or using force feedback [12], [13]. Although the Peg-in-Hole scenario has a wide range of solutions, there is no available framework to test methods involving complex-shaped pegs so far. In fact, despite the propositions in [9], [14], [15] which consider atypical pegs, the materials and shapes of the objects analyzed are widely different. A comparison is thus impossible, as the difference in clearance and hardness of the surfaces change the nature of the assembly task (see [16]).

B. Bimanual Coordination

Multi-arm coordination is of importance in our case, since the proposed benchmark requires cooperation between multiple arms for the stability of the insertion and assembly task. It is represented by an asymmetrical task, where each limb has different motor outputs coordinated in order to achieve the goal. Such asymmetrical coordination has been studied less frequently in the past than the coordination of limbs performing similar motions (e.g., walking or typing), as discussed in [17]. Nonetheless, research on the topic in [18] has shown that in humans the non-dominant hand uses position control, and the dominant hand uses force control; this is mirrored in the task we propose, where one arm stabilizes the watch face while the dominant arm performing the actual assembly requiring force precision.

C. Deformable Objects Manipulation

Manipulation becomes more challenging when the objects are deformable, mainly due to the infinite degrees of freedom in objects' shape, which result in huge uncertainties during manipulation. It is the case for part of the presented task, which is classified as a deformable linear object manipulation problem in a recent survey by Sanchez *et al.* [19] that divides approaches by type of object manipulated and by goal (deformation sensing, grasping or object-specific goals).

Tasks involving manipulation of rubber bands have been already considered by the community, as in [20] for optimal motion planning using a precise deformation model. And while Jimenez *et al.* [21] survey more model-based algorithms for deformable objects manipulation planning, uncertainties can also be handled differently. For example, [22] targets manipulation without prior knowledge of the deformable objects, and [23] does not require the simulation or modeling of the deformation. Furthermore, Zhu *et al.* [24] propose a framework to reduce the uncertainty in the manipulated deformable object by taking advantage of environment contacts in the task of routing a cable.

Finally, although machine learning approaches can be used to tackle this issue, this project still deserves further exploration. Nonetheless, an interesting take on the matter has been introduced in [25], [26], where a policy for cloths manipulation is learned via RL, with the latter also discussing the transfer from simulation to the real world.

D. Machine Learning Approaches

As previously mentioned, issues related to uncertainties and missing knowledge of the environment may be solved with the use of machine learning algorithms. Zhu *et al.* survey in [27] the approaches to learn the assembly task using Learning from Demonstration, from either kinesthetic, motion-sensor or teleoperated demonstrations. Other works have instead taken advantage of RL, as discussed above for cloth manipulation in [25], [26], for peg-in-hole assembly in [28], [29], and for other complex manipulations in [30], [31].

III. The Benchmark

As stated in the introduction, the benchmark consists of two (2) independent tasks: (a) the *watchmaking* task, and (b) the *rubber-band* task. The benchmark accepts submissions for approaches that fall in any of the following four categories:

- Adaptive Control (AC): in this case, the optimal plan (in kinematic space) to solve the task is assumed to be given and the goal is to find a control algorithm that is able to find the force profiles and adapt to the noisy observations of the computer vision part. No data-driven method is allowed in this category.
- Motion Planning (MP) Under Uncertainty: in this case, the goal is to find a working plan (not given) to solve the task with the ability of handling the uncertainty that comes from the computer vision estimation. No data-driven method is allowed in this category.
- Offline Learning (OFL): in this case, the optimal plan (in kinematic space) to solve the task is assumed to be given or not (this is up to the users to decide; different leaderboards are available per case) and the goal is to find a data-driven algorithm that is able to reliably solve the tasks. In this category, all the learning must be offline, that is, performed only once with the collected real-world samples. Combination of learning methods with motion planning algorithms is allowed.
- Online Learning (ONL): in this case, the goal is to propose a trial-and-error learning algorithm that is able to find a working controller (or policy) that can reliably solve the tasks despite the noisy observations. The algorithm is expected to interact with the system and improve over time. The final optimized controller/policy is evaluated.

Rules and constraints per category are described in the individual tasks. The benchmark requires at minimum (a) 4 degrees of freedom robotic arms and (b) availability of a set of cameras (different number for each setup) to track the main object parts. Next, we briefly describe the set-up and baseline for each task. Detailed information is available in the benchmark website[1] and the attached protocol descriptions.

A. Watchmaking Task

This task consists in completing one step of the assembly of a typical watch, namely inserting the plate, using a dual-arm robot system holding the required tools at their end-effector. This task poses the following main challenges to robotic systems:

- Manipulating objects via a tool;
- Placing an irregularly shaped object (the plate) in the correct groove that requires millimeter accuracy control;
- Handling a semi-deformable object: the plate's leg;
- Multi-arm coordination: one arm stabilizing the system, and the other(s) performing the leg's bending and final plate insertion.

