A Pragmatic Framework for Mobile Redundant Manipulator Performing Sequential Tasks

Olivier RAYMOND
Arts et Metiers Institute of Technology
LISPEN and SAFRAN
Niort, FRANCE
Olivier.RAYMOND_1@ensam.eu

Adel OLABI
Arts et Metiers Institute of Technology
LISPEN
Lille, FRANCE
Adel.OLABI@ensam.eu

Richard BEAREE
Arts et Metiers Institute of Technology
LISPEN
Lille, FRANCE
Richard.BEAREE@ensam.eu

***Abstract*—In this paper, a framework combining base placement, path planning and redundancy resolution for a mobile manipulator performing sequential tasks, such as screwing, drilling or assembling tasks, is proposed. For a set of given tasks, the outputs of the proposed algorithm meet the following practical performance indicators: minimization of the number of the base positions, minimization of the number of manipulator joint configuration changes, feasibility of each task considering the force capacity of the manipulator (which takes benefit of redundancy resolution) and path planning of the end-effector motion with obstacle avoidance. The effectiveness of the proposed approach is evaluated considering a 3 DOFs mobile platform and a 7 DOFs manipulator performing screwing in a application with 42 tasks.**

***Index Terms*—Redundant mobile manipulator, Force capacity, Path planning, Redundancy resolution, Combinatorial optimization**

I. Introduction

Industry 4.0 has opened the way to multiple forms of automation that aims to optimize productivity. In this context, mobile platforms and mobile manipulators have increasing demand in logistics, manufacturing or assembly processes due to their high degree of flexibility.

Beyond the extended mobility, another interest using mobile manipulation is to increase the workspace of the manipulator. Considering a robotic application described by a set of sequential tasks, one of the main issues consists in generating an optimized set of platform positions associated to a set of manipulator configurations taking into account time, kinematic and collision avoidance constraints [1]. Additional constraints may be considered, since the mobile manipulator is kinematically redundant for a given pose (position and orientation of the end-effector). Indeed, a conventional mobile manipulator has at least nine Degrees Of Freedom (DOFs). The redundancy of mobile manipulators could be exploited to respect or supplement different constraints such as collision avoidance, trajectory planning or stiffness optimization. For sequential operations like screwing, drilling or assembling, ensuring the manipulator's capacity to supply sufficient force to perform the task is mandatory. It is especially true for lightweight collaborative robots with low payload.

This research was partially supported by Safran Electrical and Power. We are thankful to our colleague Emmanuel FRERE, who provided expertise and materials that greatly assisted the research.

Some additional information from this paper are present in my GitHub (https://github.com/Isdinval) to clarify the present project: some methods are explained and the visualization of the result's simulation is present. The code isn't added since it is confidential.

A. State of the art

1) End-effector force capacity determination: For a given task with the required wrench f to be produced at a robot end-effector, the joint's torques τ depends on the robot configuration q (see equation 1), with the robot geometric Jacobian matrix J:

$$\tau(q) = J^T(q).f \qquad (1)$$

Many approaches to estimate the Force Capacity (FC) are suggested in the literature. Yoshikawa introduced first the force manipulability ellipsoids concept [2], then the dynamic manipulability ellipsoids [3]. Chiacchio improved it by including the effect of gravity [4] and proposed a new definition for redundant manipulators [5]. The case of manipulators in singular configurations is also analyzed. Kokkinis *et al.* introduced the force polytopes [6]. Chiacchio completed the force polytope definition with the redundant case [7]. Bolwing *et al.* proposed the Dynamic Capability Equations [8]. More recently, Busson developed the Force Capacity Index (FCI) which describe accurately a specific wrench capacity on the redundancy space of serial redundant robots to define a relevant configuration for a given task [9].

2) Positioning the mobile manipulator: In many applications, reaching a sequence of tasks with the mobile manipulator's end effector is required. It is done thanks to a combination of platform and manipulator re-positioning. In the following methods [10, 11, 12, 13], authors search to find only one optimal base position. Berenson [10] presented an optimization based approach to path planning for mobile manipulators with two steps: optimization and planning. In the optimization step, optimal configurations are generated using a co-evolutionary algorithm and for the planning step, the path is created using Rapidly-exploring Random Tree (RRT) methods. Zacharias [11] also presented a two-step framework to carry out constrained linear trajectory. Du [12] proposed a method to find the optimal base motion using an index to maximize the manipulability measure of the manipulator. Bodily [13]

