2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

One Problem, One Solution: Unifying Robot Design and Cell Layout Optimization

Jan Baumgärtner[1], Alexander Puchta[1] and Jürgen Fleischer[1]

Abstract— **The task-specific optimization of robotic systems has since the inception of the field been divided into the optimization of the robot and the optimization of the layout of its workstations. In this letter, we argue that these two problems are interdependent and should be treated as such. To this end, we present a unified problem formulation that enables for the simultaneous optimization of both the robot kinematics and the workstation layout. We demonstrate the effectiveness of our approach by jointly optimizing a robotic milling system. To compare our approach to the state of the art, we optimize the robot's kinematics and layout separately. The results show that our approach outperforms the state of the art and that simultaneous optimization leads to up to eight times better solutions.**

I. INTRODUCTION

Although robots possess the inherent ability to perform a multitude of diverse tasks across various industrial domains, each individual robot is often limited to the execution of a singular repetitive task. This has opened the door for specialized robotic systems optimized for specific tasks. Optimization criteria vary from cycle time [1] to deformation over the task [2] and even the mechanical complexity [3]. These include modular systems such as presented in [4] but also complete custom solutions as in [5]. On the algorithmic side, this has raised interest in task-specific robot optimization. Examples include the work by [3] and [6] who try to find optimal robot designs for a given task based on modular components. However, in industrial settings, optimizing the layout of a robot's environment can often improve system performance. For instance, moving the task closer to the robot can reduce the required robot workspace and thus the required robot size. This indicates that the optimal robot design is subject to the environment in which it is used. In the optimal robot design literature, this is often neglected and the robot is assumed to be used in a fixed environment. Layout optimization meanwhile is itself a well-studied problem in the field of manufacturing. Here many works have tried to find optimal placements of workstations relative to a given robot. This is not trivial since the optimal placement of a robot can often depend on the robot's optimal joint trajectory. In summary, we have discovered that two optimization problems, usually considered separate in the literature, are interdependent. This work aims to unify both problems into a single optimization method that can be applied to different design problems.

*Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB-1574 – 471687386.

[1]Jan Baumgärtner, Alexander Puchta, and Jürgen Fleischer are with the wbk Institute of Production Science, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany jan.baumgaertner@kit.edu

Our main contribution is a unified problem description presented in section III. This description enables the application of new optimization algorithms explored in section IV. We not only use these to optimize continuous robot design spaces but also the kinematic structure of the robot. Additionally, we demonstrate how the problem can be extended to instead optimize robots built from predefined modules using a simple three-step approach. Our solutions are benchmarked against the existing state of the art for both placement and robot optimization in section VI. We show that our approach outperforms the state of the art in both subdomains and that it can also optimize the unified problem. Finally, we discuss the limitations of the problem formulation and our solution, as well as suggest future research directions in section VII.

II. RELATED WORK

Since this work aims to unify robot design and layout optimization it is heavily influenced by both fields. In this section, we will give a brief overview of the literature of both fields.