The plate is semi-deformable, and it necessitates bending in order to put it in place. Additionally, its small size requires the usage of a second tool to keep the object stable. The tools that are used in the actual setup are: *the tweezers, a screw driver and a wooden tool*. The task should be performed by a pair of robotic arms as follows:

One arm holds/supports the tweezers at its end-effector, while the other arm holds/supports the stick (or similar tool) at its end-effector. At onset, the plate is already held stiffly in the tweezers (i.e. the robot does not have to pick it up).[2] The arms must insert the plate in the correct location/groove on the watch face. This requires orienting the piece correctly and then bending its leg to stay in place.

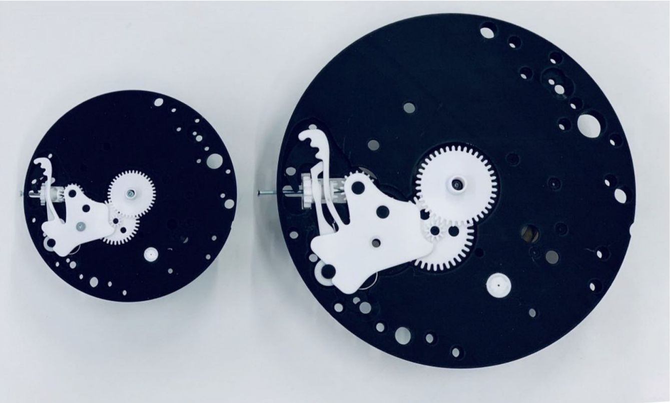

Fig. 2. Scaled-up watches 3D printed from CAD drawings: $3.5\times$ and $5.8\times$ bigger.

In order to evaluate the robustness of the proposed methods, the task needs to be performed for three (3) different watch face orientations and replicated five (5) times per orientation for each watch scale.

Watch Pieces: The real watch from which this task is inspired is quite small with a diameter of 37 mm. While, in the long run, many commercial solutions will be offered to robotic arms precise enough to manipulate such small object, it is not realistic to expect many labs to have such highly precise robots today. Hence, to avoid any additional challenge to an already difficult problem, we propose to use *scaled-up versions of the parts*. To this end, we have devised two different scaled-up versions: (a) around $3.5\times$, and (b) $5.8\times$ bigger than the actual one (see Fig. 2). These ratios are computed such that standard components like screws are available for the scaled-up watch. Furthermore, to increase reproducibility, we provide the CAD files of the parts in the original and the two scaled-up versions. We expect the potential users of the benchmark to 3D print them using technologies similar to FDM for watchface (black material in Fig. 2) and SLA for other components (white material in Fig. 2). Choosing 3D printing technologies depends on users preference and potential; however, the only constraint is that the technology must have precision up to 1 mm for watchface and 0.5 mm for other components.

Tools: The usage of the tweezers is essential for humans when they perform the task with the actual watch (that is, the original small size), as fingers are too big to hold the object, and skin contact with the watch has to be avoided to prevent contamination. However, given that we use scaled-up versions for the robotic implementation, it may be sufficient to use a standard robotic gripper to replace the tweezer. We let users decide if they want to opt for that solution or they wish to have a tweezer mounted on the robot's end-effector. Whichever solution is chosen, we set only as constraint that the tool at the end-effector have the following characteristics: (a) it only has two legs, and (b) it only has one translational degree of freedom; hence it can only "pinch". Regarding the second tool, users are allowed to use any "stick-like" tool suited to the size of their robotic set-up. The "stick" should not be actuated (i.e. no degree of freedom) and hence only extend the tip of the robot, in a similar fashion to the wooden tool held in hand.

[1] https://www.epfl.ch/labs/lasa/sahr/benchmark

[2] Picking it up could be a nice extension of the benchmark.

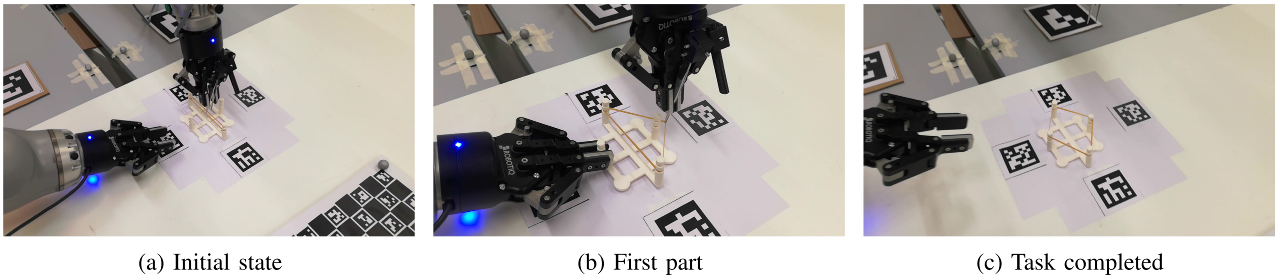
(a) Initial state (b) First part (c) Task completed

Fig. 3. Elastic rubber-band task: steps to accomplish the task.