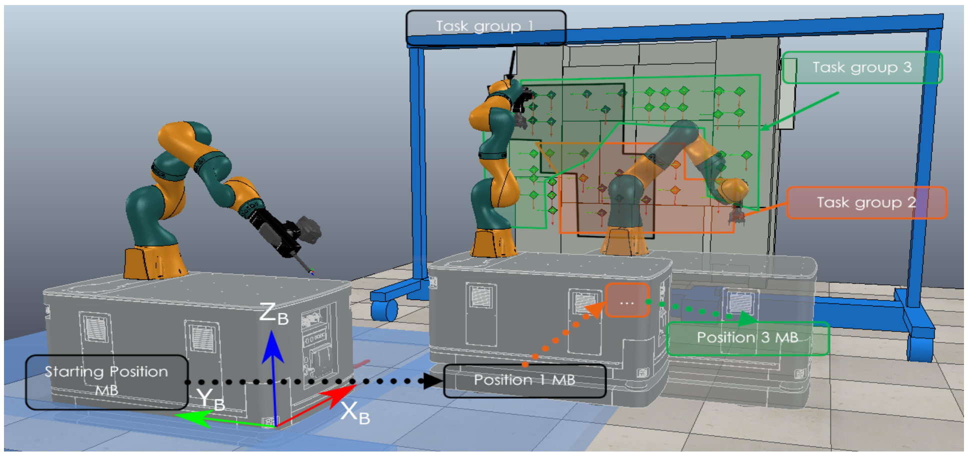

Fig. 1: Schematic diagram of a screwing application with a view of the output of the proposed method. Multiple positions of the Mobile Base (MB) are represented. Each MB position is linked to a task group.

presented a method for mobile manipulator that optimized simultaneously position and joint position to follow smooth specified end-effector trajectory.

Some other methods are based on the exploitation of the capability and reachability of the robot. Malhan [14] constructed the capability map to evaluate whether the waypoints on the surface of a workpiece satisfy the position constraints while avoiding singularity. Finally, some base positioning methods are based on the positioning of the base given the end-effector pose. Ren [15] proposed a method to optimize the base positioning of a mobile manipulator to reach a sequence of end-effector positions with orientation constraints to perform local painting tasks. Vahrenkamp [16] presented a method based on reachability analysis. The base position of a mobile manipulator can be efficiently found from the inverse reachability distribution. Xu [17] proposed to plan a sequence of base positions to efficiently and robustly collect objects stored in distinct trays. Rishi [18] developed a motion planner that finds the minimum number of mobile base placements to find robotic arm trajectories that can cover a large complex part. A branch and bound search algorithm has been developed with an efficient branch guiding and pruning heuristics.

In this paper, a base positioning method based on the end-effector pose is presented. It is a multi-objectives framework combining the FC criteria and the minimization of the number of mobile base by exploiting the kinematic redundancy of the mobile manipulator and using the graph theory. An industrial experimentation has been done to validate screwing problems that required torque management.

B. Use-case description and objectives

Fig. 1 illustrates the application considered in this work, where a 10 DOFs mobile manipulator composed of 7 DOFs serial robot arm mounted on a 3 DOFs holonomic mobile platform has to perform a sequence of discrete tasks, such as screwing. The spatial distribution of the tasks is large enough that several mobile base motions are required. From the starting position, the mobile manipulator moves to the first base position and performs a series of screwing tasks (group 1) while avoiding self-collision and the collision between the mobile manipulator and its environment, and so on for the other tasks group.

The application objectives are :

1) Minimize the number of movements of the mobile base: The total operation time increases with the number of base positions due to: a) The mobile manipulator decelerates and then accelerates at each base position and b) due to the positioning accuracy of conventional base (a few centimeters), the resulting base position may deviate significantly from the desired position. The mobile robot has then to perform a time-consuming fine repositioning process. It is therefore essential to minimize the number of movements of the base.

2) Ensure the task feasibility: With its 10 DOFs, the mobile manipulator is a kinematically redundant system. Without loss of generality, three redundant parameters are considered in this work: the Y-axis and X-axis position of the mobile manipulator (Y_B, X_B) (see fig. 1) and the swivel angle $Swiv$ (see fig. 2). The FCI method [9], which determines the manipulator's FC inside the redundancy space according to the joints configuration, is used in this study to fix the choice of the redundancy parameters.