A. Robot Design Optimization

In the field of Robot design optimization, one can differentiate between two main subfields: Modular robot design tries to build a robot from predefined modules while continuous robot design relies on optimizing continuous kinematic parameters. Module-based design approaches tend to *grow* additional modules from the base until they reach the target position or trajectory. This is typically encoded using a reachability metric [3] which is either optimized using heuristic-based search algorithm [3] or using reinforcement learning [6]. Extending these growing approaches to layout optimization is not trivial since the performance of the growing robot is dependent on its optimal placement which will change as it grows, requiring nested optimization loops. For this reason, we will instead rely on continuous robot design optimization methods. Continuous robot design mostly focuses on optimizing the kinematics of a robot. These are described using various methods, with the Denavit-Hartenberg parameterization being a popular example due to its efficiency; it requires only 4 parameters (θ, d, a, α) per link, making optimization very efficient. We will also later use this parameterization for our sample problem. The main challenge in optimizing robot kinematics lies in mapping the desired task space waypoints to the joint and design parameter space of the robot. Many works use inverse kinematics or path planning to convert the desired task space waypoints to joint space [7]. They then use iterative methods such as genetic algorithms [5] [8] or particle swarm optimizers [7] to optimize the design parameters. Due to the need to convert the task space waypoints to joint space, these methods often require a large number of iterations making them inefficient. Additionally, they often fail to incorporate that different inverse kinematic solutions might have different performances. Recent works have thus formulated the problem explicitly using forward kinematics [9] resulting in nonlinear optimization problems (NLP). These NLP can be optimized using gradient-based methods to find not only the optimal design parameters but also the optimal joint trajectory. Tracking the desired waypoints is achieved using soft constraints forcing the robot end effector near the target waypoints. This approach was further developed in [10], which, despite its name, does not consider the environment and extends the method to include tool geometries that account for reaction forces with the environment. Our approach takes many ideas from these works, such as the use of link length regularization described in [9] but does not use forward kinematics. Unfortunately, neither approach considers the optimization of the layout of the environment around the robot.

B. Layout Optimization

The field of manufacturing science has a long history of optimizing the environment around a robot stemming from the field of factory planning. The goal is generally to optimize the performance of the cell. Therefore, the focus is on the placement of the functional parts with which the robot interacts. These functional parts will be referred to as workstations in the following. These workstations are generally abstracted to their respective toolpaths which are known a priori. This means that formally the problem is to find the optimal placement of toolpaths relative to a given robot. Thus, the optimization must address the same mapping problem encountered in robot design optimization. Completely parallel to robot design optimization, the problem was first formulated using inverse kinematics [1]. Here the authors tried to improve the cycle time of a dual-arm assembly station by minimizing the Euclidean distance between workstations. However, this is only an approximation since the shortest path of the end effector is not necessarily the fastest. Again we would require a trajectory optimization to evaluate a given design, and again this would result in a nested optimization loop. The solutions commonly used in mobile-manipulator base position planning suffer from a similar problem. A great overview of these can be found in [11]. Instead of using forward kinematics to solve this problem like for robot design optimization or in [12], recent works have instead used differential kinematics by formulating the placement problem as a modified optimal control problem [13]. Our approach is based on a similar differential kinematic formulation but incorporates the placement parameters differently.

III. PROBLEM FORMULATION

Robot design and layout optimization share a similar history in terms of problem formulation. Both started with an inverse kinematics formulation where the task space waypoints are converted to joint space waypoints and later switched to a forward kinematics formulation like in [9]. Recent work in the field of layout optimization has now argued that this is inefficient because it treats subsequent joint poses as independent. In general, both robot design and layout optimization have mainly been treated as design optimization problems where trajectory optimization had to be included somehow. We propose to instead think of these problems as trajectory optimization problems where the design parameters are included as additional joints which we will call design joints. Link lengths or workpiece positions might be thought of as additional prismatic design joints while the orientation of the workpiece might be thought of as additional revolute design joints. These design joints are then optimized together with the robot joints with the additional constraint that they have to be constant over time. Notice that this formulation does not differentiate between robot and layout design parameters. This means that we can isert these design joints between the robot and the workstation to get layout optimization or between the robot joints to get robot design optimization. Fig. 1 illustrates this for robot design optimization, workpiece placement optimization, and a combination of both. Thus unified the problem can be formulated as a modified trajectory optimization problem:

$$\begin{aligned} \underset{q(t_0),u}{\text{minimize}} \quad & \int_{t_0}^{t_f} wL_t(p(t)) + L_d(p(t),q(t),u(t))dt \\ \text{subject to} \quad & \dot{p}(t) = J(q(t))\dot{q}(t) \\ & \dot{q}_m(t) = u \\ & \dot{q}_d(t) = 0 \end{aligned} \tag{1}$$

where q is the full joint vector $q(t) = (q_m(t), q_d(t))^T$, $q_m(t)$ describes the moving joints and $q_d(t)$ the design joints. The vector p describes the task space position of the robot end effector and is updated using the Jacobian $J(q(t))$. The optimization variable u describes the joint velocities of the moving joints. The scalar w is a weighting factor that can be used to balance the tracking and design costs. In practice, the system is also often subject to numerous equality and inequality constraints such as joint limits or collision constraints. In contrast to previous works such as [9] constraints are not used to force the end effector to follow a given toolpath, instead, we introduce a tracking cost L_t.