Setup Details: In this benchmark, we want to focus on the manipulation part of the task. Thus, to remove potential irrelevant differences, we provide a computer vision (CV) setup to estimate the location of the robots and the watch parts that is based on one RGB camera and ChArUco markers,[3] and we expect all participants to use it (see Sec. IV). Also, since the predictions are noisy, using this computer vision setup also allows us to assess uncertainty handling algorithms.

Available Information: For all four categories of submissions, the users are expected use our CV setup to get the estimated poses of the watch parts. In the OFL and ONL cases, users can choose to disregard this setup and use up to one RGB camera feedback (that can either be fixed or mounted on the end-effector of one of the arms) provided that they do not use any hand-crafted computer vision algorithm to track the objects. In other words they are only allowed to use it as a raw sensor observation as input to the algorithm; they can also use both the provided CV setup and the raw image data if they wish.

Users can also use (if needed) the kinematic and dynamic models of the robotic manipulators as well as the rigid body version of the watch parts. Additionally, the information that the leg of the plate is semi-deformable can be exploited, although the an actual deformation model should not be available (it can be learned or estimated though).

B. Rubber-Band Task

This task consists in manipulating an elastic rubber-band to create a specific shape on a board with sticks (see Fig. 3). Manipulating the rubber-band requires *a control strategy that takes into account the forces generated*, which adds one more challenge to the benchmark. The task should be performed by a pair of robotic arms as follows:

One arm holds/supports the tweezers (or similar tool) at its end-effector, while the other arm uses a gripper to stabilize the system. The rubber band is already placed on the board in the initial shape. The arms must manipulate the rubber-band in order to achieve a specific shape on the board.

In order to evaluate the robustness of the proposed methods, the task needs to be performed for three (3) different board orientations and replicated five (5) times per orientation.

Required Parts: For completing the task, one needs the specific board model and a rubber band. Details about the specifications of the rubber band can be found in the benchmark website. Again, to increase reproducibility and ease the 3D printing process, we provide the CAD files of the board along with physical properties.

Tools: The usage of the tweezers might make the handling of the elastic rubber band easier. However, it may be sufficient to use a standard robotic gripper to replace the tweezer. As in the previous task, we let users decide which option they prefer, and we still impose the same "pinch" constraint on the end-effector tool. These constraints apply for both arms.

Setup Details: Similarly to the previous task, we provide a CV setup to estimate the location of the robots, the board and the location and shape of the rubber-band that is based on two RGB camera and ChArUco markers, and we expect all participants to use it (see Sec. IV).

Available Information: The available information that is given/permitted as input to the algorithms follow the same rules as in the previous task.

IV. Tracking the Objects and Robots With Computer Vision

As described above, we provide a computer vision setup and methods to construct coordinate systems for each robot arm and the watch base using ChArUco markers. ChArUco markers are the combination of ArUco markers and chessboard pattern. They can be easily detected by cameras and provide better estimation of position than purely using ArUco markers. We use the python version of OpenCV [32] to implement the tracking algorithm.

In our setup, ChArUCo markers are placed at each corner of each robot arm base. Thus, a collection of four markers are used for each robot arm. Another collection of four markers are placed on the watch base to track its location, and a ChArUco chessboard is placed on the table for easier combination of reference frames. Each marker has to be perfectly flat for accurate detection of position. We require one RGB camera placed in a manner that all markers are visible within its point of view (see Fig. 4). For the second task, we require an additional RGB camera that observes the workspace from above and tracks the position/shape of the rubber band.

Assuming that the four markers at the base of each robot arm form a perfect square, we construct the coordinate system for each robot arm as follows: the origin is located at the center of the four markers. The x-axis points towards a specific marker (different for each robot), and the z-axis points up towards the ceiling. A similar method is applied to construct the coordinate system of the watch base or the board. We also use a ChArUco chessboard to make the calculations easier.

