Minimum-Effort Task-based Design Optimization of Modular Reconfigurable Robots

Edoardo Romiti, Navvab Kashiri, Jörn Malzahn and Nikos Tsagarakis

***Abstract*— The flexibility and adaptability of modular and reconfigurable robots opens up new opportunities for on-demand robot morphology optimization for varying tasks. In particular, multi-arm robotic systems can expand the solution space for any given task. In this paper, we present a novel approach to exploit this feature for generating optimal fit-to-task robot structures with respect to a minimum-effort objective. By describing the task in terms of relative poses between the end-effector and the constraint frame, and making use of the relative Jacobian, the minimum effort optimization problem can be equally expressed for single-arm or multi-arm robots. We test our approach for a peg-in-hole and a contour-following task and compare the performance of the optimal solution obtained with that of a standard manipulator configuration.**

I. INTRODUCTION

Current manufacturing trends of shorter product lifecycles, growing numbers of product variations, smaller lot sizes and mass-customization demand collaborative robots, which can quickly be adapted to frequently varying production processes [1]. This paper focuses on reconfigurable robots composed of modules, which the user can quickly and reversibly disconnect and reconnect in a variety of configurations and on a timescale of seconds to few minutes. They offer the required versatility to realize fit-to-task kinematic arrangements on demand and with minimal down-times of the production line. Their physical embodiment can be customized and optimized to meet varying task requirements in terms of payload, workspace, environment constraints and task dexterity with maximum energetic efficiency. By composing the reconfigurable robot using only the minimum degrees of freedom (DOF) required by the task, they also exhibit an energetic benefit over standard six- or seven-axis robots. This work leverages on the relative Jacobian [2], [3] to distribute these minimum required DOF across multiple kinematic chains of an optimized reconfigurable robot to further reduce the energy demand.

The related literature focuses on the optimization of continuous parameters, such as the robot limb lengths, while keeping the morphology (e.g. number of actuators and their connectivity) of the robot design fixed. The work in [4] proposes a self-reorganized reconfigurable robot system along with a decision policy that takes into account the task work points, the required accuracy and forces to derive a suitable robot configuration. While evolutionary techniques [5], [6] have demonstrated to be effective in deriving the robot configuration, the work in [7] introduces a heuristic approach that optimizes the design of a robotic system starting from user-provided motion tasks. The method can successfully derive the robot design starting from these required motions. In [8] a task-based robot design for fixed configuration robots was introduced. In the proposed formulation all design parameters are assumed continuous values, and hence conventional optimization techniques are employed. This problem is typically referred to as kinematic synthesis [9], [10].

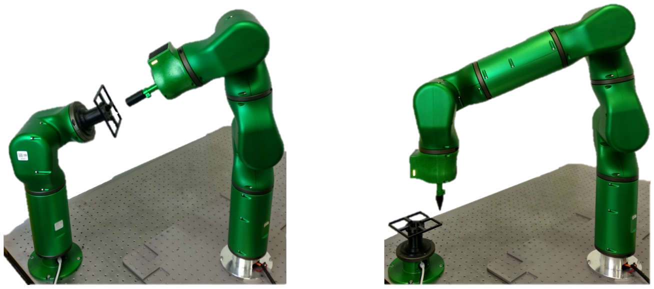


Fig. 1: Two robot configurations for the peg-in-hole (left) and contour-following task (right).

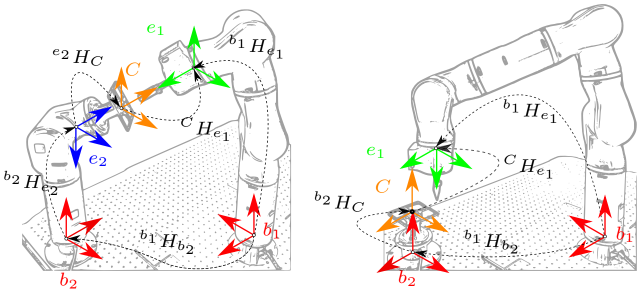
The contribution of this work is a novel method to derive the minimum effort arrangement of reconfigurable modular robots for a given manipulation task. In contrast to previous works, the proposed method evaluates the discrete space of feasible robot designs considering also multiple kinematic chains to derive the optimal robot configuration, minimizing the effort during task execution. In addition, this work presents a holistic process, which enables the automatic on-the-fly module discovery, kinematic topology identification, model generation and software/control configuration of the reconfigurable robot. This process enables the user to build and run the optimal robot from real modules and for the real-world task within only a few minutes after the optimization.

The article is organized as follows. In Section II we introduce the problem definition while Section III details the design optimization problem and associated performance indexes. Section IV presents the reconfigurable robot system and holistic auto-configuration process. Section V describes the task-based design optimization method. Section VI discusses the experimental results obtained with the reconfigurable robot for a peg-in-hole and a contour-following task. Section VII concludes with an outlook to future works.

