Optimization of robot configurations for motion planning in industrial riveting

Hakan Girgin[1,2], Teguh Santoso Lembono[1,2], Radu Cirligeanu[3] and Sylvain Calinon[1,2]

***Abstract*— Collaborative robots (cobots) have been increasingly used in the industry in recent years. The cobots are often 7-axis kinematically redundant manipulators. Exploiting these redundancies in industrial tasks is still a challenge as the change in the robot joint configuration can be unnecessarily large from one task to another depending on the initialization of the numerical inverse kinematics. In this article, we address this challenge in a particularly important under-explored industrial task: percussive riveting. The load exerted on the rivet for its plastic deformation can be dangerous for the robot if the natural frequency of the robot coincides with the frequency of the impact forces. In this work, we propose to take into account the vibrational characteristics of the manipulator to determine the robot joint configurations that minimize the overall end-effector's displacement. We also propose to exploit these safe joint configurations for each rivet hole in a task and motion planning algorithm called RoboTSP to generate the optimal trajectory visiting all target holes. We demonstrate our method with a Franka Emika Panda robot in a simulated environment.**

I. Introduction

Riveting is the mechanical process of joining metallic (or composite) parts together by means of an additional metallic part called rivet, serving to join the parts through adjacent surfaces. In principle, there are two riveting methods depending on the type of forces applied on the rivets: squeezing (or one-shot) and percussive (or hammering). The squeezing method can only be performed in an automatized way as it requires a large upsetting force to be applied to deform the rivet instantly on large components. On the other hand, the percussive method is usually performed manually with a smaller impulsive force applied to deform a rivet cumulatively by a series of hits.

The manual percussive riveting typically requires two people. One person uses an air hammer (pneumatic riveting hammer) to deform the rivet (the operator Op.1 in Fig. 1), while the other uses a metal bar, called a bucking bar pressed against the rivet on the opposite side of the assembly in order to hold it in place and counteract the force (the operator Op.2 in Fig. 1). The regions where this counteractive force needs to be applied are difficult to reach. As a consequence, the workers who perform this job need to have strong and long arms. They need to work for long periods of time in uncomfortable poses and are subject to vibrations induced by the pneumatic hammer that can cause issues in ergonomic, health and safety of the workers. The assembly process thus becomes time-inefficient and generates high physical stress on the workers performing it. Another disadvantage of the manual riveting is that for an aircraft, there are many structures to be riveted. Therefore the development of adaptive robot controllers and planners for cobots helping in the riveting process has a huge potential.

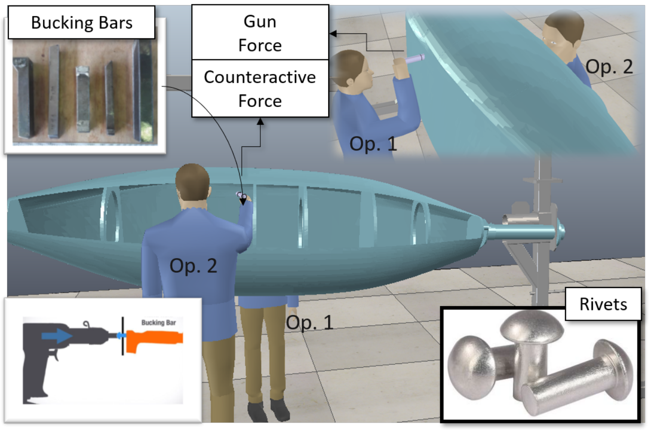


Fig. 1: Simulated rendering of the setup in the aircraft construction facility. The assembly structure to be riveted by two operators (the photo of the real setup cannot be shown due to confidentiality). The operator Op.1 applies the forces with the pneumatic gun while the other operator Op.2 applies a counteractive force on the rivet hole.

As the riveting is a kinematically redundant task even for 6-axis robots as explained in Section IV, there are infinitely many solutions to reach these rivets. In this article, we exploit the kinematic redundancy of the proposed collaborative robotic riveting system in the vibrational response when the robot end-effector is subject to percussive forces, and also in the planning of the whole task. The structure to test the proposed robotic solution is shown in Fig. 2.