$$L_t(p) = \|p(t) - s(t)\|^2 \tag{2}$$

where $s(t)$ is the desired toolpath the trajectory should track over the time interval $[t_0, t_f]$. The design costs such as cycle time, robot size, or manipulability are encoded in the design cost L_d. Examples can be found in [9], [13] or (3). Note that this problem formulation only considers kinematics and not dynamics. However dynamic design parameters can be included by substituting the Jacobian with a description of the robot's dynamics. For this work, we will only consider kinematic design parameters to better compare our approach to the state of the art which is mainly focused on kinematics. To show that this formulation is useful for optimizing both problems, the next section will introduce design problems and show how these can be optimized using this formulation.

IV. Example Application

To investigate the proposed formulation's effectiveness, We will consider the example of designing a robotic system with the task of milling a figure eight in a planar metal surface. The design objective is to optimize surface quality by minimizing the magnitude of the static deformation caused by robot compliance multiplied by applied forces during the toolpath $s(t)$:

$$L_d(q(t), s(t)) = \|C(q(t))F(s(t))\|^2 \tag{3}$$

A robot's compliance C is dependent on its pose through the Jacobian $J(q(t))$ of its forward kinematics and joint stiffness matrix K [2]

$$C(q(t)) = J(q(t))^T K^{-1} J(q(t)). \tag{4}$$

All moving joints $q_m(t)$ are assumed to be equally stiff, which is why their corresponding value in K can be simply set to 1 for optimization purposes. The design joints $q_d(t)$ are assumed to be infinitely stiff. The force $F(s(t))$ acting on the robot is a result of the tool biting into the material and is thus dependent on the toolpath. Assuming the tool is always fully engaged, the force can be approximated from the force model described in [14] as follows:

$$F(s(t)) = \alpha \frac{\dot{s}(t)}{\|\dot{s}(t)\|} \tag{5}$$

where α is a constant that depends on various process parameters assumed constant such as the material to be cut or other process parameters assumed constant. To optimize the robotic system we also need to define the design space. In the context of the trajectory optimization formulation, this means defining the design joints and their limits. The placement of the workpiece can be modeled using three prismatic and three revolute joints. This actually moves the robot and not the workpiece. However, using a coordinate transformation we can extract the position of the workpiece relative to the robot. The prismatic joints enable movement in the x, y, and z directions, constrained within a cuboid representing the work cell's available space. Three variations of the design space are considered. Firstly, the optimal design and placement of a conventional industrial robot modeled using Denavit-Hartenberg (DH) parameters with six revolute joints is examined. Since only continuous DH parameters are optimized we will call this problem the continuous design problem. Secondly, the design space is expanded to explore different kinematic structures, including joint type and number. This will be called the kinematic structure design problem. In both problems, the design joints representing physical size (DH parameters d or a) are constrained to be positive, and a regularization term L_r is added based on [9] that penalizes large values of d or a to prevent excessively large robots. Lastly, the modular design problem investigates the design and placement of a modular robot. Each module k is modeled by a set of DH parameters q_j. This example will use five different modules outlined in Table II. An overview of the three design spaces is provided in Table I.