II. PROBLEM DEFINITION

The objective of this study is to derive the optimal modular robot configuration for a given manipulation task or set of tasks, considering also multi-robot arrangements. Assuming common manipulation tasks, without loss of generality, we

The authors are with the Humanoids and Human-centered Mechatronics Lab, Istituto Italiano di Tecnologia, Genova, Italy edoardo.romiti@iit.it



I. dual-arm case

II. single-arm case

Fig. 2: The representation of main coordinate frames for the single/dual-arm robot manipulation task.

focus on deriving the optimal robot configuration for single- and dual-arm configurable robot setup that allow to execute these tasks with the lowest energy consumption.

An extensive literature exists regarding compliant control of robots performing manipulation tasks [11], [12], where both the position and the contact forces of the end-effector need to be controlled. It is convenient to describe the tasks in a coordinate frame defined with respect to the task space and geometry, where natural and artificial constraint can be defined [13]. In assembly manipulation tasks for example, defining a *constraint frame* with respect to the part geometry allows to describe the end-effector behavior in an easier way [11]. Furthermore, the task of a dual-arm manipulator can be conveniently expressed in terms of relative poses between the two end-effectors. This allows to simplify the manipulation problem of the dual-arm system to that of a single-arm-like manipulator. While such a problem can have a high degree of redundancy if two conventional robots are considered (6-7 DOF), it can intentionally be designed to have just the right number of DOFs required for the task when deploying two modular and reconfigurable robot arms.

The trajectory resolution of the two cooperating arms is one of the fundamental problems in dual-arm systems. By considering the two robot arms as a single system, and exploring the relative Jacobian between the two end-effectors, the task-space trajectories can be generated in a constraint frame $\{C\}$ attached to one end-effector. In [2], [3] relative motions of the two manipulators are generated using the relative Jacobian pseudo-inverse. Fig. 2 illustrates the main coordinate frames associated with the two single/dual-arm manipulation tasks, where b_1 and e_1 denote the base and the end-effector of the first arm in both cases. In dual-arm case, b_2 and e_2 represent the base and the end-effector of the second arm, while in single-arm case b_2 represents a fixed coordinate frame rigidly connected to frame $\{C\}$ on which the task is described i.e. where the task is executed. By defining the task in this frame, the task definition is the same for both single-arm and dual-arm robots. Similarly, there is only one task Jacobian ${}^{e_1}J_C$ in both cases. It is obtained just by a rotation in the $\{C\}$ frame for the single-arm system, while a relative Jacobian is extracted for the dual arm robot, as described in [14]. This definition enables a unified task formulation for both, the single-arm and the multi-arm robot setup. It allows the derivation of single and multi-arm robot configurations within the same optimization. We can then evaluate the different arrangements of the robotic modules by composing single- and multi-arm systems with a total number of DOFs equal or greater than the number of DOFs required by the task. The final optimization parameter for enhancing the system energetic performance is the second base position b_2 relative to the first base b_1. This determines where the task is executed i.e. work-cell position.

III. The Proposed Approach

An index for evaluating the performance of a single or dual arm robot is the *force transmission ratio* between the joint torques and the task forces. The intersection of the force/velocity ellipsoids with a unit vector $\boldsymbol{u}$ representing a Cartesian direction, quantifies the force/velocity transmission ratio of the manipulator [15], [16]. By considering the manipulator Jacobian $\boldsymbol{J}$, the force transmission ratio is then:

$$\alpha = (\boldsymbol{u}^T \boldsymbol{J}\boldsymbol{J}^T \boldsymbol{u})^{-\frac{1}{2}}. \tag{1}$$

This concept was extended to multi-arm system in [17]. Manipulability ellipsoids may then be derived similarly for the multi-arm case [18]. In this work we chose to use this index as a constraint of the optimization problem, in such a way that a lower bound is set to the force transmission ratio:

$$\alpha(\boldsymbol{q}) = (\boldsymbol{u}^T \boldsymbol{J_r}(\boldsymbol{q})\boldsymbol{J_r}^T(\boldsymbol{q})\boldsymbol{u})^{-\frac{1}{2}} > \bar{\alpha}_{th}. \tag{2}$$

Having a lower bound on α, shown by $\bar{\alpha}_{th}$, a minimum force transmission ratio along $\boldsymbol{u}$ is set, thereby ensuring the capability of the designed robot to apply forces along a given direction $\boldsymbol{u}$. A higher value of α results to lower torques in the joints for given forces at the end-effector, therefore inherently leading to a better energetic economy. This can be particularly important for tasks presenting high forces on the tool along a given direction $\boldsymbol{u}$. For instance, a peg-in hole task or a drilling task imply high forces on the direction perpendicular to the contact surface, while a milling process implies elevated forces along the direction tangent to the desired path. For contactless tasks or in specific directions within tasks with contacts, the *manipulability measure* is obtained to account for the manipulation capability of the robot. A lower bound $\bar{w}_{th}$ is then applied on this measure:

$$w(\boldsymbol{q}) = \sqrt{\det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q}))} > \bar{w}_{th}. \tag{3}$$

In the optimization study of this work, the key performance index utilized to evaluate the different configuration solutions of the modular robotic system, is the minimum *effort* [19], [20]:

$$\epsilon(\boldsymbol{\tau}(t)) = \int_0^{t_{tot}} \boldsymbol{\tau}^T(t)\boldsymbol{\tau}(t)dt, \tag{4}$$

where $\boldsymbol{\tau}$ symbolizes the vector of the joint torques.

