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

Time-Optimal TCP and Robot Base Placement for Pick-and-Place Tasks in Highly Constrained Environments

Alexander Wachter[1], Andreas Kugi[1,2], and Christian Hartl-Nesic[1]

***Abstract*— This work proposes a highly parallelized optimization scheme to simultaneously optimize the robot base and tool center point (TCP) placement within a robotic work cell for a sequence of pick-and-place tasks. The placement is optimized for minimum cycle time by considering the scenario holistically, including point-to-point trajectory planning while respecting the kinodynamic constraints of the robot, collision avoidance in highly constrained environments, redundancy in grasp configurations and inverse kinematic solutions, and the cyclic constraint of the process. The proposed algorithm is applied to optimize the robot base and TCP placements in a spatially constrained packaging scenario, demonstrating a cycle time reduction of $41\,\%$ compared to state-of-the-art approaches. The results are validated experimentally using a KUKA LBR iiwa with 7 degrees of freedom, where the TCP placement is realized using topology optimization and 3D printing.**

I. Introduction

Robotic work cells are essential in modern manufacturing environments, where efficiency and throughput are critical. Achieving optimal performance in robotic cells entails careful attention to various components, including robot base placement, workpiece placement, trajectory planning, and the intricate design of robot end-effectors. Failure to optimize these key aspects can result in significant production delays, which in turn lead to increased production costs and energy consumption.

State-of-the-art approaches utilize performance metrics such as manipulability [1] and reachability indices [2] to guide optimization of the robotic work cell while often not considering the holistic process scenario [3], [4]. Especially in highly constrained environments, these methods fail to provide feasible placements. Consequently, it is essential to fully consider the obstacle environments, (free-space) point-to-point (P2P) robot motions, and all feasible robot configurations to obtain the optimal layout of a robotic work cell. Additionally, in pick-and-place tasks, an object may be grasped using different poses and must be placed using a compatible release pose. This is referred to as *grasp redundancy* and is, together with motion planning, essential to achieve minimum cycle times.

This paper proposes a mutual optimization procedure for robot base placement and tool center point (TCP) placement for pick-and-place tasks, considering (free-space) P2P robot motions, grasp redundancy, and a cyclic constraint. The primary objective is time optimality for a complete cyclic pick-and-place task. By utilizing the resulting trajectories and the robot model, forces acting on the TCP and the robot flange can be computed. This force profile is used in a topology optimization procedure, which generates a 3D-printable part to realize the optimal TCP placement in a real experiment. A video of the simulations and the experimental results is found at www.acin.tuwien.ac.at/3tcp. The main contributions of this work are as follows:

- an asynchronous, highly parallelized optimization procedure for robot base and TCP placement in pick-and-place tasks with grasp redundancy,
- the incorporation of a P2P trajectory planner considering collisions, kinodynamic limits, different robot configurations, multiple grasp poses, and a cyclic task constraint,
- an experimental verification of the optimal robot base and TCP placement using topology optimization and IMU (inertial measurement unit) measurements.

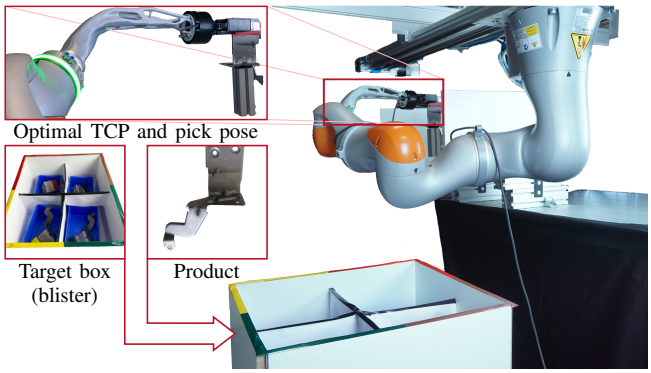

Fig. 1. Scenario of the experimental setup: Optimal robot base and TCP placement for a robot work cell, in which products are picked up from a manufacturing machine and sorted into a target box using time-optimal robot trajectories.

II. Related Work

The optimal placement of robots, their TCP, and all components within a work cell is crucial for achieving optimal task execution time and operational efficiency. Prominent state-of-the-art methods to evaluate the layout of a robotic work cell rely on well-known performance indices [5], e.g., the manipulability index [1], reachability index distributions [2], minimum singular value [6], and the Jacobian condition number [7]. These indices are computationally efficient and have also been used to design a complete layout of a work cell [8] by minimizing the traveled distance between task poses and maximizing the manipulability index.

[1] Alexander Wachter, Christian Hartl-Nesic, and Andreas Kugi are with the Automation & Control Institute (ACIN), TU Wien, Vienna, Austria, {wachter,hartl,kugi}@acin.tuwien.ac.at
[2] A. Kugi is with the AIT, Austrian Institute of Technology GmbH, Vienna, Austria, andreas.kugi@ait.ac.at
* This project is funded by the FFG (FO999896399). www.ffg.at

Optimization procedures incorporating full robot motions are considerably more demanding, as the robot trajectories must be updated for each robot base or TCP placement. For instance, in [9], a full trajectory planner for a given set of complex continuous paths was integrated into the optimization. Similarly, the relationship between cycle time variations and robot base placement for given paths was investigated in [3], although collisions were not considered. Both approaches are constrained to predefined paths only and do not account for free-space movements, which constitute a significantly larger search space and increase complexity due to collision avoidance.

The large search space when considering free-space robot motions poses additional challenges for trajectory planning. The problem of determining the shortest distance traveled by the manipulator for a spot-welding application was addressed in [10]. Similarly, the work [4] considers the shortest path and minimizes joint velocities for a given end-effector velocity while disregarding collisions. In contrast, [11] employed a two-stage optimization scheme, which first computes the reachability index map and then evaluates the actual cycle time for specific local minimums of the map. This approach only provides a list of base placement candidates rather than incorporating the cycle time within the optimization process. Integrating layout optimization with comprehensive robot trajectory planning has not yet been addressed, as noted in [12].