Design Problem	Robot Design Space
Continuous	$A_c := \{d_i, a_i, \alpha_i \mid i \in 1, \ldots, 6\}$
Kinematic Structure	$A_s := \{n, \{d_i \wedge \theta_i, a_i, \alpha_i \mid i \in 1, \ldots, n\} \mid n \in 1, \ldots, 6\}$
Modular	$A_m := \{\text{Modules from Table II}\}$

TABLE I: Design space for each design problem

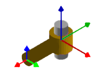
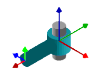
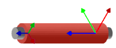
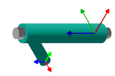
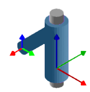
Visualization	θ	d	a	α
	*	0	0.5	0
	*	0	0.5	$\pi/2$
	*	1	0	0
	*	1	0.5	0
	*	0.5	0.5	$-\pi/2$

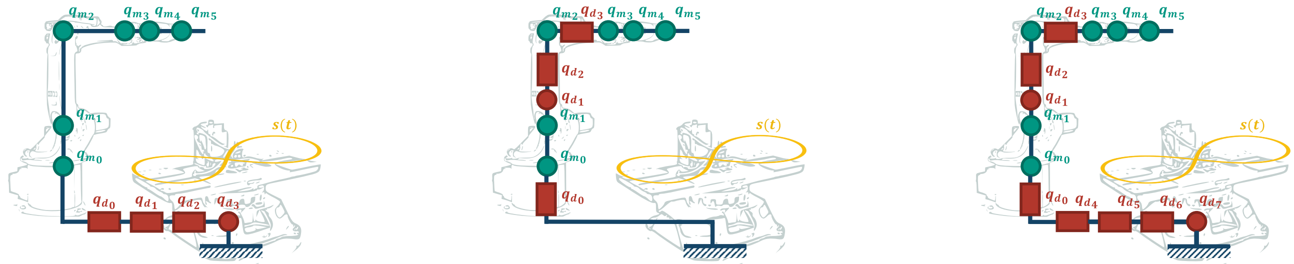
TABLE II: Modules in the modular design problem with visualization of the DH parameters. * describes the parameter being actuated.

V. Optimizing the Design Problems

In this section, we will optimize the design problems we have formulated. We begin with the continuous design problem, which can be directly formulated as a trajectory optimization problem. Subsequently, we demonstrate how the solution for the continuous problem can be extended to address the kinematic structure design problem and the modular design problem.

A. Optimizing the continuous design problem

The continuous design problem takes the form of the trajectory optimization problem defined in (1). To tackle this problem, we employ the direct collocation method [15]. Direct collocation is a type of trajectory optimization algorithm in which the joint trajectory is approximated using a set of piecewise polynomials. Instead of directly optimizing the joint trajectory, collocation optimizes the parameters of these polynomials. This results in a very sparse optimization problem that scales well with increasing degrees of freedom [15]. By setting the polynomial degree to 3 for moving joints and 0 for design joints, we ensure a smooth moving



(a) Workpiece placement problem (b) Robot design problem (c) Unified design problem

Fig. 1: Design joints for different design problems

joint trajectory while keeping the design joints constant. This implicitly satisfies the design joint constraint $\dot{q}_d(t) = 0$. All of these reasons make direct collocation particularly well suited for the continuous design problem. Solving the collocation yields the optimal joint trajectory $q_m(t)$ and optimal design joints $q_d(t)$.

B. Optimizing the kinematic structure design problem

Solving the kinematic structure problem involves determining the values of continuous design joints, identifying whether a joint is prismatic or revolute, and selecting the number of joints. As such it is a direct extension of the continuous design problem that adds several discrete decisions. While this might initially seem like a complex mixed integer problem, for a 6-DOF robot there are only 64 possible kinematic structures. These structures are typically encoded as a string where P represents a prismatic joint and R a revolute joint. And due to exponential scaling, considering all robots with 6 or fewer joints only yields 123 such structures. Given the efficient nature of optimizing the continuous design problem, it is feasible to simply optimize it for each structure. The kinematic structure design problem can thus be solved by optimizing the continuous design problem for each possible structure and then selecting the best one. Best in this case means the structure that yields the lowest design cost while still being able to track the toolpath. Although this might seem like a very naive approach, it can easily be parallelized and is guaranteed to find the optimal structure given a suitable continuous design problem solver.