3) Generate short paths: To generate shorter paths and improve workstation safety by removing large movements, the choice of joints configuration and task order is essential. The choice of the manipulator configurations is done thanks to two sub-criteria (see fig. 2): the manipulator's form and the swivel angle $Swiv$. Hence, two constraints are added to maintain whenever possible: a) the same manipulator's form and b) the same swivel angle. The manipulator's form corresponds to a joint configuration for a given pose which is chosen thanks to the choice of a binary combination of the triplet (shoulder, elbow, wrist). They are eight possibles combinations for the considered redundant manipulator (Kuka IIWA). The tasks order choice may be solved by using a combinatorial optimization technique: the traveling salesman problem. Additionally, in case it is not possible to fix the same form, a strategy can be implemented to limit and reduce the traveling path between two tasks: a wrist motion is shorter than a wrist+elbow+shoulder motion. Thus, a qualification of the path planning is possible.

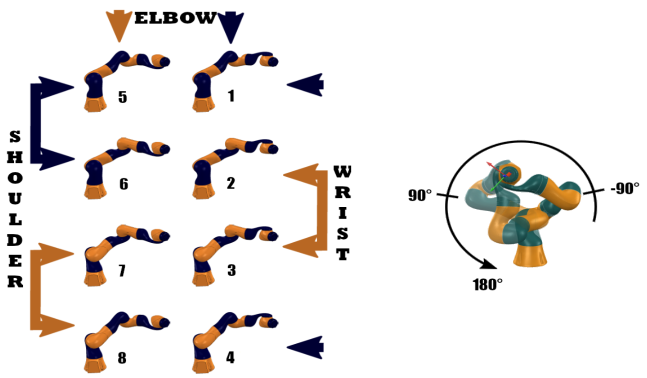

Fig. 2: Manipulator's forms for a given pose (left) and the Swivel Angle for 3 manipulator configurations (right)

This study proposed a three-step framework to meet efficiently the previous goals. The steps are to: a) minimize the number of the mobile base positions, b) provide redundancy resolution according to the FC at the end-effector level guaranteeing the task success and c) generate the path considering obstacle avoidance. The results section (III) demonstrates the effectiveness of the proposed approach by simulation and experimental tests.

II. Methodology

This section details the proposed three-step framework. The flowchart depicted in fig. 3 describes the interactions between the different steps and their associated inputs and outputs.

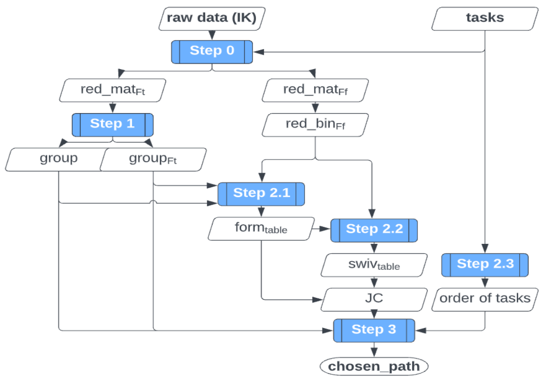

Fig. 3: Input/output diagram of the proposed framework

A. Step 0 - Initial data transformation

The redundant space is discretized in N_x, N_y values, representing the planar coordinates (Y_B, X_B) of the Mobile Base (MB) position (see fig. 1) and N_{swiv} values for the chosen manipulator redundancy parameters, i.e. the discrete values of the swivel angle $Swiv$ (see fig. 2). This stage consists in transforming raw data (joints configuration obtained by using the Inverse Kinematic model of the robot) into enriched data to make informed decisions. The outputs are the reduced feasible task matrix red_mat_{Ft} and the reduced feasible form matrix red_mat_{Ff}. They correspond respectively to all the feasible tasks (from a kinematic and force point of view) for each triplet (Y_B, X_B, $Swiv$) and all the feasible forms for each quadruplet (Y_B, X_B, $Swiv$, $Task$).

B. Step 1 - Tasks groups and sequence of mobile base positions

The step 1 goal consists in generating the group of feasible mobile base positions. The main sub-steps are: 1. Gather input data, 2. Create a directed graph, 3. Find the shortest path in this graph and 4. Retrieve the mobile base positions. The corresponding pseudocode is presented in algorithm 1.