The effort required for the execution of a certain task T_j can be approximated by considering a set of poses $\tilde{\boldsymbol{q}}$ and the static joint torques $\boldsymbol{\tau}_s$ generated by the joints in those postures. With a discretization of the task trajectory in p_j points, (4) is approximated as follows:

$$\epsilon_{T_j}(\boldsymbol{\tau}_s(T_j)) = \textstyle\sum_{i=1}^{p_j} \Delta t_i(\boldsymbol{\tau}_s^T(\tilde{\boldsymbol{q}}_{\boldsymbol{i}})\boldsymbol{\tau}_s(\tilde{\boldsymbol{q}}_{\boldsymbol{i}})), \tag{5}$$

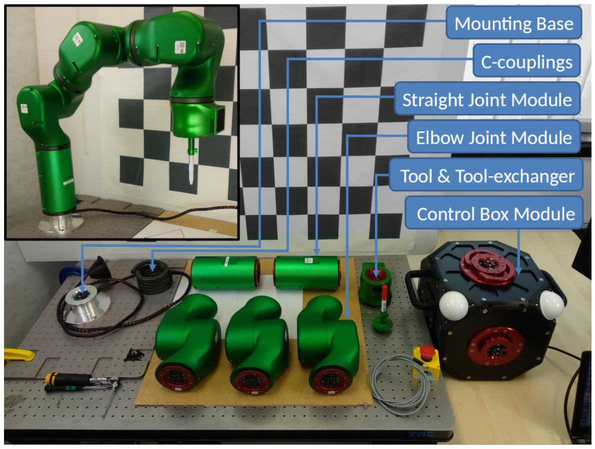


Fig. 3: The components of the reconfigurable robot include: joint modules, tool-exchanger module, mounting base and control box.

where Δt_i is the time the robot spends in the i-th pose $\tilde{\boldsymbol{q}}_{\boldsymbol{i}}$, and $\sum_{i=1}^{p_j} \Delta t_i = t_{T_j}$ is the total time to complete the task.

In the case where not only one, but a set of tasks need to be executed by the robot, the effort required by all distinct tasks can be minimized by considering a multi-objective cost function. If N different tasks are to be executed by the robot and each of them is repeated γ_j times, the effort related to the whole process can be computed as $\epsilon_{tot} = \sum_{j=1}^{N} \gamma_j \, \epsilon_{T_j}$. The robot configuration $\mathcal{D}$ can then be optimized to minimize all N task objectives, where γ_j acts as a weight for the j-th objective. The optimization problem to solve becomes then:

$$\begin{aligned} \min_{\mathcal{D},{}^{b_1}\boldsymbol{H}_{b_2}} \quad & \epsilon_{tot} = \sum_{j=1}^{N} \gamma_j \sum_{i=1}^{p_j} \Delta t_i(\boldsymbol{\tau}^T(\tilde{\boldsymbol{q}}_{\boldsymbol{i}})\boldsymbol{\tau}(\tilde{\boldsymbol{q}}_{\boldsymbol{i}}))) & \\ \text{s.t.} \quad & {}^{C}\boldsymbol{H}_{e_1}(\mathcal{D}, {}^{b_1}\boldsymbol{H}_{b_2}, T_j, \tilde{\boldsymbol{q}}_{\boldsymbol{i}}) = {}^{C}\bar{\boldsymbol{H}}^{i}_{e_1} & 1 \leq j \leq N \\ & \alpha(\mathcal{D}, {}^{b_1}\boldsymbol{H}_{b_2}, T_j, \tilde{\boldsymbol{q}}_{\boldsymbol{i}}) > \bar{\alpha}_{th} & 1 \leq i \leq p_j \\ & w(\mathcal{D}, {}^{b_1}\boldsymbol{H}_{b_2}, T_j, \tilde{\boldsymbol{q}}_{\boldsymbol{i}}) > \bar{w}_{th} & (6) \end{aligned}$$

where the robot design configuration $\mathcal{D}$ and the coordinate transformation ${}^{b_1}\boldsymbol{H}_{b_2}$ are the optimized variables. The optimized design configuration must satisfy constraints on the reachability of the p_j relative poses ${}^{C}\bar{\boldsymbol{H}}^{i}_{e_1}$ defined by the N tasks. Moreover, on each of these points, lower limits on the manipulability measure and force transmission ratio along a certain direction might be present. If one task does not involve interaction and contact forces, the force transmission ratio constraint can be omitted.

IV. The Reconfigurable Robotic System

The in-house developed reconfigurable robot prototype in Fig. 3 consists of a number of actuated joint modules, a mounting base, an end-effector and a control box module.

A. Module Description

Each module features one or more gendered electromechanical flange interfaces. The flange interfaces permit a reliable mechanical, power and EtherCAT communication connection between any two modules. The connection is securely locked with a pair of identical C-shaped couplings establishing form-closure around the cone shaped mechanical interface part. Each joint module is powered by an "orange" size Alberobotics actuator [21], [22] realized by the combination of a Brushless DC (BLDC) motor, a Harmonic Drive (HD) transmission with a 160:1 gearing ratio and a torque sensor. The "straight" type joint modules are actuated about the common central normal of the modular flange interfaces, while the "elbow" type joint modules rotate about a central axis orthogonal to the common normal of the modular flange interfaces. The mounting base module permits to position and mechanically fix one or more kinematic chains of joint modules.The control box module hosts a power supply and a compact PC with real-time operating system dedicated to execution of the centralized software and control modules. The end-effector module in this prototypical implementation includes a magnetic actuator and serves as tool-exchanger to quickly switch between tools in different tasks.