The literature has widely assessed grasp redundancy in the context of grasp optimization and motion planning. Optimization-based solutions [13] and sampling-based approaches [14] have been proposed for a discrete set of grasp positions [15] and continuous grasp poses [16]. To the best of the author's knowledge, grasp redundancy has not been addressed for optimizing the robotic work cell.

A passive mechanical adapter can be generated and printed using topology optimization [17] and 3D printing, which experimentally realizes the optimized TCP placement.

Topology optimization was also used to examine robots in the most stretched configuration and study them under static [18] and dynamic loads [19]. Optimizing the robot topology for a given set of trajectories has also been proposed in [20].

This work introduces a novel approach to optimize the robot base and TCP placement, considering the entire process scenario, including grasp redundancy and free-space robot motions in a highly constrained environment. To this end, a P2P trajectory planner is incorporated into a highly parallelized optimization process to find time-optimal placements.

III. Problem Description

This paper aims to optimize the robot base placement, represented by the two-dimensional vector $\mathbf{b} = [b_\mathrm{x}\ b_\mathrm{y}]^\mathrm{T}$ describing the offset of the robot base from the world frame, and the TCP placement $\mathbf{t}$ for time optimality in a pick-and-place task in highly constrained environments. The task is specified by a sequence $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_L\}$, where odd indices l refer to pick operations, while place operations are indicated by even indices l, see Fig. 2. These operations are

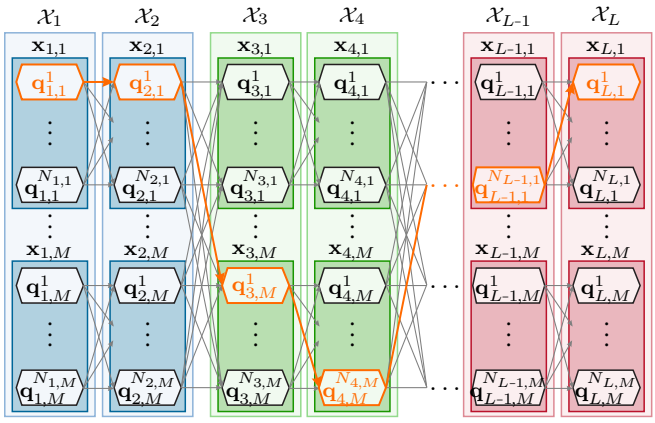

Fig. 2. Graph representation of the pick-and-place sequence $\mathcal{X}$ with the sets of redundant grasp poses $\mathcal{X}_l = \{\mathbf{x}_{l,m} \,|\, m = 1, \ldots, M\}$, $l = 1, \ldots, L$. Sets with odd index l contain pick poses; an even index l refers to place poses. For each pose $\mathbf{x}_{l,m}$, a set of IK solutions $\mathcal{Q}_{l,m} = \{\mathbf{q}^n_{l,m} \,|\, n = 1, \ldots, N_{l,m}\}$ is associated. An example solution is highlighted in orange.

planned and executed in the joint space by $L-1$ consecutive free-space P2P movements, where $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$, and $\ddot{\mathbf{q}}(t)$ are the position, velocity, and acceleration of the robot's joints, respectively.

Objects can be picked using different grasp poses and must be placed appropriately using a matching release pose at the target location. In this work, this *grasp redundancy* is considered by specifying each pick and place operation as a set of poses $\mathcal{X}_l = \{\mathbf{x}_{l,m} \,|\, m = 1, \ldots, M\}$, from which the most suitable one is chosen. Specifically, each pick pose $\mathbf{x}_{l,m}$ with odd l has a corresponding release pose $\mathbf{x}_{l+1,m}$ with the same index $m \in \{1, \ldots, M\}$. Conversely, after placing the object at the release pose $\mathbf{x}_{l+1,m}$, the robot can approach the next pick pose $\mathbf{x}_{l+2,m'}$ with any index $m' \in \{1, \ldots, M\}$. These task-specific constraints are illustrated in Fig. 2 as gray connecting lines.

For each pose $\mathbf{x}_{l,m}$, $m = 1, \ldots, M$, contained in $\mathcal{X}_l$, $l = 1, \ldots, L$, a set of inverse kinematics (IK) solutions $\mathcal{Q}_{l,m} = \{\mathbf{q}^n_{l,m} \,|\, n = 1, \ldots, N_{l,m}\}$ is obtained, where $N_{l,m}$ is the number of IK solutions for the pose $\mathbf{x}_{l,m}$. Additionally, a cyclic task execution is ensured by the constraint $\mathbf{q}_1 = \mathbf{q}_L$.

Finally, a 3D-printed part for the robot flange is designed using the optimal TCP placement $\mathbf{t}^*$. The 3D model is generated using topological optimization, considering the dynamic force profile at the TCP obtained from the dynamic model and the optimal robot trajectory $\mathbf{q}^*(t)$.

IV. TCP Generation

A geometric shape and a collision mesh are generated for each candidate TCP placement $\mathbf{t}$ during optimization. A placement $\mathbf{t} = [\mathbf{c}^\mathrm{T}\ \mathbf{d}^\mathrm{T}\ r]^\mathrm{T}$ is parametrized by seven scalar values with the 3D points $\mathbf{c} = [c_\mathrm{x}\ c_\mathrm{y}\ c_\mathrm{z}]^\mathrm{T}$ and $\mathbf{d} = [d_\mathrm{x}\ d_\mathrm{y}\ d_\mathrm{z}]^\mathrm{T}$, and the rotation r, see Fig. 3. From these parameters, the geometric shape is given as a cubic B-spline curve $\mathbf{u}(s)$, $0 \leq s \leq 1$, as

$$\mathbf{u}(s) = (1-s)^3\mathbf{p}_0 + 3(1-s)^2 s\mathbf{p}_1 + 3(1-s)s^2\mathbf{p}_2 + s^3\mathbf{p}_3\ ,$$

with the four control points $\mathbf{p}_0 = \mathbf{0}$, $\mathbf{p}_1 = [0\ 0\ 2\,\mathrm{cm}]^\mathrm{T}$, $\mathbf{p}_2 = \mathbf{c}$ and $\mathbf{p}_3 = \mathbf{d}$. The z-offset of $\mathbf{p}_1$ is chosen to ensure