Our contributions are as follows. First, we propose to combine the vibrational characteristics of a 7-axis collaborative robot, Franka Emika Panda for the percussive force loading, with its kinematic redundancy. We investigate how to determine joint configurations that minimize the displacement of the end-effector during the riveting process for the robot safety and the good quality riveting. Second, we propose to exploit these well-behaved joint configurations in a task and motion planning algorithm to estimate the optimal path in

[1] Idiap Research Institute, Martigny, Switzerland (e-mail: firstname.lastname@idiap.ch)

[2] Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

[3] ROMAERO SA, Bucharest, Romania (e-mail: radu.cirligeanu@romaero.com)

*Supported by the European Union H2020 CoLLaboratE project (https://collaborate-project.eu/) under grant agreement 820767 and by the H2020 MEMMO project (Memory of Motion, http://www.memmo-project.eu/), under grant agreement 780684.

the configuration space given the rivet hole sequence.

The paper is organized as follows. Section II gives related work on vibration analysis and motion planning for this kind of applications. Section III gives a background on the vibration analysis of flexible joint robots. In Section IV, we first present the detailed methodology of the optimization procedure to determine vibration resistant configurations and then explain how these will be incorporated and exploited in an optimal task and motion planning algorithm. We discuss the results of our proposed approach on the envisaged riveting structure using a Franka Emika Panda robot in Section V, and we conclude our work with Section VI.

II. Related Work

A. Vibration analysis for percussive riveting

The focus of the work presented in [1] is to develop a numerical analysis of the robot vibration during the percussive riveting for fatigue life analysis of the robot. In [2], the authors build on the previous work to find a preferable workspace for fatigue life improvement of robots with flexible joints subject to percussive loading. In [3], the authors propose to optimize the process parameters such as the end-effector pose and the force impact frequency in order to minimize the tool misalignment caused by percussive loading. The work in [5] proposes new displacement-force frequency response ratio indices to describe the vibration characteristics of flexible joint robots when subject to percussive loading. However, these works do not consider the redundancy in the robot kinematics as proposed by this work.

A similar work in [4] deals with the vibration analysis of a semi-autonomous robotic riveting system considering the end-effector as point mass. The authors find the natural frequency of the robot given a fixed configuration and design a controller to cope with the vibration. In contrast, our work copes with the previous step, i.e., to find good robot configurations to plan an optimal reaching the riveting holes.

Although applicable to a broader range of robots, [1], [2], [3], [5] present results only on a 3 degrees-of-freedom (DoF) robot that can hardly reach the desired riveting position with a single configuration. In contrast, our work focuses on the optimization of joint configurations of a 7-DoF robot by exploiting the redundancy of the robot with respect to the desired task. Besides extending to higher DoF systems, with the proposed optimization procedure and the planning strategy, our work is an extension of [5], as we optimize the equilibrium joint configuration to minimize the end-effector displacements at different rivet hole positions.

The robotic system proposed in these works also differ from the proposed solution in two major aspects. First, the collaborative system considered here is a semi-automatized platform envisaged as a collaborative solution with a human partner holding the gun and applying the percussive forces. This allows the solution to be more adaptable and flexible, which can be applied to more diverse structures and reprogrammed on the execution by the help of the operator. Second, we exploit the kinematic redundancies to minimize the end-effector displacement of the robot holding the bucking bar due to the vibrations involved in the riveting process, in order to avoid dangerous robot configurations.

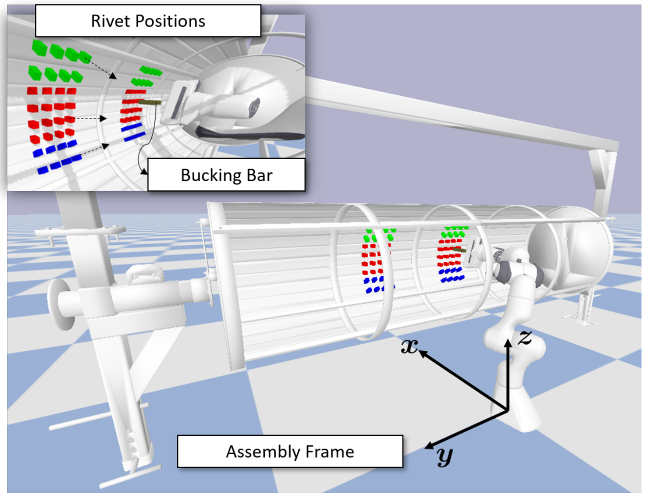


Fig. 2: Simplified riveting structure simulated in PyBullet, showing the Franka Emika Panda robot holding the bucking bar at a rivet hole, the placement of the rivet holes in this structure, and the chosen 64 rivet hole positions (represented by top (green), middle (red) and bottom (blue) blocks) used to showcase the capability of our approach.