B. Automatic Software Reconfiguration

The method used in this paper is illustrated in Fig. 4. It is capable of handling multiple module chains as well as tree-like robot topologies. After the physical assembly (*step 1*) of the reconfigurable robotic system, the chains of the interconnected modules form an EtherCAT network. Scanning the network, reveals all participant modules and determines their parent-child relationship within the network topology tree (*step 2*). A unique identifier stored in each module enables to look up each module's kinematic and dynamic parameters along with semantic information in a centralized database (*step 3*). This information is then used to automatically reconstruct the physical robot topology (*step 4*) and the corresponding kinematic and dynamic models of each module chain. Only in the case of remotely fixed module chains, the user has to manually define the position (b_2) of the distal mounting base modules. With the auto-generated kinematic and dynamic models, the impedance control architecture of the reconfigurable robot systems is then auto-configured without any user intervention (*step 5*).

The combination of the fast electromechanical interfaces provided by the robot modules with the above fully automated process of physical system discovery, topology identification, model generation and software and control configuration enables a user to quickly implement the optimal robot configuration derived from the methodology of Sec. III and subsequently verify its efficacy in the execution of the targeted real-world tasks.

V. Task-based Design Optimization

We use bit string encoding to represent the candidate robot configuration [6], [23]. With mainly two types of joint modules and no passive links of variable length, the maximum number of possible assembly configurations for the presented prototype is still low, making it feasible to just compute all of them. The bit string encoding is convenient and allows the efficient transition to more elaborate search techniques such as Genetic Algorithms [24] in forthcoming works with more module types.

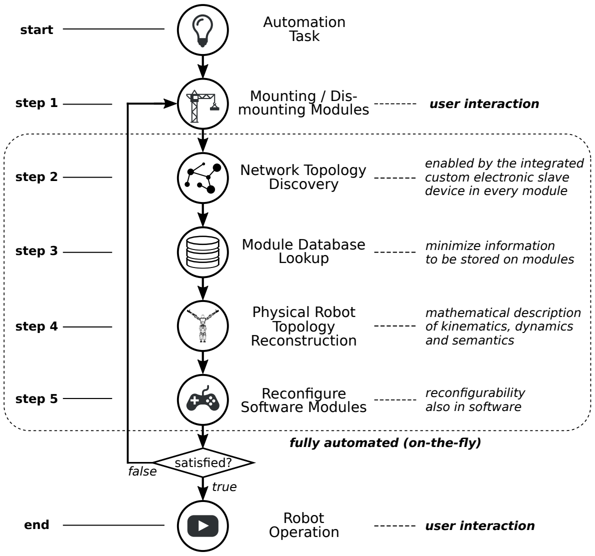

start
Automation Task
step 1
Mounting / Dis-mounting Modules
user interaction
step 2
Network Topology Discovery
enabled by the integrated custom electronic slave device in every module
step 3
Module Database Lookup
minimize information to be stored on modules
step 4
Physical Robot Topology Reconstruction
mathematical description of kinematics, dynamics and semantics
step 5
Reconfigure Software Modules
reconfigurability also in software
fully automated (on-the-fly)
satisfied?
false
true
end
Robot Operation
user interaction

Fig. 4: Overview of the automatic robot recognition and software reconfiguration.

A bit string represents a robot kinematic chain with every bit denoting a module: 0 for straight Joint module, 1 for elbow Joint module. The left-most bit encodes the first module in the chain. An n-bit string represents univocally an n-DOF robot design $\mathcal{D}$ assembled from Joint modules. We partition this bit string into sub-strings to also represent different designs of multi-chain robots with a combined total of n DOFs. Separating the string at multiple positions x_i, with $0 < x_i < n$ and $1 < i < l$, we generate l sub-strings, where l corresponds to the number of robot kinematic chains.

The number of robot configurations with n-DOFs when considering single-chain robots is 2^n. When considering also multi-chain robots, the number of configurations with n-DOFs in total becomes $2^n m$ where m is the number of combinations of kinematic chains

$$m = 1 + \textstyle\sum_{a=2}^{l} |\{(\zeta_{a,1}, ..., \zeta_{a,a}) \in \mathbb{N}^a : \forall d \in [2, a] \\ \zeta_{a,d-1} \geq \zeta_{a,d} \quad \wedge \quad \textstyle\sum_{b=1}^{a} \zeta_{a,b} = n\}|,$$

when the ordering of the chains is not relevant. For instance, the number of combinations when considering robot structures with up to $l = 2$ kinematic chains is e.g. $64 \times 4 = 256$ with six DOFs, and $32 \times 3 = 96$ with five DOFs.