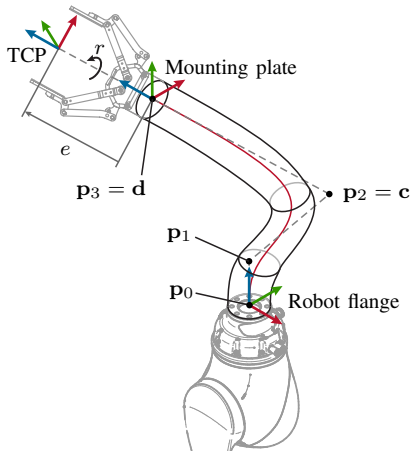

Fig. 3. Schematic drawing of the geometric shape and the collision mesh of a candidate TCP placement with the parameters $\mathbf{t} = [\mathbf{c}^{\mathrm{T}}\ \mathbf{d}^{\mathrm{T}}\ r]^{\mathrm{T}}$ and the control points $\mathbf{p}_0 = \mathbf{0}$ and $\mathbf{p}_1 = [0\ 0\ 2\,\mathrm{cm}]^{\mathrm{T}}$ given in the flange frame.

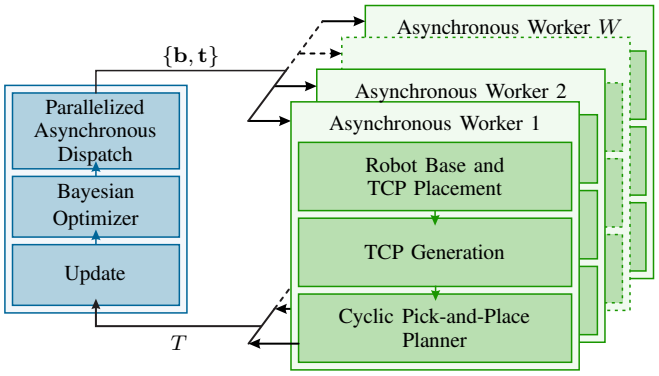

Fig. 4. Schematic drawing of the proposed optimization scheme. A Bayesian optimizer asynchronously dispatches robot base and TCP placements $\{\mathbf{b}, \mathbf{t}\}$ to be evaluated by W asynchronous workers in parallel. Each worker generates the TCP and the robot trajectories using the cyclic pick-and-place planner and evaluates the minimum cycle time T. The results are returned asynchronously to the Bayesian optimizer.

that the object can be mounted on the robot flance and to prevent interference between the last axis and the TCP. The TCP frame for the parameters $\mathbf{t}$ is obtained as the Frenet-Serret frame [21] at $\mathbf{u}(1) = \mathbf{p}_3$, translated along the local z-axis by the distance e and rotated around the local z-axis by the angle r, see Fig. 3. The collision mesh is generated as a tube with a fixed radius along the B-spline. For use in Mujoco [22], this mesh is decomposed into five convex meshes using V-HACD [23].

V. Placement Optimization

In this section, the proposed optimization scheme for the robot base and TCP placement is described in detail.

The Bayesian optimizer from [24] works at its core, which asynchronously dispatches candidate robot base placements $\mathbf{b}$ and TCP placements $\mathbf{t}$ to the W workers, see Fig. 4. These workers generate the TCP, see Section IV, and evaluate the total cycle time using the cyclic pick-and-place planner, which employs a P2P trajectory planner. Both planners are explained later in this section. This way, the surrogate model of the cycle time is iteratively updated, which enables the Bayesian optimizer to find the global minimum $\{\mathbf{b}^*, \mathbf{t}^*\}$ efficiently. Finally, a topology optimization approach is proposed in this section, allowing for the realization of the optimal TCP placement $\mathbf{t}^*$.

Algorithm 1 P2P planner: Given the robot base placement $\mathbf{b}$ and the TCP placement $\mathbf{t}$, a database $\mathcal{D}$ of all previously evaluated trajectories, compute the minimum time $T_{\mathbf{q}_1,\mathbf{q}_2}$ for a trajectory from the start configuration $\mathbf{q}_1$ to the end configuration $\mathbf{q}_2$.

1: **procedure** P2P-Planner($\mathbf{q}_1, \mathbf{q}_2, \mathbf{b}, \mathbf{t}, \mathcal{D}$)
2: **if** isInvalid($\mathbf{q}_1$,$\mathbf{b}$,$\mathbf{t}$) $\vee$ isInvalid($\mathbf{q}_2$,$\mathbf{b}$,$\mathbf{t}$) **then**
3: **return** *inf*
4: **end if**
5: **if** existsSimilar($\mathbf{q}_1$,$\mathbf{q}_2$,$\mathbf{b}$,$\mathbf{t}$,$\mathcal{D}$) **then**
6: VP-STO.init$\leftarrow \mathcal{D}(\mathbf{q}_{1,\mathcal{D}}$,$\mathbf{q}_{2,\mathcal{D}}$,$\mathbf{b}_{\mathcal{D}}$,$\mathbf{t}_{\mathcal{D}})$
7: **end if**
8: $T_{\mathbf{q}_1,\mathbf{q}_2} \leftarrow$ VP-STO($\mathbf{q}_1$,$\mathbf{q}_2$,$\mathbf{b}$,$\mathbf{t}$) ▷ modified version
9: **return** $T_{\mathbf{q}_1,\mathbf{q}_2}$
10: **end procedure**

A. Point-to-Point Trajectory Planner

The Via-Point Stochastic Optimization (VP-STO) planner is applied to optimize trajectories in highly constrained environments with a focus on time optimality [25]. This planner utilizes a low-dimensional, time-continuous trajectory representation in joint space based on via points and systematically considers the kinodynamic constraints of the robotic system. Furthermore, VP-STO employs stochastic optimization techniques to refine the via points and uses an evolutionary strategy to minimize the trajectory duration.