B. Task and Motion Planning

Riveting tasks involve a large number of holes to be visited. At each hole, there can be multiple robot configurations that can satisfy the task (a redundant 7-DoF robot with 6-DoF task would have infinite number of possible configurations). Naively choosing the configurations and the hole sequence would result in long trajectories between one configuration and the next. This motivates the need for task and motion planning algorithms that can choose the visiting order of the holes, to select the optimal robot configuration at each hole, and to plan the trajectory between any two consecutive configurations. Many algorithms have been proposed, e.g., using group spanning trees ([6], [7]), genetic algorithms ([8], [9]), or multi objective constraint optimization [10], but most of them require substantial planning times when the number of tasks is large. RoboTSP is proposed in [11] as a way to solve the task and motion planning for a large number of holes in a drilling task. The computation time is much faster than the previously proposed approaches, while still maintaining the quality of the solutions.

III. Background

For the vibrational analysis of the robot arm subject to percussive forces, we follow the work in [5]. Classical dynamic modeling of robotic manipulators assumes that the links of the robot are rigid and the equations of motion of a n-DoF manipulator can be written as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}+\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_{\mathrm{m}}+\boldsymbol{\tau}_{\mathrm{ext}}, \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the generalized mass matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is coriolis and centrifugal forces matrix, $\boldsymbol{g}(\boldsymbol{q})$ is the gravity

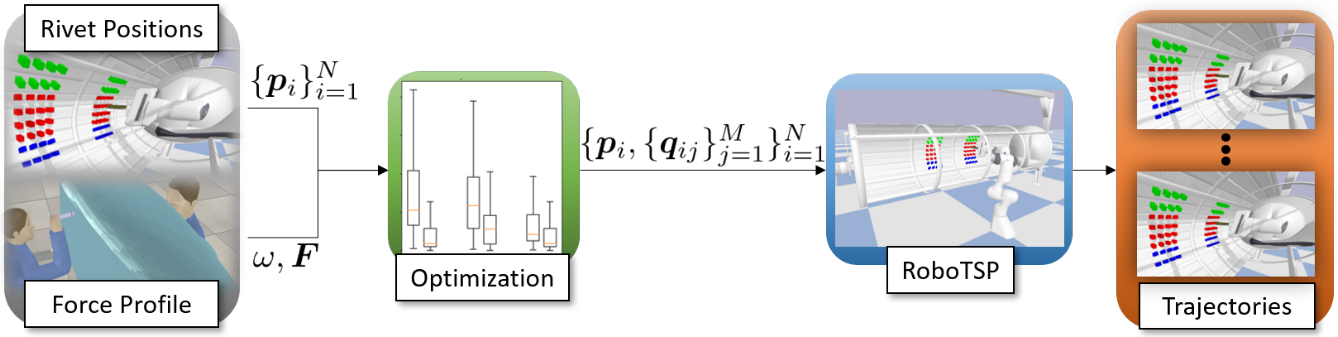

Rivet Positions
$\{\boldsymbol{p}_i\}_{i=1}^N$
$\omega, \boldsymbol{F}$
Force Profile
Optimization
$\{\boldsymbol{p}_i, \{\boldsymbol{q}_{ij}\}_{j=1}^M\}_{i=1}^N$
RoboTSP
Trajectories

Fig. 3: Overview of the proposed method. Given a set of target poses $\{\boldsymbol{p}_i\}_{i=1}^N$ and a force signal $\boldsymbol{F}$ with a frequency ω, we use optimization to find robot configurations that reach these poses while minimizing vibrations. For each pose $\boldsymbol{p}_i$, M robot configurations $\{\boldsymbol{q}_{ij}\}_{j=1}^M$ are found. RoboTSP determines the best configuration among the M choices for each hole, and finally plans the optimal joint angle trajectories visiting all these configurations.

vector, $\boldsymbol{\tau}_\text{m}$ is the robot torque commands and $\boldsymbol{\tau}_\text{ext}$ is the external torques applied to the robot. For riveting applications, the robot needs to reach a desired configuration $\boldsymbol{q}_d$ from which any deflection is undesired. In other words, the desired joint velocities and accelerations are zero, $\dot{\boldsymbol{q}}_d{=}\ddot{\boldsymbol{q}}_d{=}\boldsymbol{0}$. The controller to achieve this task is designed as

$$\boldsymbol{\tau}_\text{m} = \boldsymbol{K}_d(\boldsymbol{q}_d-\boldsymbol{q})+\boldsymbol{C}_d(\dot{\boldsymbol{q}}_d-\dot{\boldsymbol{q}})+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}+\boldsymbol{g}(\boldsymbol{q}), \tag{2}$$