In this optimization procedure, the position and orientation of the eventual l-th robot base ${}^{b_1}\boldsymbol{H}_{b_l}$ are the other variables to be optimized. The dimension of the space to search is dependent on the size of the work-cell intended for the robot. Techniques such as octree encoding [25] could be used to efficiently represent larger 3D work-cells. In this work, we presume the origin of the two frames to be in coinciding $x - y$ planes, having only discrete orientation offset along the z axis. The search space for the x, y position coordinates of the base frames is a square of R with steps of h, resulting in a grid of $(\frac{R}{h} + 1)^2$ points. The first chain base is placed at the $(0, 0)$ coordinates by default, which is therefore excluded from candidate positions, i.e. $(\frac{R}{h} + 1)^2 - 1$ points. Due to the symmetry of the modules we only consider orientations of $\frac{\pi}{2}$ and π with respect to the first chain base. An exemplary case with a total of five DOFs with up to two kinematic chains, when the base frames is placed within a square of $0.6\,\mathrm{m}$ with steps of $0.2\,\mathrm{m}$, leads to $32 \times 3 \times 15 \times 2 = 2880$ combinations. To avoid implausible combinations, a set of connection rules between modules are applied e.g. to prevent from connecting subsequent modules with collinear rotational axes, and to also account for possible limitations on the number of available modules of each type.

Kinematic and dynamic simulations can then be used to evaluate the performance of each design. In this optimization process only a kinematic simulation is performed, where a velocity-based inverse kinematics is used to generate the trajectory and check the feasibility of the defined task with respect to the constraints. The candidate robot must reach a finite number of task points, which are assumed to approximate the given trajectory. Constraints considered include a reachability constraint and a manipulability constraint as described in (6). The reachability is checked by imposing thresholds on the errors of position and orientation on the waypoints of the trajectory. Lower bounds are set for the manipulability w and for the force-transmission ratio α along the task tangential direction. During the task execution the predicted gravity compensation torques are computed and recorded along with the end-effector poses. In this step also the performance index is computed as in (5).

As the objective of the optimization is to minimize joint effort and effectively energy consumption, we target robot structures having the minimal overall number of DOFs to satisfy the task specifications. In this paper we consider a peg-in-hole task, which requires five DOF. Only arrangements of robots with a total number of five joints are considered. We consider also a surface-following task as a second task for the same robot. This leads to a multi-objective optimization problem. We follow a hierarchical approach. The module configuration is first optimized for the task requiring the maximum number of DOF. Solutions for the other task with lower DOF must then reside in the solution set obtained for the first task. This reduces optimization run time.

VI. Results

A. Optimization and Simulations

In this section, we evaluate the best robot assembly configuration to perform a peg-in-hole task and a contour-following task. Since the former task requires larger number of DOF, we first evaluate a set of candidates satisfying the constraints of this task. Pose errors are evaluated with respect to two goal poses ($p_j = 2$) defining the task: ${}^{C}\bar{\boldsymbol{H}}_{e_1}^{1}$ and ${}^{C}\bar{\boldsymbol{H}}_{e_1}^{2}$. These two relative poses between the end-effector frame and the constraint frame correspond to a pre-insertion pose where the peg is centered with the hole at a distance of 0.05 m from it and a final pose where the two frames are still aligned and the peg is inserted for the whole length: 0.05 m. The threshold on the error between the relative and actual poses is of $0.001\,\mathrm{m}$ for position and of $0.005\,\mathrm{rad}$ for orientation. The lower limits on the manipulability measure w and the force-transmission-ratio α are set to 0.02 and

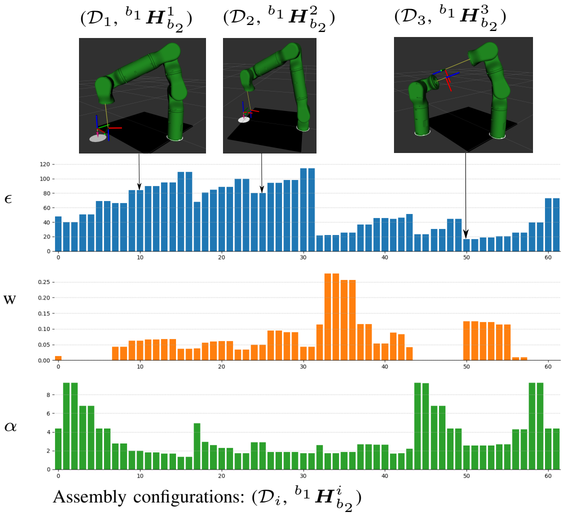


Fig. 5: Effort, manipulability and force transmission ratio on the first task optimization (peg-in-hole).