Algorithm 1: Tasks group and Mobile Base (MB) positions determination

```
Input: Reduced feasible task matrix: $red\_mat_{Ft}$,
Output: MB positions groups: $group$, feasible task
        group: $group_{Ft}$
$node_{pair}$ ← Build the pair of nodes (see algorithm 2).
$root_{nodes}$ ← Retrieve the root nodes.
$end_{nodes}$ ← Retrieve the end nodes.
$state_{tree}$ ← Create the directed graph.
Retrieve the length of the paths between each root
  nodes and end nodes in $state_{tree}$.
Retrieve the minimum path (minimum number of
  nodes) which maximizes the number of feasible tasks
  (spatial criteria).
Retrieve the MB position $group$ and the feasible task
  group $group_{Ft}$.
```

The node pair $node_{pair}$, used in algorithm 1, represents the source and target nodes and their directions in a directed graph. There are two types of input data that are required to compute $node_{pair}$. Algorithm 2 describes specifically the steps of this computation. The input is the set of the reachable and feasible tasks for each base position forming a N_xxN_yxN_{Swiv} dimension matrix named red_mat_{Ft}, with N_{Swiv} the number of swivel angle values. The node equivalent matrix is obtained from the previous matrix thanks to the conversion of the matrix index representation into a linear index representation (function Sub2Ind) for each base position.

Concerning the construction of the pair of nodes, it is subject to several constraints. In the red_mat_{Ft} matrix, the only cells considered are those having a solution (non-zero values). Moreover, for each cell having a solution, a search among all the other non-zero cells is carried out having for condition that they must have the maximum task from the current cell. The nodes correspond to the value of the linear index of red_mat_{Ft}. Once the pair of nodes is obtained, the construction of the directed graph is possible (see algorithm

Algorithm 2: Creation of the node pair

```
Input: Reduced feasible task matrix : $red\_mat_{Ft}$,
Output: Pair of nodes: $node_{pair}$
for each couple $(Y_B, X_B)$ from $red\_mat_{Ft}$ do
    if $red\_mat_{Ft}(Y_B, X_B)$ has solutions then
        $cur\_node \leftarrow$ Sub2Ind($Y_B, X_B$)
        for each couple $(Y_B, X_B)$ from $red\_mat_{Ft}$ do
            if $red\_mat_{Ft}(Y_B, X_B)$ has solutions
             containing the maximum task from the
             current cell then
                $new\_node \leftarrow$ Sub2Ind($Y_B, X_B$)
                $node_{pair}$.add([$new\_node$, $cur\_node$])
            end
        end
    end
end
```

Algorithm 3: Choice of the manipulator form

```
Input: Groups: $group$, feasible task group: $group_{Ft}$,
       reduced binary feasible form $red\_bin_{Ff}$
Output: form table $form_{table}$
for each group do
    for each task of the current group and each form
     of the current task do
        $form_{table} \leftarrow$ Add tasks which have at least
         one feasible swivel value with the given form
         from $red\_bin_{Ff}$.
        if the length of $form_{table}$ is equal to the length
         of the current group of feasible tasks then
            Break.
        end
    end
end
```

1). In this graph, several path queries are done between each root nodes and each end nodes. The chosen solutions are the queries minimizing the number of nodes, i.e. the number of task groups, which is equivalent to the number of MB positions and maximizing the number of feasible tasks. It may be possible that several solutions exist, then the cartesian distance between each MB positions is used as spatial criterion to differentiate the solutions.

C. Step 2 - Redundancy resolution and manipulator configurations

The previous algorithm allowed to establish groups of tasks and determine the set of MB positions. The next step consists in setting the value of the redundant parameter (the swivel angle) and the form of the manipulator for each task of the same group. The goals for this manipulator configurations selection stage are to ensure the desired wrench f, to keep the same joint configuration and the same swivel angle value as much as possible.

1) Choice of the manipulator form: This choice can be done thanks to the reduced feasible form matrix red_mat_{Ff} for each swivel value and each task. Each cell of this matrix contains the feasible forms (from 1 to 8) of the manipulator represented as the triplet (shoulder, elbow, wrist). By bursting this matrix into 8 sub-binary matrix red_bin_{Ff}, it is possible to determine the chosen forms for each task. The algorithm 3 outputs the choice of the manipulator form, which corresponds to the first available solution.

2) Choice of the swivel angle value: Once a form has been associated with each task ($form_{table}$), the possible swivel values for these forms are then retrieved. The reduced binary feasible form matrix red_bin_{Ff} is then reduced to a single binary matrix representing the tasks as a function of the swivel angle. The first step is to eliminate unnecessary solutions. The solutions with the least change of swivels are kept. A search by block column is thus to be carried out.