where $\boldsymbol{K}_d$ and $\boldsymbol{C}_d$ are the desired stiffness and damping matrices, respectively. Letting $\boldsymbol{\delta}{=}\boldsymbol{q}-\boldsymbol{q}_d$ be the perturbations or deflections in joint angles from the desired configuration, one can express the forced vibration equation of the manipulator when it is subject to external percussive forces $\boldsymbol{F}(t)$, by substituting (2) into (1) as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{\delta}}+\boldsymbol{C}_d\dot{\boldsymbol{\delta}}+\boldsymbol{K}_d\boldsymbol{\delta} = \boldsymbol{J}^\top\boldsymbol{F}(t), \tag{3}$$

where $\boldsymbol{J}$ is the Jacobian matrix. We then perform the vibration analysis with a simple harmonic force $\boldsymbol{F}(t){=}\bar{\boldsymbol{F}}\mathrm{e}^{j\omega t}$ which results in the steady-state response of the joint deflections $\boldsymbol{\delta}(t) = \bar{\boldsymbol{\delta}}\mathrm{e}^{j\omega t}$ for the forced vibration problem in (3) where $\bar{\boldsymbol{\delta}}{=}(-\omega^2\boldsymbol{M}+j\omega\boldsymbol{C}+\boldsymbol{K})^{-1}\boldsymbol{J}^\top\bar{\boldsymbol{F}}$. The mass matrix and the Jacobian are evaluated at the equilibrium configuration assuming small deflections. Note that this analysis can easily be extended to other periodic forces such as square signal produced by the percussive riveting using the superposition principle and Fourier expansion of the force signal. Using $\bar{\boldsymbol{\delta}}$, we can express the end-effector displacement amplitude as

$$\Delta\boldsymbol{x} = \boldsymbol{J}^\top\bar{\boldsymbol{\delta}} = \boldsymbol{H}(\omega)\bar{\boldsymbol{F}}, \tag{4}$$

where $\boldsymbol{H}(\omega){=}\boldsymbol{J}^\top(-\omega^2\boldsymbol{M}+j\omega\boldsymbol{C}+\boldsymbol{K})^{-1}\boldsymbol{J}^\top$ is the *displacement-force frequency response*.

IV. Methods

In percussive riveting applications, periodic forces can cause large end-effector displacements if the natural frequency of the manipulator is close to the frequency of the force profile. The displacement amplitude defined in (4) can inform us on how large the end-effector displacements would be if such forces were to be applied. These displacements are undesirable as the accuracy of the end-effector position is important for a good quality riveting and very large displacements can break the robot or reduce its lifespan. Our work thus aims to minimize the end-effector displacements when the robot tool is subject to percussive forces.

The task to hold the bucking bar at the desired riveting hole requires accurate positioning in all 3 DoF in the Cartesian space and only 2 DoF for orientation (see Fig. 2). This one DoF allows for more than 6-axis robots to admit infinitely many configurations for a given hole, such as for Franka Emika Panda used in this work. We therefore would like to find the minimum-displacement robot configurations given each rivet hole pose. Even though finding one well-behaved joint configuration may be enough for just one rivet hole, the transitions from one hole to another may not be optimal if the motion planner is provided with only one configuration for each hole. We would like therefore to find an ensemble of configurations for each hole, where each configuration minimizes the end-effector displacements. Figure 3 shows the overview of the proposed approach.

A. *Optimal Joint Configurations*

Our problem is to find joint configurations that correspond to given 5-DoF rivet hole poses, but also that minimize the displacement amplitude $\Delta\boldsymbol{x}$ of the end-effector resulting from the vibrational forces. We cast this optimization problem as the minimization of several cost functions. Given a force frequency ω, the overall cost to optimize is therefore

$$c(\boldsymbol{q}) = c_\text{pos}(\boldsymbol{q})+c_\text{orn}(\boldsymbol{q})+\beta c_\text{disp}(\boldsymbol{q},\omega), \tag{5}$$

where β defines the weight of the cost on the displacement of the end-effector $c_\text{disp}(\boldsymbol{q},\omega)$ relative to the costs on the position of the end-effector $c_\text{pos}(\boldsymbol{q})$ and to the cost on the orientation of the end-effector $c_\text{orn}(\boldsymbol{q})$. We implemented this cost function in Tensorflow and used the Adam optimizer with its default parameters to determine M solutions for each of the 64 rivet holes. Note that we initialize the problem from random configurations that are within the joint limits. We first chose β to be a very small number such as 10^{-5} and gradually increased it such that we do not see big deflections