Having created the coordinated frames for each robot and the watch face (or the board) with respect to the camera's frame, we are able to compute transformations between any two frames and thus get an estimated position of the watch face with respect to the right robot, for example. The precision accuracy of the

[3] https://docs.opencv.org/4.1.0/d9/d6a/group__aruco.html

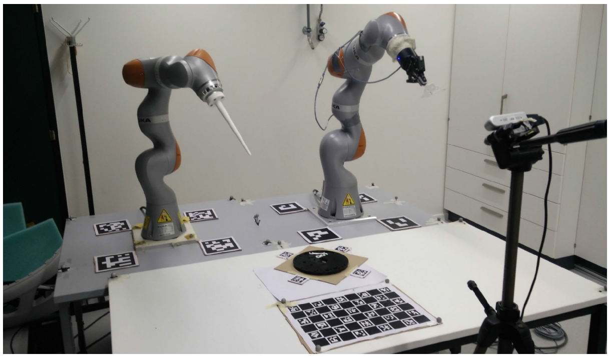

Fig. 4. Experimental setup with ChArUco markers.

system in our experiments is around 2–4 cm, which is small enough to be corrected or used in algorithms, but big enough to not allow the tasks to be performed as is, as they require millimetric precision to be performed.

V. Evaluation Protocol

In this section, we describe the evaluation protocol that quantifies the performance of the system in completing the tasks. We expect potential users of the benchmark to report the performance metrics of their approach per sub-task: (a) watchmaking task with big watch, (b) watchmaking task with small watch, (c) rubber band task. Additionally, the users are expected to report their robotic setup, but also the pipeline of their approach (e.g., whether they used some simulator or not). We define two (2) global metrics that apply to all the submission categories, but also per category metrics:

- **Global:**
  - Success rate metric (**G1**);
  - Normalized average completion time (**G2**).
- **OFL Metrics:**
  - Normalized number of offline real-world samples needed to create the datasets and/or models (**A1**);
  - Normalized model learning time (**A2**).
- **MP Metric:**
  - Normalized average time needed for the planner to output the plan (**M1**).
- **ONL Metrics:**
  - Normalized interaction time needed to train the solution controller (**T1**);
  - All OFL metrics (if used).

The total score of the benchmark (for comparison reasons), $S_{\text{benchmark}}$, is a weighted sum over all sub-tasks and scores. We also define a few offline sub-metrics that should help in identifying what went wrong in cases of failure. We set $N = 5$ the number of repetitions and $J = 3$ the number of different initial orientations, and we give the metrics as follows.

A. Global Metrics

The global metrics aim at evaluating how fast and robust the solutions of the proposed methods are.

G1 - Success rate metric:

$$S_{\text{G1}} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{J} b_{ij}}{NJ} \tag{1}$$

G2 - Normalized average completion time:

$$S_{\text{G2}} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{J}(t_{\max} - t_{ij})}{NJt_{\max}} \tag{2}$$

where b_{ij} is a binary value determining if the i-th repetition for j-th initial orientation was successful, t_{ij} is the time needed to complete the task in the i-th repetition for j-th initial pose, and $t_{\max} = 60\,\text{s}$ is the maximum time allowed to perform the task (i.e., if $t_{ij} > 60\,\text{s}$, then $t_{ij} = 60\,\text{s}$).

B. Offline Learning Metrics

A1 - Normalized number of offline real-world samples:

$$S_{\text{A1}} = \frac{k_{\max} - k}{k_{\max}} \tag{3}$$

A2 - Normalized model training time:

$$S_{\text{A2}} = \frac{t^{\text{train}}_{\max} - t^{\text{train}}_{ij}}{t^{\text{train}}_{\max}} \tag{4}$$

where k, t^{train} are the number of samples needed to create the datasets and the time to train the models respectively, $k_{\max} = 10000$ is the maximum number of offline samples, and $t^{\text{train}}_{\max} = 1800\,\text{s}$ is the maximum time for model training.

C. Motion Planning Metric

M1 - Normalized average planning time:

$$S_{\text{M1}} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{J}(t^{\text{plan}}_{\max} - t^{\text{plan}}_{ij})}{t^{\text{plan}}_{\max}} \tag{5}$$

where t^{plan}_{ij} the time the planner took to output a result in the i-th repetition of the j-th initial orientation, and $t^{\text{plan}}_{\max} = 1800\,\text{s}$ is the maximum time available for a planner to output a result.

D. Online Learning Metric

T1 - Normalized interaction time:

$$S_{\text{T1}} = \frac{t^{\text{te}}_{\max} - t^{\text{te}}}{t^{\text{te}}_{\max}} \tag{6}$$

where t^{te} is the interaction time needed to learn a controller for the task [33], and $t^{\text{te}}_{\max} = 1800\,\text{s}$ is the maximum allowed interaction time to solve a task.