C. Optimizing the modular design problem

The modular design problem initially appears quite different from the previous two problems because it only has discrete design joints. We also can't simply optimize over all possible combinations like in the kinematic structure design problem because the number of possible robots is now much larger. Instead, we propose to just assume that the design joints are continuous, then optimize the continuous design problem, and then round the solution to the nearest modular robot. For this to work, it is crucial that the continuous solutions closely resemble the modules. This is achieved by introducing an additional design cost, defined as:

$$L_{mod} = \sum_{i=1}^{6} \prod_{k=1}^{K} \|q_{d,i}(t) - q_k\|^2 \qquad (6)$$

Here, $q_{d,i}$ represents the vector describing the design joints in the i-th DH transformation, and $q_j(t)$ represents the vector describing the DH parameters of module j. This function is minimized if each set of design joints $q_{d,i}(t)$ is equal to one of the modules q_j. In practice, additional weights are added to balance the influence of the angular and linear components of $q_{d,i}(t)$. Depending on the size of the robot one can weigh the angular and linear components of $q_{d,i}(t)$ differently to prevent the angular components from dominating the cost. After optimizing this modified continuous design problem each design joint $q_{d,i}(t)$ is rounded to its closest module $q_j(t)$. The placement problem is then optimized again to account for potential changes resulting from the rounding process. This process assumes that modules share the same joint type and that the number of moving joints is predetermined. If this is not the case continuous optimization can be extended to the kinematic structure design problem.

VI. Results

In the introduction of this paper, we asserted that the robot design problem is intertwined with the workpiece placement problem. In this section, we examine this claim using the example application of an optimal milling system. We expect that an optimally placed and designed robot would outperform and look different compared to an optimally designed robot placed at the origin. Our measure of performance is the design cost (3) integrated over the time interval $[t_0, t_f]$ which encodes our objective of minimizing the deformation of the robot. For the optimization results to be meaningful, the performance of the optimizer needs to be benchmarked first. Therefore, we will conduct a series of experiments to evaluate the performance of the trajectory optimization approach for the continuous design problem. We will then investigate how the trajectory optimization approach performs on the kinematic structure design problem and the modular design problem.

A. Results for the continuous design problem

Before investigating how combining placement and design optimization affects performance we first benchmark the trajectory optimization approach (TO) against existing state-of-the-art approaches. Here we use the forward kinematics-based optimization approach (FK) proposed by Whitman et

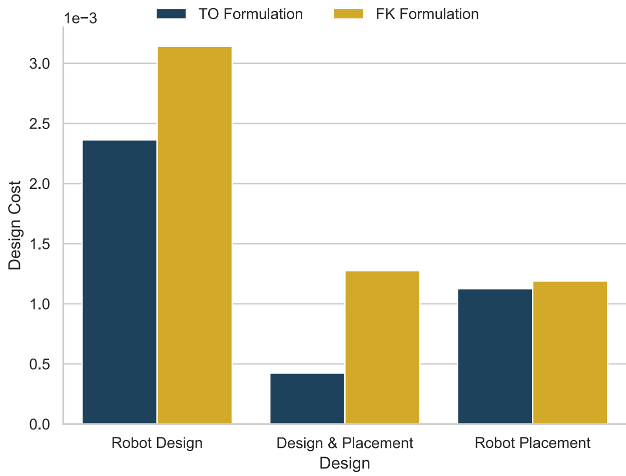
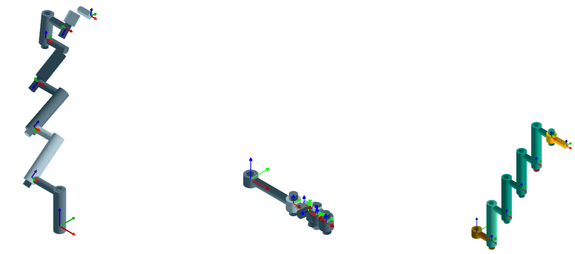


Fig. 2: Results of the 6R robot design and workpiece placement problem with design costs according to (3) integrated over the whole trajectory.