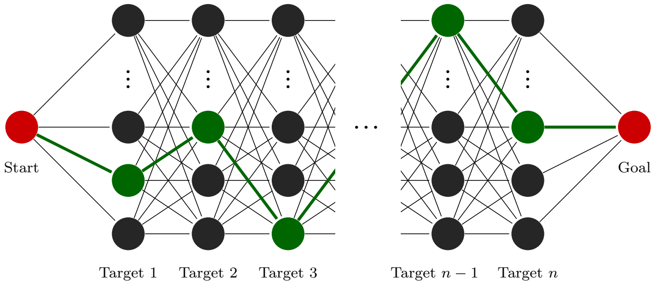


Fig. 4: Determining the sequence of optimal configurations using RoboTSP [11]. Each node in the graph is associated to a robot configuration at a particular target/hole. A Dijkstra algorithm is then used to compute the optimal path that minimizes the total weight of the traversed edges.

for the cost on the position and the orientation. Such an approach is standard in Lagrangian optimization methods.

Cost on the end-effector pose: The end-effector position relative to the rivet hole position is denoted as $\boldsymbol{\mu}_p \in \mathbb{R}^3$. Along with a precision matrix or weighting matrix $\boldsymbol{Q}_p$, used in this work as an isotropic diagonal matrix, we can represent the precision of positioning and construct a quadratic cost as $c_{\text{pos}}(\boldsymbol{q}) = \boldsymbol{\mu}_p^\top \boldsymbol{Q}_p \boldsymbol{\mu}_p$. The configuration $\boldsymbol{q}$ also needs to keep 2 degrees of orientation of the end-effector fixed relative to the rivet hole surface normal, as the bucking bar should not lose contact with the surface. Rotation around the surface normal axis is allowed as they do not interfere with the riveting process. We denote the desired relative orientation as the unit quaternion $\boldsymbol{\mu}_q \in \mathcal{S}^3$, where $\mathcal{S}^3$ denotes a sphere manifold. Using Riemannian manifold properties, we define the orientation cost as $c_{\text{orn}}(\boldsymbol{q}) = \boldsymbol{\mu}_q^\top \boldsymbol{Q}_q \boldsymbol{\mu}_q$ as explained in [12], where $\boldsymbol{Q}_q$ describes the precisions which allows rotations around the surface normal axis.

Handling joint limits: We use a sigmoid function to squash the optimization variable $\boldsymbol{q}$ in between its joint limits with $\hat{\boldsymbol{q}} = \text{sigmoid}(\boldsymbol{q})(\boldsymbol{q}_{\text{u}} - \boldsymbol{q}_{\text{l}}) + \boldsymbol{q}_{\text{l}}$, where $\boldsymbol{q}_{\text{l}}$ and $\boldsymbol{q}_{\text{u}}$ represent the lower and the upper limits, respectively.

Cost on the displacement: The end-effector displacement $\Delta \boldsymbol{x} \in (\mathbb{R}^3, \mathcal{S}^3)$ defined in (4) depends on the frequency, amplitude and direction of the force applied to the end-effector, but also on the configuration of the robot. Our goal is to find configurations which minimize $\Delta \boldsymbol{x}$. However, the computation of such cost requires differentiable model of the mass matrix $\boldsymbol{M}(\boldsymbol{q})$ and the Jacobian $\boldsymbol{J}(\boldsymbol{q})$. We implemented, in Python, the Tensorflow version of the Kinematic and Dynamics Library (KDL) written in C++. This allows us to access all the differentiable Tensorflow functions with optimization tools and differentiable KDL functions. The cost of displacement becomes $d^2 = c_{\text{disp}}(\boldsymbol{q}, \omega) = \|\Delta \boldsymbol{x}(\boldsymbol{q}, \omega)\|^2$.

B. Task and Motion Planning

The above optimization outputs M configurations for each riveting hole p_i, and we need to plan the robot motion to reach all holes. One configuration needs to be chosen at each hole, and then the trajectory visiting the two consecutive configurations needs to be computed. One naive way is to choose the configuration with the smallest displacement $d{=}\Delta \boldsymbol{x}$ at each hole, but it will result in relatively long trajectories between subsequent riveting holes because the selected configurations will likely be far apart.

We propose to use an optimal task and motion planner called RoboTSP [11]. The planner can select the optimal configuration at each position/hole such that it minimizes the desired optimality criteria. In [11], the criterion is to minimize the total length of the trajectory. For the riveting task, we add another criterion corresponding to the total displacement d of the selected configurations.

The planner works in two stages, starting from the task space and then the configuration space. First, it finds the optimal sequence of rivet holes by formulating it as a Travelling Salesman Problem (TSP), where the objective is to minimize the travel distance in the task space. Given the visiting order, the planner then determines the optimal configuration at each hole. In our case, since the riveting task is a collaborative task with another human operator, we prefer the hole order to be specified manually by the operator and skip the first step of RoboTSP. When the human operator can follow the robot motion easily, RoboTSP can determine the visiting order, further improving the final trajectory quality.