1.0. The trajectories of the two manipulators are generated by using the relative Jacobian ${}^{e_1}J_C$ and solving a velocity-based inverse kinematics using [26]. The *effort* index for this task can be computed evaluating (5) in the two poses. Fig. 5 shows the result of this first step of the optimization, where the effort index, the manipulability measure and the force-transmission-ratio are plotted for every robot configuration satisfying the reachability constraint. A total of 62 configurations satisfy the task constraints. It shows there are a number of configurations that despite of presenting low effort index, exhibit very low manipulability index w and are therefore discarded. The number of candidates after the first step is then 43. The second step optimization runs on the remaining candidates, to evaluate the execution of a contour-following task. Pose errors are evaluated with respect to five waypoints ($p_j = 5$) discretizing the end-effector trajectory in the constraint frame. This task simulates a glueing operation where the robot end-effector follows the contour of a surface at a small distance. The poses are defined on the edges of a square with a side length of 0.011 m while keeping a distance of 0.01 m from the constraint frame. The same procedure as the first task is used to simulate the task; however in the inverse kinematics, the relative position of the end-effectors have priority over the orientation since the relative orientation is not important in this task. The manipulability is then computed by only considering the reduced translational Jacobian. Being a contactless task, the force-transmission-ratio constraint is omitted. The effort index is computed as before using the static torques at the five waypoints of the trajectory and the overall effort of the two task is evaluated. The combined effort index of the two tasks is displayed in Fig. 6, together with the manipulability measure. The minimum-effort solution results to the dual-arm with 3+2 DOF ($\mathcal{D} = \mathcal{D}_3 = [[0, 1, 1], [0, 1]]$) and the coordinates of b_2 $(0.4, 0.4)$ with an offset angle of π with respect to b_1.

B. Experiments

For the experiments, the optimal robot for the two tasks is built (Fig. 1, left) and its performance is compared to that of a single chain 5-DOF robot (Fig. 1, right). Thanks to the prototype's reconfigurability features detailed in Sec. IV, switching between the two robot configurations is a matter of only minutes. We chose to compare the winner design $\mathcal{D}_3$ with $\mathcal{D}_1$, one of the best designs for the 5-DOF robots in terms of torque and manipulability, as presented in Fig. 5. The robot is controlled using a Cartesian impedance controller, where the relative poses obtained from the simulation are sent as reference and played back. To operate the robot we make use of XbotCore [27] as software framework and of CartesI/O as control interface [28].

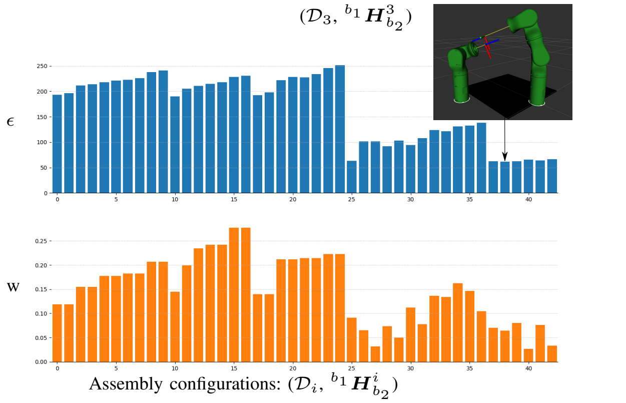


Fig. 6: Torques and manipulability of the resulting solutions after the 2nd task optimization (contour-following). The best robot design minimizing the effort index is also shown in the picture.

For the peg-in-hole task a very simple strategy was chosen where the arm holding the peg approaches the hole from the direction normal to the hole. The inevitable inaccuracy in the positioning of the peg with respect to the hole are compensated by the compliant control strategy, with an appropriate choice of the gains as suggested by [29]. In this experiment a peg of diameter 0,024 m and length 0,05 m is inserted in a hole of diameter 0,025 m. Fig. 7 shows the reference and actual values of the relative poses, the points where the peg enters in contact with the hole, and the points where the insertion starts and the peg moves. We can see that both robots can equally perform the desired task. To analyze the effort index associated with the task, Fig. 9 and Fig. 11 illustrate the time evolution of the torques $\boldsymbol{\tau}$ during the execution of the task, which evidence significantly lower torques of the dual arm robot. For the contour-following task the concept is the same and Fig. 8 shows the results of the experiment. Fig. 10 and 12 show the time evolution of the torques during the execution of the task. It is evident the dual arm system completes the task using much lower torques.

To better estimate the energy spent over the task execution, considering $E = V \int_0^{t_t} i dt$, the motor currents are approximated based on $\boldsymbol{\tau} = \eta K_T r \boldsymbol{i}$, where the corresponding parameters for our actuators are: $\eta \simeq 0.7$, $K_T = 0.078\,\text{Nm/A}$, $r = 160$ and $V = 48\,\text{V}$. From the integral of the norm of the torque over time an estimate of the energy consumption for the different robots can be computed, and the corresponding results are reported in Tab. I. The effort and energy consumption of the dual arm system is over nine

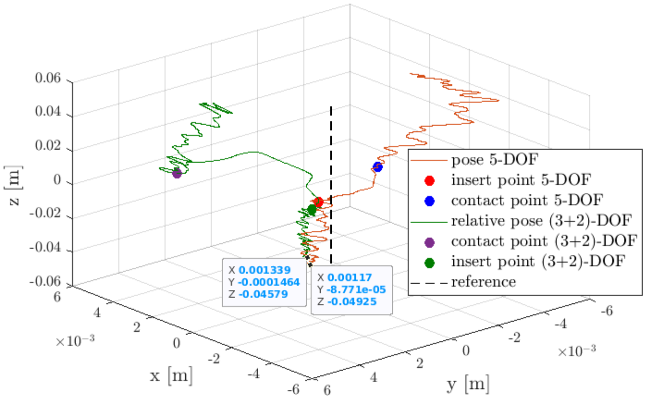


Fig. 7: 3D trajectories of the peg with respect to the hole for the 5DOF and 3+2DOF cases. Final error in position is reported for both cases.