The original algorithm was adapted to be incorporated into the robot base and TCP placement optimization in this work. In order to mitigate the risk of converging to local minima, the initial exploration phases use a $100\times$ larger population, facilitating broad exploration and increasing the likelihood of identifying the global optimum. After the initial 20 evolutionary steps, the population size is reduced to retain only the 20 most promising candidates. These adjustments serve to reduce the algorithm runtime and enhance convergence speed. Additionally, a warm start approach using a trajectory database $\mathcal{D}$ is implemented, leveraging prior solutions for similar robot base ($\mathbf{b}_{\mathcal{D}}$), TCP placements ($\mathbf{t}_{\mathcal{D}}$) and the respective IK solutions ($\mathbf{q}_{1,\mathcal{D}}$, $\mathbf{q}_{2,\mathcal{D}}$). Initially, the database $\mathcal{D}$ is empty and each solution trajectory computed during the optimization is added to it. To maximize the effectiveness and keep the query time low, the database size is limited and each entry is assigned a time-to-live. The validation of trajectories in terms of collision avoidance is conducted with the Mujoco Mjx physics engine [22].

Algorithm 1 details the utilization of VP-STO to plan a single P2P trajectory, transitioning from the initial configuration $\mathbf{q}_1$ to the target configuration $\mathbf{q}_2$, for a candidate placement of the robot base $\mathbf{b}$ and TCP $\mathbf{t}$. Initially, it is ensured that both configurations $\mathbf{q}_1$ and $\mathbf{q}_2$ are collision-free for the candidate placements $\mathbf{b}$ and $\mathbf{t}$ (lines 2-4). Subsequently, the trajectory database $\mathcal{D}$ (line 5) encompassing all

Algorithm 2 Cyclic Pick-and-Place Planner: Given the task sequence $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_L\}$ and a database $\mathcal{D}$, compute the minimum total time T to traverse all sets in $\mathcal{X}$, considering the grasp redundancy and the cyclic constraint.

1: **procedure** CYCLIC-PICK-AND-PLACE-PLANNER($\mathcal{X}$,$\mathcal{D}$)
2: $V \leftarrow \{\}$ ▷ Initialize set of vertices
3: $E \leftarrow \{\}$ ▷ Initialize set of edges
4: **for** $l = 1$ **to** L **do** ▷ Compute all IK solutions
5: **for** $m = 1$ **to** M **do**
6: $\mathcal{Q}_{l,m} = \{\mathbf{q}_{l,m}^1, ..., \mathbf{q}_{l,m}^{N_{l,m}}\} \leftarrow \text{IK}(\mathbf{x}_{l,m})$
7: **end for**
8: **end for**
9: **for** $l = 1$ **to** $L-1$ **do** ▷ Compute all costs
10: **for** $m = 1$ **to** M **do**
11: **for all** $\mathbf{q}_1$ **in** $\mathcal{Q}_{l,m}$ **do**
12: V.append($\mathbf{q}_1$)
13: **if** l is odd **then**
14: $\mathcal{Q}' \leftarrow \mathcal{Q}_{l+1,m}$
15: **else**
16: $\mathcal{Q}' \leftarrow \bigcup_{m'=1}^{M} \mathcal{Q}_{l+1,m'}$
17: **end if**
18: **for all** $\mathbf{q}_2$ **in** $\mathcal{Q}'$ **do**
19: V.append($\mathbf{q}_2$)
20: $T_{\mathbf{q}_1,\mathbf{q}_2} \leftarrow$ P2P-Planner($\mathbf{q}_1, \mathbf{q}_2$,$\mathbf{b}$,$\mathbf{t}$,$\mathcal{D}$)
21: E.add($\mathbf{q}_1, \mathbf{q}_2, T_{\mathbf{q}_1,\mathbf{q}_2}$)
22: $\mathcal{D}$.add($\mathbf{q}_1, \mathbf{q}_2, \mathbf{b}, \mathbf{t}, T_{\mathbf{q}_1,\mathbf{q}_2}$)
23: **end for**
24: **end for**
25: **end for**
26: **end for**
27: $G \leftarrow (V, E)$ ▷ Construct the graph
28: $T \leftarrow$ Dijkstra(G) ▷ Considering cyclic constraint
29: **return** T
30: **end procedure**

previously found trajectories is queried to find similar trajectories satisfying the condition $d(\mathbf{q}_1, \mathbf{q}_{1,\mathcal{D}}) + d(\mathbf{q}_2, \mathbf{q}_{2,\mathcal{D}}) < \varepsilon$, where $d(\cdot,\cdot)$ denotes the Euclidean distance function, and ε is a threshold. Upon finding a matching trajectory, its via points are incorporated into the initial population of the VP-STO (line 6), and the exploration parameter σ is reduced to accelerate convergence towards a novel valid solution within just a few iterations.

B. Cyclic Pick-and-Place Planner

The cyclic pick-and-place planner finds the time-optimal trajectory for a complete task $\mathcal{X}$, considering the grasp redundancy and the cyclic constraint, see Fig. 2 and Section III.