The second step is to determine the optimal configuration at each hole. The planner builds a graph connecting the configurations at the consecutive holes (Fig. 4). Each configuration is associated to a node in the graph. Configurations from one hole are connected to the configurations from the next hole. The edge of the graph is assigned a weight that consists of both the configuration space distance and the average displacement $w_{ij} = ||\boldsymbol{q}_i - \boldsymbol{q}_j||^2 + \frac{1}{2}\lambda(d(\boldsymbol{q}_i) + d(\boldsymbol{q}_j))$, where λ is the relative weight of the displacement cost. We then use Dijkstra algorithm to find the optimal path from the first to the last node that minimizes the total weight of the traversed edges, giving us the optimal configuration for each hole. Given the sequence of configurations, we then use the bi-directional RRT algorithm, i.e. RRT-Connect [13] to plan the joint angle trajectory visiting these configurations.

V. Results and Discussion

To evaluate the proposed approach, we consider the structure in Fig. 2 with a total of $N{=}64$ rivet holes separated into three parts (top, middle and bottom) for better visualization of the vibration analysis results. For each hole, we determine $M{=}500$ configurations that reach the hole with the corresponding positions and orientations. In order the determine the Jacobian and the mass matrix of the robot, we used the URDF file of the Panda robot with the parameters estimated in [14]. We used a Tensorflow implementation of the Adam optimizer with a learning rate of 0.01.

A. Optimal Joint Configuration Results

We conducted 3 different experiments where the percussive forces are square signals with a frequency of 70, 100 and 150 rad/s. We first found 500 inverse kinematics solutions which do not take into account the vibrational effects, that we call *Before*, and we further optimized those with the displacement cost included, that we call *After*,

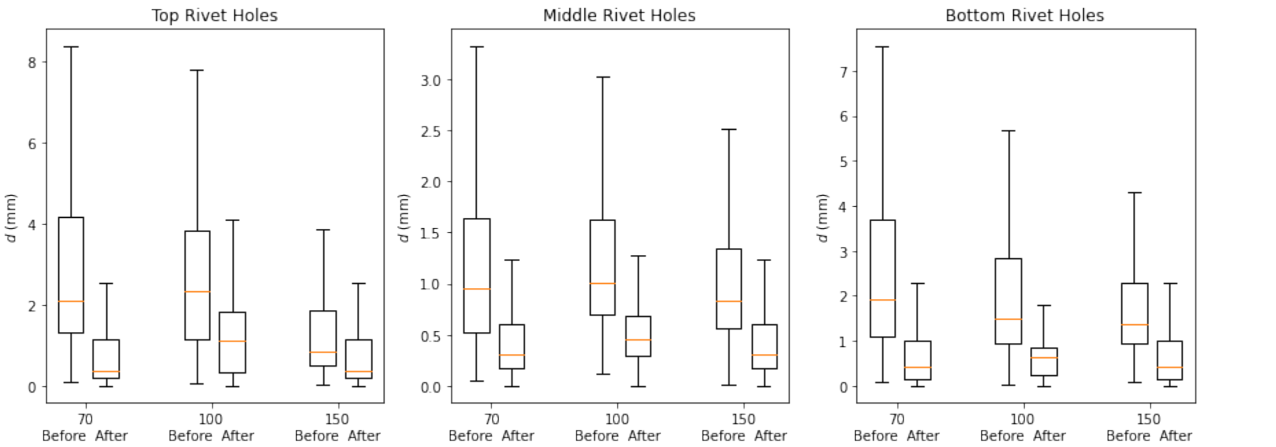


Fig. 5: The median-variance plot of the displacement distance d of the end-effector when it is subject to a square signal with frequencies $\omega = 70$,100 and 150 rad/s at the chosen 64 rivet holes for 500 configurations of the Franka Emika Panda robot. *Before* refers to case where the configurations correspond to the rivet hole poses, but not yet optimized for the displacement, and *After* refers to the proposed optimization procedure.