The search by block column is a data transformation technique in which the combinatorial problem is reduced to "column" solutions. It is an iterative method whose result is to find columns of data (swivel angle here) starting from an entry index and finishing with a zero value. The new entry index corresponds to the index of the zero value found previously. The exploration is finalized when the table has been fully explored from row_1 to row_{end}. The table associating each task with a swivel value is named $swiv_{table}$.

3) Task order reassignment: For each task, a form and a swivel value has been assigned. It is possible to retrieve the joints configurations. These sub-criteria has been used to reduce the length of the future path generated. However, it is possible to reduce it even more thanks to another combinatorial optimization method: the Traveling Salesman Problem (TSP).

D. Step 3 - Point-to-point path planning

The used path planning method is based on the LazyRRT algorithm [19] thanks to the Open Motion Planning Library (OMPL) [20]. This single-query algorithm takes the manipulator kinematic constraints and the position of the obstacles into consideration. The founded path is smoothed by using spline interpolation. LazyRRT algorithm is a combination of the RRT algorithm and the Lazy algorithm that tries to connect, by a tree, the starting configuration and the goal configuration. It possesses a particularity concerning the collision management: it is checked only when a path has been generated. Concerning the path generation itself (see algorithm 4), several paths will be generated between the tasks via the LazyRRT algorithm. A classification based on the minimization of the joint motion length is performed. The chosen path is the one with the shortest joint motion length.

III. Results

In this section, a use-case with 42 tasks is considered and validated experimentally on the test bench presented in fig. 4. The mobile platform is the KMR holonomic platform from Kuka and the manipulator is a Kuka IIWA (payload 14kg), which is a seven DOFs serial robot. For the FC criteria, the required wrench at the end-effector is f = [270 N, 0 N, 0 N, 0 N.m, 0 N.m, 0 N.m] expressed in the Tool frame. In this experimentation, configurations and paths are searched using

Algorithm 4: Point-to-point path planning

```
Input: Groups: $group$, feasible task group: $group_{Ft}$,
       joints configurations of all tasks: $JC$ and the
       order of tasks
Output: Chosen path: $chosen\_path$
for each group and each task in the group do
    for each $path\_idx$ in $nbPaths$ do
        $path \leftarrow$ LazyRRT($JC_{Task\_prev}, JC_{Task\_curr}$)
        $path\_Total(group, path\_idx)$.add($path$)
    end
    $j_{dst} \leftarrow$ Classify paths in $path\_Total(group, :)$ by
      their joints distance
    $chosen\_path \leftarrow$ Path which minimizes $j_{dst}$
end
```

the proposed framework: 1) tasks are reachable and feasible, 2) the same manipulator form is maintained as much as possible and 3) the path is feasible and collision-free.

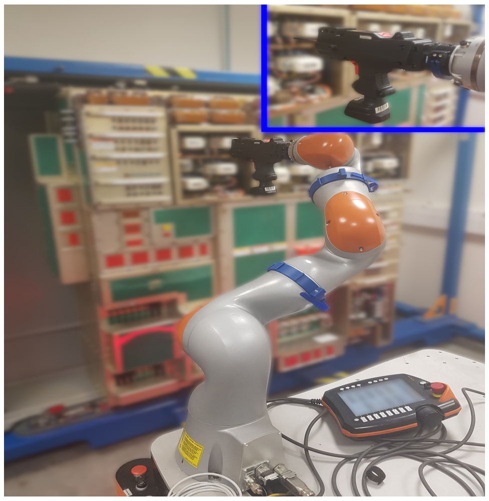

Fig. 4: Test Bench - A mobile redundant manipulator performs a screwing operation of a contactor on an electrical cabinet.

Three different scenarios are compared for the previous use-case. The first scenario, referred as non-redundant, considers the non-redundant manipulator case (i.e. $Swiv = 0$), which is equivalent to consider a conventional 6-DOFs manipulator. For the second scenario, hereafter referred as pure FCI based-method, the mobile manipulator configurations are chosen among the solutions which maximize the FCI. Finally, the third and last scenario, hereafter referred simply as full method, takes benefit of the manipulator redundancy parameter $Swiv$ and produces a set of configurations and paths, which ensures the capacity to produce the required wrench.