(a) Optimized robot (A)

(b) Optimized robot and placement (B)

(c) Optimized modules (C)

Fig. 3: Kinematic representation of the optimized robots for the three design problems of section IV.

al. [9]. In its continuous form it is given by:

$$\begin{aligned} \underset{q_m, q_d}{\text{minimize}} \quad & \int_{t_0}^{t_f} L_d(q_m(t), q_d) dt \\ \text{subject to} \quad & L_t(p(q_m(t), q_d)) \leq \varepsilon \end{aligned} \tag{7}$$

where ε is a small constant and L_t is the tracking cost defined in (1). Here q_d is not dependent on time. While this formulation was proposed for robot design optimization, using the notion of design joints it can also be used for placement optimization. For this reason, the FK approach will be used as a benchmark for both the placement problem and the combined problem. The resulting total design cost (3) is shown in Fig. 2. Since both algorithms require a discretization of the toolpath $s(t)$ we used 80 waypoints for both algorithms. Fig, 2 shows that the TO approach outperforms the FK approach in each case with the gap increasing as the number of design joints increases. This indicates that formulating the design problem as a trajectory optimization problem is a promising approach. The Figure also shows that the combined design and placement yields better results no matter the algorithm. One might simply attribute this to the fact that the combined problem has more degrees of freedom. However, placement alone performs better than design optimization alone, even though it has much fewer degrees of freedom. This indicates that placement is an integral part of the design process, in some cases even more important than the design of the robot itself. Looking at the resulting robots shown in Fig. 3, we see a clear difference. The robot that was jointly optimized and placed appears much smaller with shorter links and revolute joints closer to the end effector. This further indicates that the placement problem is not independent of the robot design problem and needs to be considered when optimizing a robot.

B. *Results for the kinematic structure design problem*

To evaluate the effectiveness of our proposed kinematic structure design for milling tasks, we compare its performance against what we believe to be the optimal configurations, based on insights from traditional serial milling machines. These machines, depending on their specific model, adopt kinematic configurations like PPPRRR, PPPRR, PPPR, or PPP, positioning revolute joints close to the tool or end effector to reduce deformation introduced by leverage effects. If our optimization algorithm is effective, it should similarly identify these configurations as optimal. Specifically, for our task of milling on a planar surface, the simplest three-degree-of-freedom (3-DOF) PPP structure is expected to be the most optimal. Looking first at the optimal 6-DOF robot given the number of prismatic joints in Fig. 4, we see that the optimal robot uses as few revolute joints as possible as expected. Looking at the right side of the figure we see that given several revolute joints the optimization indeed prefers them near the endeffector. The same is true for the 5-DOF robot as shown in Fig. 5, which also outperforms the 6-DOF robot. This trend continues for 4-DOF and 3-DOF robots leaving the PPP structure as the optimal structure. We thus conclude that our optimization approach can be used to effectively find optimal structures given the number of joints as well as the optimal number of joints themselves.

C. *Results for the modular design problem*

While existing approaches like [6] offer modular robot design methods, they cannot be extended to incorporate placement, as mentioned in the introduction. Hence, we can't use them as benchmarks for our modular design approach. However, our TO approach has already demonstrated its capability to optimize both robot design and placement. We thus mainly have to check whether introducing modular design cost (6) results in solutions that resemble the modules and how this affects the overall performance. To address this question we perform the modular design optimization and plot the resulting sets of design joints $q_{d,i}$ and modules q_j in a principal component analysis (PCA) plot (Fig. 6) to access their similarity. As we can see most joints can be clearly assigned to a module, indicating that introducing the modular design cost (6) results in solutions that resemble the modules. Note that the resulting modular robot looks very different from the original TO formulation as pictured in Fig. 3. Its performance however is similar with a design