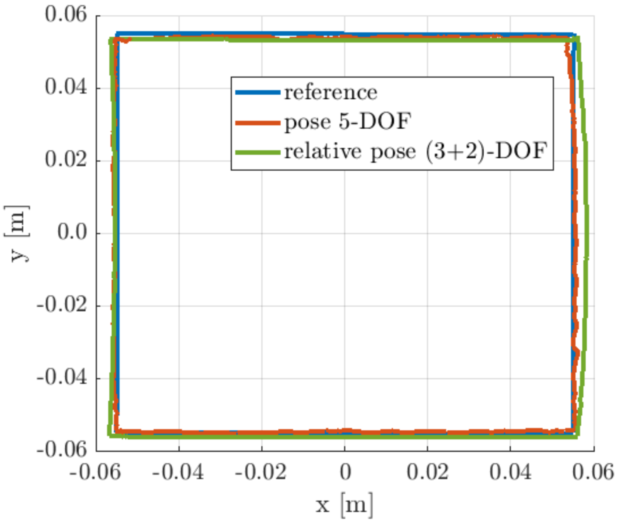


Fig. 8: The trajectories for the contour-following task with respect to the constraint frame for the 5-DOF and (3+2)-DOF cases.

TABLE I: Values of estimated energy consumption for each robot and task considered.

	$\int_0^{t_{tot}} \boldsymbol{\tau} dt$	E
5-DOF peg-in-hole	1.7607e+05	9.67417e+02
(3+2)-DOF peg-in-hole	1.8582e+04	1.02098e+02
5-DOF contour-following	1.3083e+06	7.188461e+03
(3+2)-DOF contour-following	1.3100e+05	7.19780e+02
	$[Nm \cdot s]$	$[kJ]$

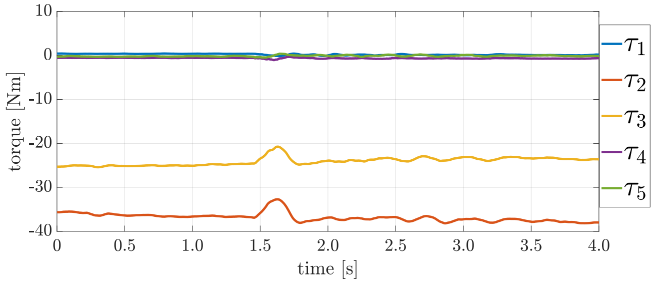


Fig. 9: Torques of the 5-DOF robot for the peg-in-hole tasks.

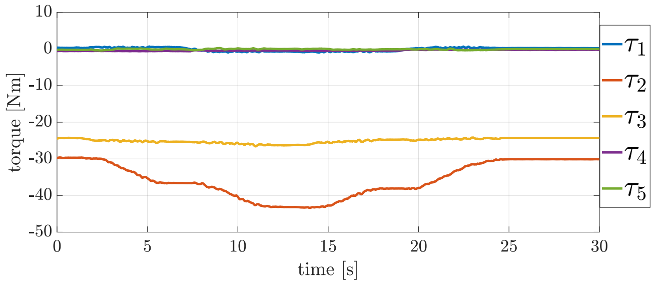


Fig. 10: Torques of the 5-DOF robot for the surface-following task.

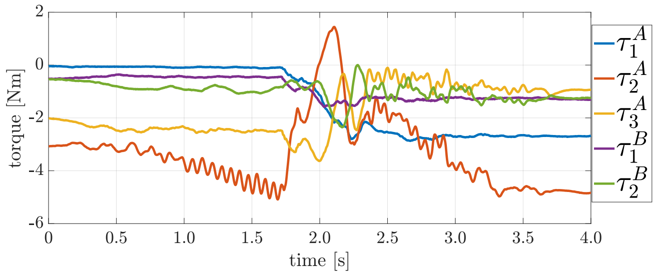


Fig. 11: Torques of the (3+2)-DOF robot for the peg-in-hole tasks. Superscript A and B indicate the 3-DOF and the 2-DOF chains.

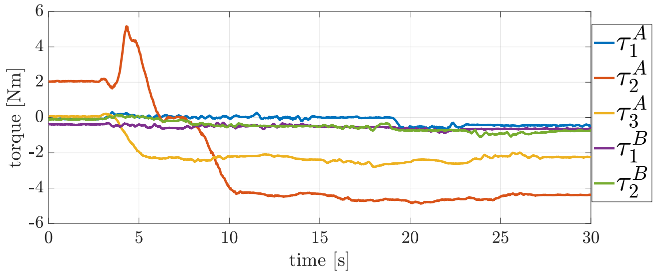


Fig. 12: Torques of the (3+2)-DOF robot for the contour-following tasks. Superscript A and B indicate the 3-DOF and the 2-DOF chains.

times smaller/better than that of the single arm system when executing the same task.