Algorithm 2 summarizes the cyclic pick-and-place planner and is described in the following. First, the vertex set V and edge set E are initialized (lines 2-3). Then, all IK solutions $\mathcal{Q}_{l,m}$ are obtained for all considered grasp and release poses $\mathbf{x}_{l,m}$, $l = 1, \ldots, L$, $m = 1, \ldots, M$ (lines 4-8). Next, in lines 9 to 26, all relevant P2P robot motions are planned using Algorithm 1, and the graph elements are added to V and E. Thus, each node $\mathbf{q}_{l,m}^n$, $l = 1, \ldots, L$, $m = 1, \ldots, M$, $n = 1, \ldots, N_{l,m}$, represents a robot configuration to reach the corresponding pose $\mathbf{x}_{l,m}$ in the task space. Each edge indicates the transition time $T_{\mathbf{q}_1,\mathbf{q}_2}$ between two configurations $\mathbf{q}_1$ and $\mathbf{q}_2$. Note that all edge costs can be computed independently and in parallel. According to the grasp redundancy constraint, the robot can only move from a pick pose $\mathbf{x}_{l,m}$ with odd l to a compatible place pose $\mathbf{x}_{l+1,m}$ with the same index $m \in \{1, \ldots, M\}$, see Fig. 2. Finally, the graph is constructed from the vertex set V and the edge set E, for which the time-optimal trajectory for task $\mathcal{X}$ is found using Dijkstra's algorithm [26] (lines 27-28). This graph search algorithm is extended to consider the cyclic constraint $\mathbf{q}_1 = \mathbf{q}_N$. Generating the complete graph of Fig. 2 is time consuming and can be avoided by employing heuristic search strategies [27].

C. Task-Specific Topology Optimization

The proposed optimization scheme, illustrated in Fig. 4, yields the time-optimal placements $\{\mathbf{b}^*, \mathbf{t}^*\}$ for the robot base and the TCP, respectively. Additionally, the optimal trajectory $\mathbf{q}^*(t)$, the joint velocity $\dot{\mathbf{q}}^*(t)$ and acceleration $\ddot{\mathbf{q}}^*(t)$ are provided by the VP-STO planner. Given the kinematic model of the robot with the geometric Jacobian $\mathbf{J}(\mathbf{q})$, the Cartesian accelerations of the TCP $\ddot{\mathbf{x}}_\text{t}$ are computed as

$$\ddot{\mathbf{x}}_\text{t}(t) = \mathbf{J}(\mathbf{q}^*(t))\ddot{\mathbf{q}}^*(t) + \dot{\mathbf{J}}(\mathbf{q}^*(t))\dot{\mathbf{q}}^*(t) \ ,$$

from which the acting forces at the TCP are calculated as force vector $\mathbf{f}(t)$ using the load mass and the center of mass, considering the distinct phases with and without grasped object. Next, the maximum force along a set of discrete directions is obtained, see Fig. 5. Finally, the topology optimization algorithm of Autodesk Fusion 360 [28] is used to generate a 3D model for the TCP placement $\mathbf{t}^*$, which considers a user-defined material, is constrained geometrically within the tube-shaped mesh shown in Fig. 3, and complies with the force profile $\mathbf{f}(t)$ with a suitable mechanical safety factor. This way, the new TCP placement is realized experimentally as a lightweight 3D-printable part, which withstands the dynamic loads at the TCP and ensures a collision-free trajectory.

VI. Simulation Results

This section investigates a highly constrained pick-and-place task in simulation, in which the time-optimal robot base and TCP placement are computed using the optimization scheme from Section V.

First, the scenario is described in detail. Then, the placement optimization is conducted two times: (a) Both placements $\{\mathbf{b}, \mathbf{t}\}$ are optimized simultaneously, (b) the robot base placement is optimized for a *trivial* TCP, i.e., for a gripper mounted directly at the robot flange without any positional offset, to underscore the significance of TCP placement with respect to cycle time. These results are compared to placements based on performance indices, and finally, the algorithm runtime is discussed.

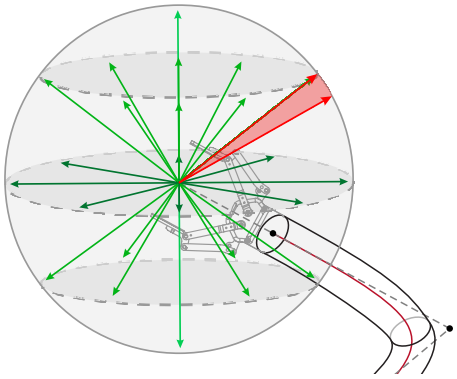

Fig. 5. Discrete force directions for the estimation of the maximum forces at the TCP.

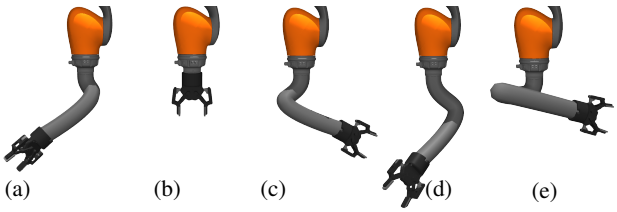

Fig. 6. Visualization of the TCP placements with collision meshes (gray tubes) for the results in Table I. (a) Robot base and TCP optimization, (b) Robot base optimization with gripper mounted directly on the robot flange (trivial TCP placement) (c) Manipulability, (d) Cartesian-distance, (e) Joint-space-distance.

A. Scenario

The considered scenario is a robotic unloading process from a manufacturing station into blister packaging. A sequence of $L = 8$ pick-and-place poses $\mathcal{X}$ is planned, with four sheet metal products being packaged. Grasping these sheet metal products has a two-fold grasping redundancy ($M = 2$), as the parallel jaw gripper can be rotated by $180°$ to achieve an alternative grasp. This results in two possible ways to pick up the product and place it into the blister. After placing the product, the robot can return to the manufacturing station due faster due to the empty gripper.

The robot considered in this scenario is the collaborative robot KUKA LBR iiwa 14 R820 with 7 degrees of freedom (DoF), for which the IK can be solved analytically. Different kinematic solutions are considered, i.e., elbow left/right and wrist left/right [29]. Additionally, an infinite number of IK solutions exist for a given pose, as the elbow of the robot can continuously rotate around the straight line between the shoulder and the wrist point. This continuous space is sampled equidistantly with 12 points.