TABLE I: Reduction of the median displacement in %

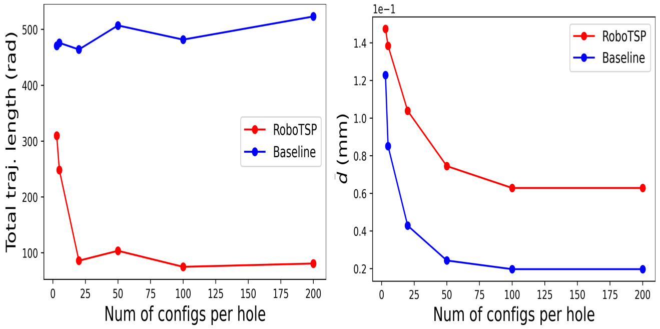
Frequency (rad/s)	Top	Middle	Bottom
70	82.9	67.79	78.17
100	51.8	54.94	56.4
150	62.52	50.98	50.74

referring to the results shown before or after the inclusion of the displacement cost. We show the results of these experiments in Fig. 5, denoting the three parts of the rivets separately to understand better the scale and the reduction in the displacements after the proposed optimization scheme. For the top and bottom rivet holes, *Before* solutions have significantly larger median and variance in the displacements than the middle rivet holes for each of the frequencies. This may be explained by the bad postures that the robot needs to take in order to reach the given top and bottom positions and orientations. *Before* solutions may be the easiest configurations that the inverse kinematics algorithm can find. However, they are not the good solutions in terms of undesired vibrational effects. This can be seen by the *After* solutions which have much lower median and variances in the displacement. Here, with the proposed optimization framework, we exploited the kinematic redundancy of the robot and of the task in order to determine the configurations that result in lower displacements and that are not easily found by inverse kinematics.

The reduction percentage of the median for each case is given in Table I. Note that a choice in the frequency value of the force means a choice in the riveting gun used in the process. One can decide which riveting operations and what kind of riveting gun to use with the available robot by conducting the proposed analysis.

B. Motion Planning Results

Having computed M configurations per hole, we then apply the RoboTSP algorithm to select the optimal configurations per hole and plan the trajectories using RRT-Connect (see the attached multimedia content). We compare against a baseline algorithm, where for each hole we select the optimal configuration as the one that has the smallest d. We use $\lambda = 10$ to balance the trajectory and the displacement cost, making sure that the trajectory is short but the end-effector displacement is still small. For this experiment, we choose the frequency value of 70 rad/s, but the method does not depend on this parameter.



(a)

(b)

Fig. 6: Comparing (a) the trajectory length and (b) the average displacement $\bar{d}$.

Fig. 6 shows the comparison in terms of the total trajectory length ($\sum_{i=0}^{T-1}\|\boldsymbol{q}_i - \boldsymbol{q}_{i+1}\|$) and the average displacement $\bar{d} = \frac{1}{T+1}\sum_{i=0}^{T} d(\boldsymbol{q}_i)$. As expected, RoboTSP produces trajectories with smaller length. It finds consecutive configurations that are close to one another, while still keeping the average displacement $\bar{d}$ small. In comparison, the baseline achieves the lowest $\bar{d}$, at the cost of much longer trajectories ($\sim$ 10 times longer). Indeed, simply selecting configurations that minimize $\bar{d}$ will likely result in configurations that are far apart. This means longer trajectory and hence longer planning time. This is especially true for the planning with the structure in Fig. 2. When the two consecutive configurations

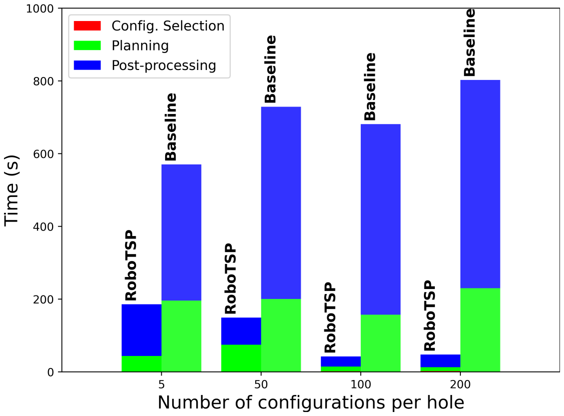


Fig. 7: Breakdown of the computational time.

are far apart, the robot often has to move in and out of the curved structure, which is a difficult planning problem as the path is very narrow. On the other hand, the configurations chosen by RoboTSP are very close to one another, and the only large movement necessary is when the robot moves from the right to the left cluster of the holes.

The planning time of both planners is shown in Fig. 7, which is divided into three parts: 1) The configuration selection step, to select the optimal configuration at each hole; 2) The motion planning step, to plan the trajectory between two consecutive configurations using RRT-connect; and 3) The post-processing step, to shorten the trajectory using the standard shortcut algorithm.