E. Total Score

The total score is computed as a weighted sum over all sub-tasks and scores:

$$S_{\text{benchmark}} = \sum_{m} w_0(S^m_{\text{G1}} + S^m_{\text{G2}}) + w_1(S^m_{\text{A1}} + S^m_{\text{A2}} + S^m_{\text{M1}} + S^m_{\text{T1}}) \tag{7}$$

where $m = \{\text{watch}_{\text{big}}, \text{watch}_{\text{small}}, \text{rubber}\}$, $w_0 = 0.25$ and w_1 adapts depending on the submission category.

F. Sub-Metrics

These additional metrics make it easier to identify which parts were problematic in cases of failure:

- **6D pose tracking accuracy:** with this metric, we assess the ability of the robotic system and the controller to achieve 6D poses with millimeter accuracy. We provide a protocol to use the proposed controller to achieve six different end-effector poses, and we measure the distance from the desired pose. The score is defined as follows (the smaller the better):

$$S_{\text{track}} = \frac{\sum_{i=1}^{6} ||\boldsymbol{p}_{\text{desired}}^{i} - \boldsymbol{p}_{\text{actual}}^{i}||}{6} \tag{8}$$

  where $\boldsymbol{p}_{\text{desired}}$ is the desired end-effector pose, and $\boldsymbol{p}_{\text{actual}}$ is the actual end-effector pose achieved. We take the difference both in translation and orientation. For the orientation, we use the angle-axis representation.
- **Safe dual-arm end-effector tracking:** the robots are given end-effector targets that would lead to a collision using a simple PD-controller, and we measure the following quantities: (a) time in collision, (b) target goal (last point in the path) accuracy. The score is defined as follows (the smaller the better):

$$S_{\text{safe}} = \frac{\sum_{i=1}^{2} t_{\text{col}}^{i} ||\boldsymbol{p}_{\text{desired}}^{i} - \boldsymbol{p}_{\text{actual}}^{i}||}{2} \tag{9}$$

  where t_{col} is the time in seconds that the robots collided (i iterates over the robots).
- **Force application:** with this metric, we measure the ability of the controller to apply contact forces in specific directions while maintaining good tracking of the desired end-effector pose. This metric requires a force-torque sensor to be mounted on the end-effector of one of the robots. One of the robots is given two desired end-effector poses and desired forces to apply. We let the system run for T seconds, and we measure the average error in force and final end-effector pose error (the smaller the better):

$$S_{\text{force}} = \frac{\sum_{i=1}^{2} \boldsymbol{F}_{\text{avg\_error}}^{i} ||\boldsymbol{p}_{\text{desired}}^{i} - \boldsymbol{p}_{\text{actual}}^{i}||}{2} \tag{10}$$

These metrics are not compulsory, but we strongly encourage potential participants to use them, as they provide useful insights on what is working well or not.

VI. Baselines

We devise three (3) baseline methods: one for the AC and two for the OFL category of the benchmark. Before moving on to each baseline method, we first describe the low-level controller of our dual-arm robot system.

Dual-Arm Centralized Control: In order to safely control both manipulators, we consider both of them as a single robot and perform Quadratic Programming (QP) based inverse kinematics control in order to find the joint commands to achieve desired end-effector poses (see [34] for more details).

Apart from the end-effector target poses and to ensure that the end-effectors paths generated by the QP solver are collision free, we devise a set of sub-goals for each body part of each robot. These sub-goals produce a velocity opposite in direction to any link of the other robot, and thus creating a velocity field that repulses the robots away from each other. To make a compromise between accuracy and safety, the magnitude of these velocities are defined with an exponential function as $v = e^{-0.5\frac{\text{dist}}{\text{th}^2}}$, where th is the minimum allowed distance between the robot links, and dist is the distance between the COM of the links. Using this control scheme, we are able to control the robots to millimeter accuracy while maintaining safety and getting collision-free paths.

A. Adaptive Control Baseline

For this baseline, we make the assumption that the tracking system provides us with perfect measurements and thus a set of waypoints for the end-effector of each arm is enough such that the task can be accomplished. The main intuition behind this baseline is, that if we are given the physical properties of the objects (at least as rigid bodies) and the tracking of the objects is good enough, then we should be able to directly solve the tasks in position control and simple planning.[4] Of course, in reality (a) we do not have perfect tracking, and (b) we do not know the exact physical properties of the objects as some of them are semi-deformable.

We use our centralized QP control scheme and a set of predefined waypoints (along with timings) that are defined for each end-effector with respect to the local coordinate frame of the watch face or the board (we do this in order to generalize to new poses of the watch face/board). In essence, each robot, j, needs to achieve a set of end-effector goals G_i^j where $i = 1 \ldots N_j$ and N_j is the number of goals of robot j. We additionally give a time duration, t_i^j, for each sub-goal after which the robot moves on the next desired waypoint.