TABLE I
OPTIMIZATION RESULTS

Method	Minimum cycle time
Robot base and TCP optimization	17.58 s
Robot base optimization with trivial TCP	25.64 s
Manipulability	29.61 s
Cartesian-distance	infeasible
Joint-space-distance	22.98 s

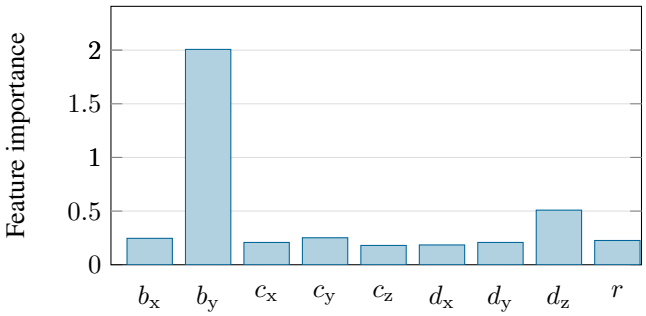

Fig. 7. Feature importance of the optimization variables $\mathbf{b}^T = [b_x\ b_y]$ and $\mathbf{t}^T = [\mathbf{c}^T\ \mathbf{d}^T\ r]$, estimated using SHAP [30].

B. Robot Base and TCP Placement

First, the simultaneous optimization of the robot base and the TCP placement using the proposed scheme from Section V is conducted. The optimal placement $\{\mathbf{b}^*, \mathbf{t}^*\}$ convergences to a cycle time of 17.58 s; see Table I. The resulting TCP placement and its collision mesh are shown in Fig. 6a.

Figure 7 shows the parameter importance for the optimization, which is estimated using *SHapley Additive exPlanations* (SHAP) [30]. It shows that the robot base placement $\mathbf{b}$ has a significantly higher impact on the cycle time with a parameter importance of 2.01, while the TCP parameters $\mathbf{t}$, described in Section IV, are between 0.18 and 0.51. The highest importance for the optimal cycle time among the TCP parameters is found for d_z, which determines how protruding the TCP with respect to the robot flange is.

C. Robot Base Placement with trivial TCP

The robot base placement optimization is executed for the trivial TCP placement, as illustrated in Fig. 6b. Hence, the gripper is directly mounted on the robot flange and the TCP placement is fixed with $\mathbf{c} = [0\ 0\ 0]^T$, $\mathbf{d} = [0\ 0\ 0.05\,\text{cm}]^T$, and $r = 0$. The optimal robot base placement results in a minimum cycle time of 25.64 s, which corresponds to a 46 % increase in cycle time compared to the optimal placement of the robot base and TCP. This showcases the potential of an optimally designed TCP.

D. Comparison to the State of the Art

This section compares the proposed optimization scheme to the state of the art. Specifically, the performance indices *Manipulability* [1], [31], *Cartesian distance* [8], and *Joint-space distance* [10] are used as trajectory cost instead of the actual duration $T_{\mathbf{q}_1,\mathbf{q}_2}$. This way, a graph analogous to Fig. 2 is built, which considers the task constraints and all feasible collision-free IK solutions. The minimum cumulative cost (for Joint-space distance, Cartesian-distance) or the maximum cumulative cost (for Manipulability) is evaluated using Dijkstra's algorithm [26].

The optimization results are listed in Table I. The optimal TCP and robot base placement maximizing the manipulability at all grasping positions yields a minimum cycle time of 29.61 s. Thus, the proposed approach achieves a 41 %

lower cycle time than the manipulability-based approach, cf. Section VI-B. The placement using the joint-space distance resulted in a minimum cycle time of 22.98 s. The optimization based on the Cartesian distance did not yield valid placements. Thus, minimizing a performance index for all pick-and-place poses does not translate to time-optimal placements.

E. Algorithm Runtime

The runtime of the proposed algorithm for determining the optimal placement of the robot base and the TCP in the scenarios described in Section VI-A is approximately 360 min on a computer with 90 cores. A single run of the P2P planner on a single core takes approximately 6 s without prior initialization from the trajectory database $\mathcal{D}$. Given proper close to optimal initialization from the database, the computation time drops on average to 1.21 s and reduces to 4.1 ms if either the start configuration $\mathbf{q}_1$ or the end configuration $\mathbf{q}_2$ are in collision with the environment, see Algorithm 1. It is worth noting that this runtime is significantly influenced by the underlying parameters of the P2P planner, the complexity of the environment, and the length of the task sequence $\mathcal{X}$.

In contrast, placement methods based on performance indices exhibit a significantly shorter runtime, typically in the order of a few minutes on the same hardware as above. However, as evidenced by the results listed in Table I, these methods deliver sub-optimal placements in terms of cycle time since the actual robot trajectory is not considered or cannot provide a feasible solution at all. In contrast, the proposed method by design guarantees a feasible placement with valid, collision-free robot trajectories at the robot's kinodynamic limits.

VII. Experimental Results

The optimal robot base and TCP placement $\{\mathbf{b}^*, \mathbf{t}^*\}$ from Section VI-B minimizes the total cycle time, considering task's grasp redundancy, the robot's kinodynamic limits and the highly constrained environment. In this section, the resulting placements for the simulation scenario are realized in the experimental setup and are then validated.

A. Task-Specific Topology Optimization

The topology optimization of Section V-C is applied to calculate the optimal TCP placement $\mathbf{t}^*$ to generate a 3D model for manufacturing using 3D printing. To this end, the Cartesian accelerations at the TCP $\ddot{\mathbf{x}}_\mathrm{t}(t)$ are computed using the robot's kinematic model, see Section V-C and Fig. 8. These results are validated by measurements using two IMUs mounted on the optimal TCP placement $\mathbf{t}^*$.