RoboTSP and the baseline only differ in the way the configuration at each hole is selected, while the remaining two steps are the same. The first step is crucial, however, as it affects the next two steps. Due to shorter configuration distances between consecutive configurations, RoboTSP results in much shorter planning time as compared to the baseline for $M > 1$. However, as M increases, the configuration graph becomes larger, and the configuration selection time increases. A good compromise is obtained at around ~ 100 configurations per hole, where the computation time is kept low while the trajectory length is small. Note that the configuration selection time even at $M = 200$ is only ~ 2 s, which is still negligible compared to the other planning times, and hence cannot be visualized easily in Fig. 7.

Finally, in the context of collaborative work with a human operator, the fast computation enables us to make a real time change and replanning of the remaining part of the task. For example, the operator may decide during the operation that the hole order should be changed. Computing a new set of optimal configurations when $M = 100$ only takes less than 2 seconds, while the planning time to reach one hole is on average less than 1 second, so this can be done online without requiring the operator to wait. We demonstrate an example of this real-time hole re-ordering in the supplementary video.

VI. Conclusions and Future Work

In this article, we addressed two major challenges in the automatization of industrial riveting with a collaborative robot. First, we proposed a principled way of exploiting the kinematic redundancies in these flexible joint robots for minimizing the forced vibration of the end-effector. We optimized redundant joint configurations for each rivet hole to result in minimal displacements of the end-effector while it is subject to percussive loading. Second, we exploit our optimized joint configurations in an optimal motion planning algorithm for a faster execution of the riveting task. We showed that our method can achieve a short total trajectory length while still minimizing the average end-effector displacement at each hole.

Future work will focus on a real industrial setup. Furthermore, in our current optimization formulation, the random initial guesses may still result in the same configurations after optimization, as there is no specific cost to force them to be different. We will investigate in future work if better optimization costs could be employed to address this issue.

References

[1] S. Nie, Y. Li, G. Shuai, S. Tao, and F. Xi, "Modeling and simulation for fatigue life analysis of robots with flexible joints under percussive impact forces," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 292 – 301, 2016.

[2] Y. Li, S. Guo, and F. J. Xi, "Preferable Workspace for Fatigue Life Improvement of Flexible-Joint Robots Under Percussive Riveting," *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 4, 2017.

[3] Y. Li, J. Ji, S. Guo, and F. J. Xi, "Process Parameter Optimization of a Mobile Robotic Percussive Riveting System With Flexible Joints," *Journal of Computational and Nonlinear Dynamics*, vol. 12, no. 6, 2017.

[4] T. Masiak, A. Kanso, R. Müller, and M. Vette, "Development of a human-robot-collaboration system using the example of a riveting process in aircraft assembly," *Journal of Robotics and Automation*, vol. 2, pp. 69–77, 2018.

[5] P. Zhang and Y. Li, "New indices for evaluating vibration characteristics of flexible-joint robots," *Applied Sciences*, vol. 10, no. 14, 2020.

[6] M. Saha, G. Sánchez-Ante, and J.-C. Latombe, "Planning multi-goal tours for robot arms," in *IEEE International Conference on Robotics and Automation*, vol. 3, 2003, pp. 3797–3803.

[7] M. Saha, T. Roughgarden, J.-C. Latombe, and G. Sánchez-Ante, "Planning tours of robotic arms among partitioned goals," *The International Journal of Robotics Research*, vol. 25, no. 3, pp. 207–223, 2006.

[8] P. T. Zacharia and N. Aspragathos, "Optimal robot task scheduling based on genetic algorithms," *Robotics and Computer-Integrated Manufacturing*, vol. 21, no. 1, pp. 67–79, 2005.

[9] P. T. Zacharia, E. K. Xidias, and N. A. Aspragathos, "Task scheduling and motion planning for an industrial manipulator," *Robotics and computer-integrated manufacturing*, vol. 29, no. 6, pp. 449–462, 2013.

[10] E. Kolakowska, S. F. Smith, and M. Kristiansen, "Constraint optimization model of a scheduling problem for a robotic arm in automatic systems," *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 267–280, 2014.

[11] F. Suárez-Ruiz, T. S. Lembono, and Q.-C. Pham, "RoboTSP–a fast solution to the robotic task sequencing problem," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2018, pp. 1611–1616.

[12] S. Calinon, "Gaussians on Riemannian manifolds: Applications for robot learning and adaptive control," *IEEE Robotics Automation Magazine*, vol. 27, no. 2, pp. 33–45, 2020.

[13] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, vol. 2, 2000, pp. 995–1001.

[14] C. Gaz, M. Cognetti, A. Oliva, P. Robuffo Giordano, and A. De Luca, "Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4147–4154, 2019.