B. *Offline Learning Baseline #1*

In this baseline, we assume that the plan to solve the task is given. The main idea behind our machine learning baseline is to assume that we have access to an accurate tracking system (e.g., motion capture system) only during training, but not at the time of evaluation, and try to correct the transformations provided by the computer vision algorithm.

To do so, we put motion capture markers on the robots, the watch face/board and the chessboard. We are now able to connect the predictions of the computer vision pipeline with the ones from the motion capture by using the ChArUco chessboard as the connection. Once the motion capture and the computer vision estimations can be combined, we collect samples of the following form: $(\tilde{\boldsymbol{T}}_{\tilde{b}}^{\tilde{X}}, \boldsymbol{T}_{\tilde{X}}^{X})$, where $\tilde{\boldsymbol{T}}_{\tilde{b}}^{\tilde{X}}$ is the pose of the X with respect to the ChArUco chessboard frame given by the computer vision system, and $\boldsymbol{T}_{\tilde{X}}^{X}$ is the transformation needed to go from the computer vision estimation of the pose of X to the motion capture estimation of the same pose. X takes the following values: (a) right robot, (b) left robot, and (c) watch face or board.

Once we collect a few of these samples, we use Gaussian processes (GP) to learn the mappings[5] $\mathcal{T} : \tilde{\boldsymbol{T}}_{\tilde{b}}^{\tilde{X}} \rightarrow \boldsymbol{T}_{\tilde{X}}^{X}$ (we refer the reader to [35] for more details about GP regression).[6]

Using the learned GP models during evaluation, we only need computer vision without the help of a motion capture system. Thus, each time we query for the pose of the watch face or

[4] This baseline is not really an adaptive control one, but it fits under the requirements of the category.

[5] We model an isometry transformation by a 3D translation and an axis-angle representation for the orientation.

[6] We used the limbo [36] library for the implementation.

TABLE I
BASELINE RESULTS FOR THE WATCHMAKING TASK. THE VALUES ARE AVERAGED OVER THE TWO DIFFERENT SCALES FOR BREVITY

Score	AC Mocap	AC	OFL#1	OFL#2
G1	0.8	0	0.6	0.4
G2	0.73	0	0.66	0.64
A1	-	-	0.98	0.88
A2	-	-	0.95	0.92

the board, we apply the correction given by the GP model to the computer vision estimated pose. Since the optimal plan is assumed to be available, we use our centralized QP control scheme to follow it.

C. Offline Learning Baseline #2

In this baseline, we assume that the plan to solve the task is not given and that human teleoperated demonstrations of the full task are provided. During the demonstration, end-effector positions and orientations of both robotics arms are recorded.

Once we get the demonstration data, we seek to find the set of waypoints that the robot has to follow in order to complete the task. For this reason, we assume that the desired behavior can be model as multiple-attractor dynamical system. In order to cluster the different sub-dynamics and extract the relative attractors (or waypoints), we take advantage of a graph-based kernel Principal Component Analysis algorithm. Having identified the waypoints the robots need to follow, we use our centralized QP control scheme to do so and the correction models to correct the computer vision estimations.

VII. BASELINE RESULTS

We provide baseline results only for the first task of the benchmark (i.e., watchmaking). We also provide an example solution of the second task using teleoperation for clarity (refer to the supplementary video or the benchmark website).

A. Adaptive Control Baseline

To showcase the importance of our benchmark, we devise two sets of evaluation tests for the AC baseline. In the first set, we assume the access to a motion capture system that gives accurate poses of the objects and the robots positions up to less than 1 mm. In the second set, we use exactly the same algorithm but with the computer vision system.

The results of these experiments are depicted in Table I (AC Mocap and AC columns). We can see that when having perfect knowledge, the AC baseline performs very well; it fails only about 20% of the time, that is around 6 times over the 30 replications (due to the small inaccuracies that accumulate from the motion capture system and the controller), and takes around 18 s for each execution. On the contrary, because of the noisy signal coming from computer vision, this baseline fails every time in the actual setup of the benchmark.

B. Offline Learning Baseline #1

We used 200 offline samples to learn the correcting transformations in our OFL#1 baseline. Overall the OFL#1 baseline improves performance significantly when compared to the AC one: it fails 40% of time, and takes around 20 s to complete the task. Moreover since it only uses 200 offline samples it is fast to compute (around 100 s) and achieves high **A1** and **A2** scores.