VII. CONCLUSIONS

The paper presents a new approach for the optimal task-based design of reconfigurable robot given a finite set of modules. We showed how, from task specifications it's possible to generate and evaluate a set of possible assembly solutions with the goal of minimizing energy consumption. The proposed approach unifies the optimization problem formulation for single-arm and multi-arm robots. In this way, a heterogeneous group of robotic systems tailored to the task is found. With the exemplified tasks we show experimentally how the optimal solution, a dual-arm configuration, reduces greatly the energy consumption compared to a traditional single-arm manipulator configuration. Future works will investigate more advanced search algorithms such as GA or A* algorithm for sped-up optimization convergence even with an expanded library of robot modules. Next, the benefits of reconfigurable multi-arm systems will be further explored using objective indices beyond minimum joint effort.

ACKNOWLEDGMENT

This work has received funding from the European Union's Horizon 2020 project CONCERT under grant agreement No. 101016007

References

[1] Y. Koren, *The global manufacturing revolution: product-process-business integration and reconfigurable systems*. John Wiley & Sons, 2010, vol. 80.

[2] C. L. Lewis and A. A. Maciejewski, "Trajectory generation for cooperating robots," in *1990 IEEE International Conference on Systems Engineering*. IEEE, 1990, pp. 300–303.

[3] W. S. Owen, E. A. Croft, and B. Benhabib, "Acceleration and torque redistribution for a dual-manipulator system," *IEEE transactions on robotics*, vol. 21, no. 6, pp. 1226–1230, 2005.

[4] T. Fukuda and S. Nakagawa, "Dynamically reconfigurable robotic system," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 1581–1586.

[5] C. Leger *et al.*, *Automated synthesis and optimization of robot configurations: an evolutionary approach*. Carnegie Mellon University USA, 1999.

[6] I.-M. Chen and J. W. Burdick, "Determining task optimal modular robot assembly configurations," in *proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 1995, pp. 132–137.

[7] S. Ha, S. Coros, A. Alspach, J. M. Bern, J. Kim, and K. Yamane, "Computational design of robotic devices from high-level motion specifications," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1240–1251, 2018.

[8] C. Paredis and P. K. Khosla, "An approach for mapping kinematic task specifications into a manipulator design," 1991.

[9] S. Manoochehri and A. Seireg, "A computer-based methodology for the form synthesis and optimal design of robot manipulators," 1990.

[10] S. Shirafuji and J. Ota, "Kinematic synthesis of a serial robotic manipulator by using generalized differential inverse kinematics," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 1047–1054, 2019.

[11] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," 1981.

[12] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[13] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[14] R. S. Jamisola, P. Kormushev, D. G. Caldwell, and F. Ibikunle, "Modular relative jacobian for dual-arms and the wrench transformation matrix," in *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE, 2015, pp. 181–186.

[15] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[16] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[17] P. Chiacchio, S. Chiaverini, L. Sciavicco, B. Siciliano, *et al.*, "Global task space manipulability ellipsoids for multiple-arm systems," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 5, pp. 678–685, 1991.

[18] M. Faroni, M. Beschi, A. Visioli, and L. M. Tosatti, "A global approach to manipulability optimisation for a dual-arm manipulator," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2016, pp. 1–6.

[19] D. E. Kirk, *Optimal Control Theory*. Prentice-Hall, 1970.

[20] B. J. Martin and J. E. Bobrow, "Minimum-effort motions for open-chain manipulators with task-dependent end-effector constraints," *The International Journal of Robotics Research*, vol. 18, no. 2, pp. 213–224, 1999.

[21] N. Kashiri, L. Baccelliere, L. Muratore, A. Laurenzi, Z. Ren, E. M. Hoffman, M. Kamedula, G. F. Rigano, J. Malzahn, S. Cordasco, *et al.*, "Centauro: A hybrid locomotion and high power resilient manipulation platform," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1595–1602, 2019.

[22] "Alberobotics," https://alberobotics.it/, accessed: 2020-03-30.

[23] O. Chocron and P. Bidaud, "Genetic design of 3d modular manipulators," in *Proceedings of International Conference on Robotics and Automation*, vol. 1. IEEE, 1997, pp. 223–228.

[24] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," 1988.

[25] H. Noborio, S. Fukuda, and S. Arimoto, "Fast interference check method using octree representation," *Advanced robotics*, vol. 3, no. 3, pp. 193–212, 1988.

[26] A. Rocchi, E. M. Hoffman, D. G. Caldwell, and N. Tsagarakis, "OpenSoT: A whole-body control library for the compliant humanoid robot COMAN," *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 6248–6253, 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7140076/

[27] L. Muratore, A. Laurenzi, E. Mingo Hoffman, and N. G. Tsagarakis, "The xbot real-time software framework for robotics: From the developer to the user perspective," *IEEE Robotics Automation Magazine*, vol. 27, no. 3, pp. 133–143, 2020.

[28] A. Laurenzi, E. M. Hoffman, L. Muratore, and N. G. Tsagarakis, "Cartesi/o: A ros based real-time capable cartesian control framework," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 591–596.

[29] 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 *IEEE ISR 2013*. IEEE, 2013, pp. 1–5.