The topology optimization is performed after estimating the force profile given the gripper mass of 917 g and a payload mass of 337 g, as described in Section V-C. Two different materials are investigated, i.e., the well-known 3D printing material *polylactic acid* (PLA) [32] and the lightweight, high-strength aluminum alloy *AlSi10Mg* [33]. The resulting 3D models are designed to withstand the estimated forces and are contained in the tube-shaped collision mesh, guaranteeing

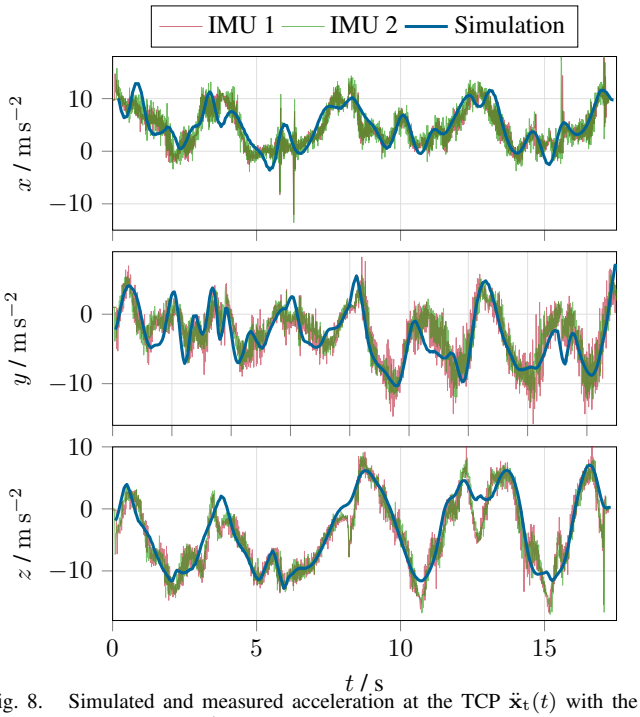

Fig. 8. Simulated and measured acceleration at the TCP $\ddot{\mathbf{x}}_\mathrm{t}(t)$ with the optimal TCP placement $\mathbf{t}^*$ from Section VI-B.

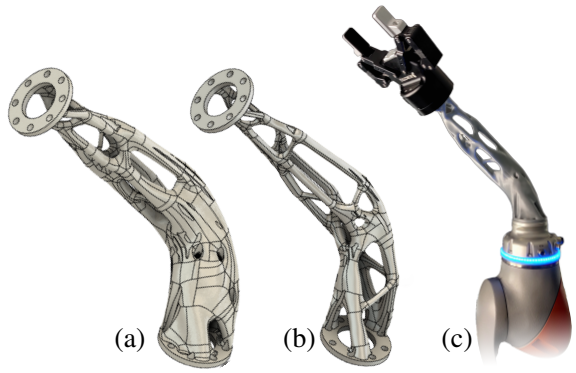

Fig. 9. Results of the topology optimization. (a) 3D model designed for PLA, (b) 3D model designed for AlSi10Mg, (c) 3D printed model from (a).

collision-free robot motions. The topology optimization uses a mechanical safety factor of 2 to minimize the TCP's mass compared to the solid collision mesh. This yields a mass reduction of 78 % for PLA and 88 % for AlSi10Mg, see Fig. 9. For the experimental verification, the 3D model is printed using PLA and mounted on the robot flange, see Fig. 9c.

B. Experimental Verification

In the experimental setup, see Fig. 1, the robot base and TCP placement are realized experimentally using a 3D-printed part. The time-optimal robot trajectory $\mathbf{q}^*(t)$ was executed, and the sequence of pick-and-place tasks $\mathcal{X}$ was performed successfully. A video of the experimental validation is found at www.acin.tuwien.ac.at/3tcp. The accelerations at the TCP were measured using two independent inertial measurement units (IMU) to obtain $\ddot{\mathbf{x}}_\mathrm{t}(t)$. The results, Fig. 8, show a good agreement between the measurement and the estimated accelerations from the previous section. This

validation highlights the effectiveness of the proposed robot base and TCP optimization and the resulting trajectories in real-world scenarios.

VIII. Conclusions

The algorithm proposed in this work optimizes the robot base and tool center point (TCP) placement for pick-and-place tasks to minimize the cycle time.

The large search space is explored rapidly and efficiently utilizing an asynchronous, highly parallelized Bayesian optimizer with a cyclic pick-and-place trajectory planner. Additionally, the proposed method considers the scenario holistically in the optimization by incorporating a point-to-point (P2P) trajectory planner with the kinodynamic constraints of the robot, collision avoidance, and redundancy in grasp configurations and robot kinematics. The resulting TCP placement is realized experimentally using topology optimization and 3D printing. The 3D-printed part is designed to be constrained geometrically within the tube-shaped mesh and to withstand the dynamic force profile present at the TCP due to the masses of the gripper and the grasped object.

The proposed method shows a significant cycle time reduction compared to state-of-the-art methods like manipulability, joint-space distance, and Cartesian distance, highlighting its potential for real-world use in robotics cell design and optimization. Future work focuses on the optimization of complete layouts of robotic work cells.

References

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

[2] A. Makhal and A. K. Goins, "Reuleaux: Robot base placement by reachability analysis," in *International Conference on Robotic Computing*, 2018, pp. 137–142.

[3] B. Kamrani, V. Berbyuk, D. Wäppling, U. Stickelmann, and X. Feng, "Optimal robot placement using response surface method," *The International Journal of Advanced Manufacturing Technology*, vol. 44, pp. 201–210, 2009.

[4] C. Valsamos, K. Miatliuk, A. Wolniakowski, V. Moulianitis, and N. Aspragathos, "Optimal kinematic task position determination – application and experimental verification for the UR-5 manipulator," *Applied Sciences*, vol. 12, no. 18, p. 9352, 2022.

[5] S. Patel and T. Sobh, "Manipulator performance measures-a comprehensive literature survey," *Journal of Intelligent & Robotic Systems*, vol. 77, pp. 547–570, 2015.