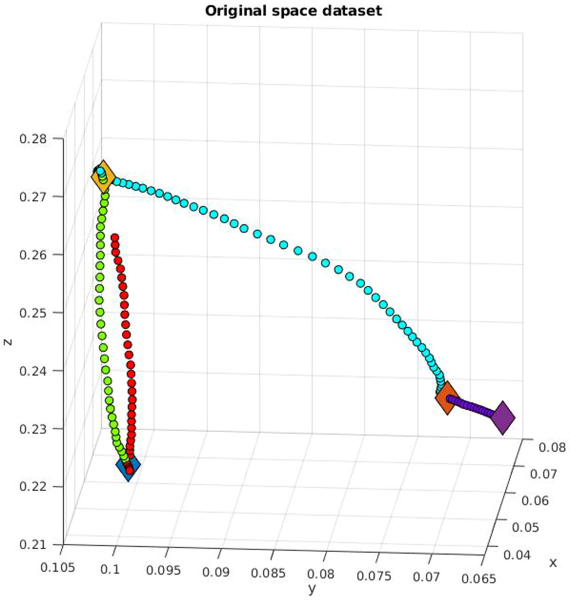

Fig. 5. Example of extracted waypoints (shown as rhombs) from a demonstrated trajectory.

The baseline used here follows the spirit of Simulation-To-Real approaches [37]. Simulation-To-Real approaches use a simulator, where ground truth is easily accessible, to learn a robust policy. This requires less knowledge of the real-world at evaluation time. Similarly, our approach uses a "lab setup" where a motion capture is available but relies only on the noisy computer vision estimate at the evaluation step.

C. Offline Learning Baseline #2

Using just one (1) demonstration, our OFL#2 baseline was able to extract the waypoints for controlling the robots (one example is shown in Fig. 5). The results are worse than the OFL#1 baseline, but much better than the AC one. In particular, the OFL#2 baseline succeeds around 40% of the time and has equivalent scores with the OFL#1 baseline in the rest of the scores (a small decrease in the samples complexity since the demonstrated trajectory consisted of 1000 samples). The reason for the decreased success rate is the fact that the waypoints we get from the data-driven approach can easily deviate a few millimeters from the optimal ones, and this precision is crucial for completing the task.

VIII. CONCLUSION

Our benchmark aims to support future advances on dexterous manipulation and bimanual coordination with semi-deformable objects. To recall, we provide four submission categories (or versions) in an attempt to make the benchmark accessible to a wide range of related fields spanning from adaptive control to trial-and-error learning.

Our baseline experiments showcase that uncertainty in the tracking of the environment, partial knowledge about the physical and dynamical properties of the objects, and precise force control are the main challenges to be tackled by the methods that will participate in this benchmark. The experiments also revealed that our benchmark tasks are practical, but also require algorithmic or theoretical advances in order to be fully solved. Our setup, metrics and sub-metrics make the benchmark accessible to a wide range of research areas, and we hope that it will generate impactful research papers.

References

[1] X. Chu, H. Fleischer, N. Stoll, M. Klos, and K. Thurow, "Application of dual-arm robot in biomedical analysis: Sample preparation and transport," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, 2015, pp. 500–504.

[2] S. Makris *et al.*, "Dual arm robot in cooperation with humans for flexible assembly," *CIRP Ann.*, vol. 66, no. 1, pp. 13–16, 2017.

[3] A. P. Del Pobil, "Why do we need benchmarks in robotics research," in *Proc. IROS Workshop Benchmarks Robot. Res.*, vol. 7, 2006, p. 105.

[4] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 36–52, Sep. 2015.

[5] D. Hackett, J. Pippine, A. Watson, C. Sullivan, and G. Pratt, "An overview of the DARPA autonomous robotic manipulation (ARM) program," *J. Robot. Soc. Japan*, vol. 31, no. 4, pp. 326–329, 2013.

[6] "RoboCup@Work," 2012-2018. [Online]. Available: www.robocupatwork.org

[7] L. Fan *et al.*, "SURREAL: Open-source reinforcement learning framework and robot manipulation benchmark," in *Proc. Conf. Robot Learn.*, 2018, vol. 87, pp. 767–782.

[8] K. Van Wyk, M. Culleton, J. Falco, and K. Kelly, "Comparative peg-in-hole testing of a force-based manipulation controlled robotic hand," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 542–549, Apr. 2018.

[9] S. Kang, Y. Hwang, M. Kim, C. Lee, and Kyo-Il Lee, "A compliant motion control for insertion of complex shaped objects using contact," in *Proc. Int. Conf. Robot. Autom.*, 1997, vol. 1, pp. 841–846.

[10] S.-k. Yun, "Compliant manipulation for peg-in-hole: Is passive compliance a key to learn contact motion?" in *Proc. Int. Conf. Robot. Autom.*, 2008, pp. 1647–1652.

[11] M. P. Polverini, A. M. Zanchettin, S. Castello, and P. Rocco, "Sensorless and constraint based peg-in-hole task execution with a dual-arm robot," in *Proc. Int. Conf. Robot. Autom.*, 2016, pp. 415–420.