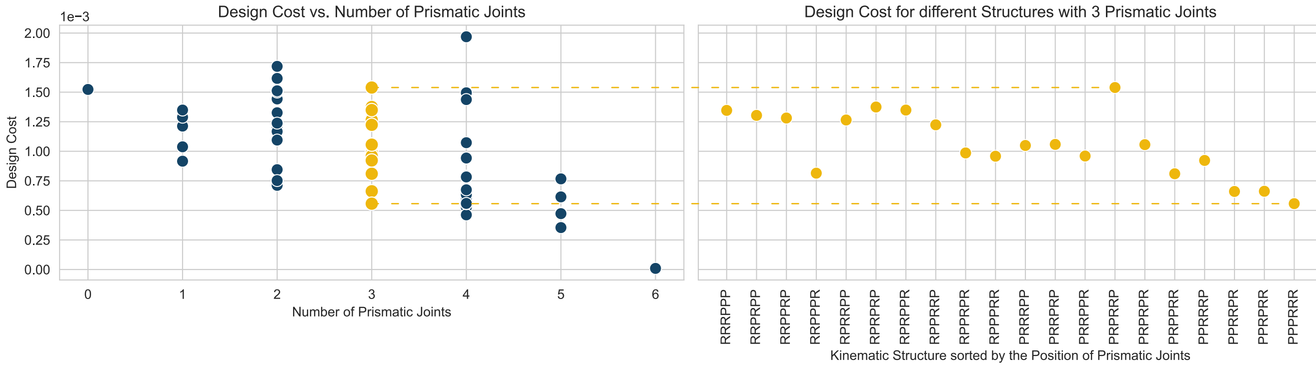


Fig. 4: Left: Deformation cost for the optimal 6-DOF robot given the number of prismatic joints. Right: Deformation cost for the optimal 6-DOF robot with three prismatic joints

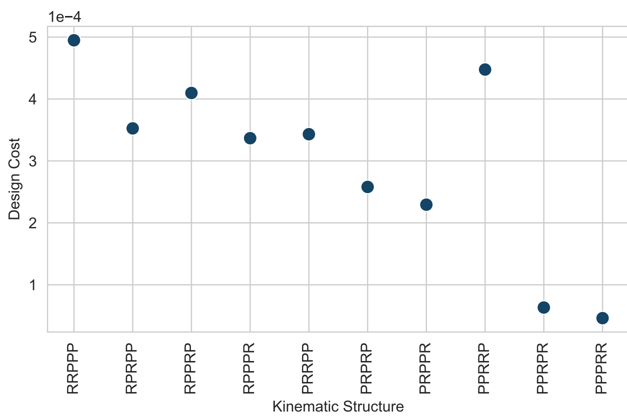


Fig. 5: Deformation cost for the optimal 5-DOF robot with three prismatic joints

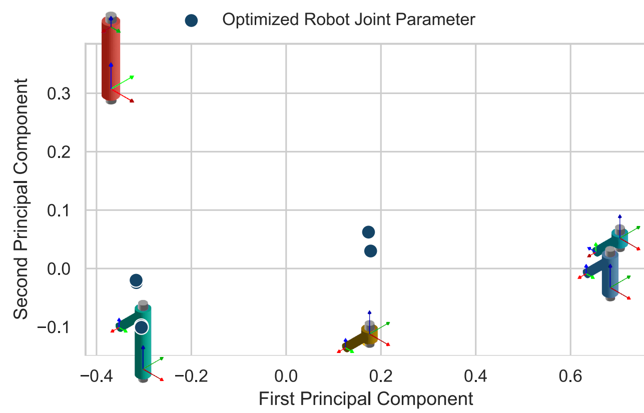


Fig. 6: PCA plot of the design joints $q_{d,i}$ and the modules q_j.

cost 6.72×10^{-4} compared to the original 4.16×10^{-4}. It also still outperforms the FK method with 1.261×10^{-3}. This indicates that our modular design approach is capable of finding suitable solutions and can extend modular robot design to incorporate placement optimization.