[6] H. Asada, "A geometrical representation of manipulator dynamics and its application to arm design," *Journal of Dynamic Systems, Measurement, and Control*, vol. 105, no. 3, pp. 131–142, 1983.

[7] J. K. Salisbury and J. J. Craig, "Articulated hands: Force control and kinematic issues," *The International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982.

[8] T. Bachmann, K. Nottensteiner, and M. A. Roa, "Automated planning of workcell layouts considering task sequences," in *Proc. IEEE International Conference on Robotics and Automation*, 2021, pp. 12 662–12 668.

[9] T. Weingartshofer, C. Hartl-Nesic, and A. Kugi, "Optimal TCP and robot base placement for a set of complex continuous paths," in *Proc. IEEE International Conference on Robotics and Automation*, 2021, pp. 9659–9665.

[10] K. Baizid, A. Yousnadj, A. Meddahi, R. Chellali, and J. Iqbal, "Time scheduling and optimization of industrial robotized tasks based on genetic algorithms," *Robotics and Computer-Integrated Manufacturing*, vol. 34, pp. 140–150, 2015.

[11] D. Spensieri, J. S. Carlson, R. Bohlin, J. Kressin, and J. Shi, "Optimal robot placement for tasks execution," in *Proc. Conference on Assembly Technologies and Systems*, vol. 44, 2016, pp. 395–400.

[12] J. Beck, "Towards an automated system for robot assembly cell layout optimization," in *Proc. Stuttgart Conference on Automotive Production*. Springer, 2022, pp. 108–117.

[13] S. Zimmermann, G. Hakimifard, M. Zamora, R. Poranne, and S. Coros, "A multi-level optimization framework for simultaneous grasping and motion planning," *Robotics and Automation Letters*, vol. 5, no. 2, pp. 2966–2972, 2020.

[14] K. Hang, M. Li, J. A. Stork, Y. Bekiroglu, F. T. Pokorny, A. Billard, and D. Kragic, "Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 960–972, 2016.

[15] J. Ichnowski, M. Danielczuk, J. Xu, V. Satish, and K. Goldberg, "Gomp: Grasp-optimized motion planning for bin picking," in *Proc. IEEE International Conference on Robotics and Automation*, 2020, pp. 5270–5277.

[16] J. Shao, J. Liao, S. Zhu, H. Zhang, and W. Song, "Trajectory optimization for manipulation considering grasp selection and adjustment," *Journal of Intelligent & Robotic Systems*, vol. 109, no. 1, p. 12, 2023.

[17] G. L. Srinivas and A. Javed, "Optimization approaches of industrial serial manipulators to improve energy efficiency: A review," in *IOP Conference Series: Materials Science and Engineering*, vol. 912, no. 3, 2020, p. 032058.

[18] I. Chavdarov, B. Naydenov, K. Yovchev, and L. Miteva, "Topology optimization of an assembled 3D printed robot," in *Proc. International Conference on Software, Telecommunications and Computer Networks*, 2022, pp. 1–6.

[19] G. L. Srinivas and A. Javed, "Topology optimization of industrial manipulator-link considering dynamic loading," in *Materials Today: Proceedings*, vol. 18, no. 7, 2019, pp. 3717–3725.

[20] S. Briot and A. Goldsztejn, "Topology optimization of industrial robots: Application to a five-bar mechanism," *Mechanism and Machine Theory*, vol. 120, pp. 30–56, 2018.

[21] M. G. Wagner and B. Ravani, "Curves with rational Frenet-Serret motion," *Computer Aided Geometric Design*, vol. 15, no. 1, pp. 79–101, 1997.

[22] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[23] K. Mamou, E. Lengyel, and A. Peters, "Volumetric hierarchical approximate convex decomposition," in *Game Engine Gems 3*, 2016, pp. 141–158.

[24] Y. Li, Y. Shen, W. Zhang, Y. Chen, H. Jiang, M. Liu, J. Jiang, J. Gao, W. Wu, Z. Yang, C. Zhang, and B. Cui, "OpenBox: A generalized black-box optimization service," in *Proc. ACM Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3209–3219.

[25] J. Jankowski, L. Brudermüller, N. Hawes, and S. Calinon, "VP-STO: Via-point-based stochastic trajectory optimization for reactive robot behavior," in *Proc. IEEE International Conference on Robotics and Automation*, 2023, pp. 10 125–10 131.

[26] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[27] B. Bonet and H. Geffner, "Planning as heuristic search," *Artificial Intelligence*, vol. 129, no. 1-2, pp. 5–33, 2001.

[28] P. Song, Y. Qi, and D. Cai, "Research and application of Autodesk Fusion360 in industrial design," in *IOP Conference Series: Materials Science and Engineering*, vol. 359, no. 1, 2018, p. 012037.

[29] I. Kuhlemann, A. Schweikard, P. Jauer, and F. Ernst, "Robust inverse kinematics by configuration control for redundant manipulators with seven DoF," in *Proc. IEEE International Conference on Control, Automation and Robotics*, 2016, pp. 49–55.

[30] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, vol. 30, pp. 4768–4777, 2017.

[31] K. Moriguchi, T. Mizokami, and K. Morishige, "Generation of a robot program and determination of an optimal workpiece placement considering the manipulability of industrial robots," *International Journal of Automation Technology*, vol. 16, no. 6, pp. 870–878, 2022.

[32] S. Farah, D. G. Anderson, and R. Langer, "Physical and mechanical properties of PLA, and their functions in widespread applications – A comprehensive review," *Advanced Drug Delivery Reviews*, vol. 107, pp. 367–392, 2016.

[33] K. Kempen, L. Thijs, J. Van Humbeeck, and J.-P. Kruth, "Mechanical properties of AlSi10Mg produced by selective laser melting," *Physics Procedia*, vol. 39, pp. 439–446, 2012.