[12] T. Tang and M. Tomizuka, "A learning-based framework for robot peg-hole-insertion generic framework for multi-agent tracking and prediction view project real time optimization for motion planning view project," in *Proc. ASME Dyn. Syst. Control Conf.*, 2015. [Online]. Available: https://asmedigitalcollection.asme.org/Citation/Download?resourceId=230546&resourceType=10&citationFormat=2

[13] F. Suarez-Ruiz and Q.-C. Pham, "A framework for fine robotic assembly," in *Proc. Int. Conf. Robot. Autom.*, 2016, pp. 421–426.

[14] H.-C. Song, Y.-L. Kim, and J.-B. Song, "Automated guidance of peg-in-hole assembly tasks for complex-shaped parts," in *Proc. Int. Conf. Intell. Robots Syst.*, 2014, pp. 4517–4522.

[15] H. Park, J.-H. Bae, J.-H. Park, M.-H. Baeg, and J. Park, "Intuitive peg-in-hole assembly strategy with a compliant manipulator," in *Proc. Int. Conf. Intell. Safety for Robot.*, 2013, pp. 1–5.

[16] Z. Jakovljevic, P. B. Petrovic, and J. Hodolic, "Contact states recognition in robotic part mating based on support vector machines," *Int. J. Adv. Manuf. Technol.*, vol. 59, no. 1–4, pp. 377–395, 2012.

[17] S. S. Obhi, "Bimanual coordination: An unbalanced field of research," *Motor Control*, vol. 8, no. 2, pp. 111–120, 2004.

[18] L. Ferrand and S. Jaric, "Force coordination in static bimanual manipulation: effect of handedness," *Motor Control*, vol. 10, no. 4, pp. 359–370, 2006.

[19] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey," *Int. J. Robot. Res.*, vol. 37, no. 7, pp. 688–716, 2018.

[20] E. Yoshida, K. Ayusawa, I. G. Ramirez-Alpizar, K. Harada, C. Duriez, and A. Kheddar, "Simulation-based optimal motion planning for deformable object," in *Proc. IEEE Int. Workshop Adv. Robot. its Social Impacts*, 2015, pp. 1–6.

[21] P. Jiménez, "Survey on model-based manipulation planning of deformable objects," *Robot. Comput.-Integrat. Manuf.*, vol. 28, no. 2, pp. 154–163, 2012.

[22] A. M. Howard and G. A. Bekey, "Intelligent learning for deformable object manipulation," *Auton. Robots*, vol. 9, no. 1, pp. 51–58, 2000.

[23] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4525–4532.

[24] J. Zhu, B. Navarro, R. Passama, P. Fraisse, A. Crosnier, and A. Cherubini, "Robotic manipulation planning for shaping deformable linear objects with environmental contacts," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 16–23, Jan. 2020.

[25] B. Balaguer and S. Carpin, "Combining imitation and reinforcement learning to fold deformable planar objects," in *Proc. Int. Conf. Intell. Robots Syst.*, 2011, pp. 1405–1412.

[26] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Proc. Conf. Robot Learn.*, 2018, pp. 734–743.

[27] Z. Zhu, H. Hu, Z. Zhu, and H. Hu, "Robot learning from demonstration in robotic assembly: A survey," *Robotics*, vol. 7, no. 2, 2018, Art. no. 17.

[28] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana, "Deep reinforcement learning for high precision assembly tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 819–825.

[29] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, and A. M. Agogino, "Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 2062–2069.

[30] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.

[31] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," in *Proc. Int. Conf. Robot. Autom.*, 2015, pp. 156–163.

[32] G. Bradski, "The OpenCV Library," *Dr. Dobb's J. Softw. Tools*, 2000. [Online]. Available: https://github.com/opencv/opencv/wiki/CiteOpenCV

[33] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, and J.-B. Mouret, "A survey on policy search algorithms for learning robot controllers in a handful of trials," *IEEE Trans. Robot.*, to be published.

[34] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization-based full body control for the DARPA robotics challenge," *J. Field Robot.*, vol. 32, no. 2, pp. 293–312, 2015.

[35] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, Cambridge, MA, USA: MIT press, vol. 2, no. 3.

[36] A. Cully, K. Chatzilygeroudis, F. Allocati, and J.-B. Mouret, "Limbo: A flexible high-performance library for Gaussian processes modeling and data-efficient optimization," *J. Open Source Softw.*, vol. 3, no. 26, 2018, Art. no. 545.

[37] S. James, A. J. Davison, and E. Johns, "Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task," in *Proc. Conf. Robot Learn.*, 2017, pp. 334–343.