VII. Limitations and Future Prospects

The experiments presented in the previous section have shown that optimizing the design of a robot should also consider its placement. However, we believe that the field of robot and layout design optimization is still in its infancy and there are many open questions. In this section, we will outline some of these, highlighting the limitations of our approaches and offer future research directions. One area requiring further investigation is the integration of design joints into trajectory planning. The number and gaps between waypoints seem to greatly influence performance but the reasons remain unclear. Additionally, planning complex trajectories between distant waypoints, especially in the presence of intricate obstacles, presents challenges that collision costs like the one proposed in [9] may not adequately address. Integrating design joints into gradient-free optimizers like rapidly exploring random trees is an open question. In modular design, the influence of available modules on optimization performance and the impact of the number of modules are important considerations yet to be explored. A general practical concern is that current algorithms provide only a single solution, while different scenarios may have multiple equally valid solutions. To provide a comprehensive picture, presenting a set of solutions would be valuable.

VIII. Conclusion

In this work, we have taken a look at the long-standing problems of task-specific robot design and workpiece placement. We highlighted that both problems can be thought of as trajectory optimization, allowing for their unification. Using the example of a milling system we showed that combining these problems leads to an eight-fold improvement in performance. We believe that this has shown that if possible robot design and placement should be optimized together.

References

[1] T. Bachmann, K. Nottensteiner, and M. A. Roa, “Automated planning of workcell layouts considering task sequences,” in *Proceedings - IEEE International Conference on Robotics and Automation (ICRA)*. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 10 087–10 092.

[2] J. H. Lee, S. H. Kim, and B. K. Min, "Posture optimization in robotic drilling using a deformation energy model," *Robotics and Computer-Integrated Manufacturing*, vol. 78, 12 2022.

[3] 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.

[4] A. Yun, D. Moon, J. Ha, S. Kang, and W. Lee, "Modman: An advanced reconfigurable manipulator system with genderless connector and automatic kinematic modeling algorithm," *IEEE Robotics and Automation Letters*, vol. 5, pp. 4225–4232, 7 2020.

[5] T. Campos, J. P. Inala, A. Solar-Lezama, and H. Kress-Gazit, "Task-based design of ad-hoc modular manipulators," in *Proceedings - International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6058–6064.

[6] J. Whitman, R. M. Bhirangi, M. J. Travers, and H. Choset, "Modular robot design synthesis with deep reinforcement learning," in *AAAI Conference on Artificial Intelligence*, 2020.

[7] D. Ramirez, J. Kotlarski, and T. Ortmaier, *Combined Structural and Dimensional Synthesis of Serial Robot Manipulators*. Springer International Publishing, 2016, vol. 569.

[8] X. Hu, W. Wan, L. Du, J. Yuan, and S. Ma, "Multi-objective geometric optimization of a multi-link manipulator using parameterized design method," 2022, pp. 7310–7316.

[9] J. Whitman and H. Choset, "Task-specific manipulator design and trajectory synthesis," *IEEE Robotics and Automation Letters*, vol. 4, pp. 301–308, 4 2019.

[10] M. Toussaint, J.-S. Ha, and O. S. Oguz, "Co-optimizing robot, environment, and tool design via joint manipulation planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6600–6606.

[11] A. Kluge-Wilkes and R. H. Schmitt, "Mobile robot base placement for assembly systems: Survey, measures and task clustering," in *Production at the Leading Edge of Technology*, B.-A. Behrens, A. Brosius, W.-G. Drossel, W. Hintze, S. Ihlenfeldt, and P. Nyhuis, Eds. Cham: Springer International Publishing, 2022, pp. 439–447.

[12] J. Baumgärtner, A. Puchta, B. Bertschinger, G. Kanagalingam, O. Sawodny, S. Reichelt, and J. Fleischer, "Increasing robot precision by stroke division," in *2023 27th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2023, pp. 205–210.

[13] J. Baumgärtner, P. Gönnheimer, and J. Fleischer, "Optimal robot workpiece placement for maximized repeatability," in *Advances in System-Integrated Intelligence*. Springer International Publishing, 2023, pp. 252–261.

[14] H. Kim and K. Ehmann, "A cutting force model for face milling operations," *International Journal of Machine Tools and Manufacture*, vol. 33, no. 5, pp. 651–673, 1993.

[15] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.