A. *Mobile base positioning and tasks feasibility*

In this part, four parameters are considered: the redundant volume resolution ($N_x.N_y.N_{Swiv}$), the number of feasible tasks, the number of Mobile Base positions (MB Positions) and the Cartesian Distance (Cart Dst) needed by the mobile base to reach every position. Fig. 5 gives a representation of the spatial limits for base motion.

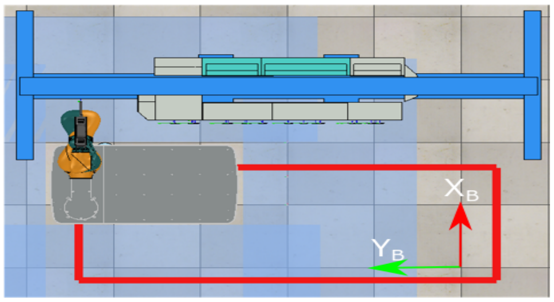

Fig. 5: Top view of the MB motion limits in the X-Y plane.

A comparison for each scenario is given in table I. The full-method scenario and the pure FCI-based scenario behave quite well: all the tasks are feasible, with only a few MB positions (2 or 3), which depends on the chosen resolution. By contrast, for the non-redundant manipulator scenario, there is only one case where all the tasks are feasible for a resolution of 18.48.1. Additionally, the number of MB positions (7 or 8), and the traveling distance will increase the operation time. Even with the higher resolution grid, the MB traveling distance is four times greater than the two other scenarios.

Parameters	Non Redundant	Pure FCI-based	Full Method
Resolution	5.5.1	5.5.5	5.5.5
Task Feasibility	83%	100%	100%
MB Positions	7	3	3
Cart Dst (m)	3.83	0.45	0.45
Resolution	10.10.1	10.10.10	10.10.10
Task Feasibility	95%	100%	100%
MB Positions	8	2	2
Cart Dst (m)	1.63	0.20	0.20
Resolution	18.48.1	18.48.25	18.48.25
Task Feasibility	100%	100%	100%
MB Positions	8	2	2
Cart Dst (m)	0.84	0.19	0.19

TABLE I: Comparison between the 3 scenarios

B. *Point-to-point path planning*

In this part, the four parameters considered are the redundant parameter resolution for the swivel angle (N_{Swiv}), the path feasibility considering the feasible tasks only, the path joint distance and the manipulator traveling time. One can note that each time a MP motion is required, the manipulator has to return to its original position for security reason. For each scenario and each resolution, the path planning is feasible (100%). For the pure FCI-based scenario, both the path joint distance and the traveling time are higher. In this case, the manipulator form changes frequently, which is a time-consuming motion for the manipulator.

For each resolution (5.5, 10.10 and 18.48), the FCI-Based and non-redundant scenarios are worse than the result of the proposed method. The improvement range of the proposed full-method vs the non-redundant scenario is: 1.between 108% (low resolution) and 223% (medium resolution) for the path joint distance, 2. between 127% (small resolution) and 243% (high resolution) for the traveling time. Similarly, the improvement range of the proposed full-method vs the pure FCI-based scenario is: 1. between 212% (medium resolution) and 292% (high resolution) for the path joint distance, 2. between 143% (small resolution) and 207% (medium resolution) for the traveling time.

Finally, the proposed full-method seems to be a truly effective and pragmatic tool for the full exploitation of a redundant mobile manipulator performing sequential operations.

Parameters	Non Redundant	Pure FCI-based	Full Method
Resolution	5.5.1	5.5.5	5.5.5
Path Feasibility	100%	100%	100%
Path Dist (°)	40.96	84.44	37.57
Travel. Time (s)	172	193	135
Resolution	10.10.1	10.10.10	10.10.10
Path Feasibility	100%	100%	100%
Path Dist. (°)	50.84	48.33	22.81
Travel. Time (s)	219	186	90
Resolution	18.48.1	18.48.25	18.48.25
Path Feasibility	100%	100%	100%
Path Dist. (°)	33.26	66.76	22.85
Travel. Time (s)	187	153	88

TABLE II: Comparison between the 3 scenarios

IV. Conclusion and future work

In this paper, a three-step framework has been proposed to position and generate the trajectories for a mobile manipulator to perform sequential tasks. The proposed solution minimizes the number of base positions, ensures the required force capacity at the manipulator end-effector level (for redundant or non-redundant manipulators), while maintaining the manipulator configuration as much as possible and generating collision-free trajectories. Simulations and experimental tests for a multitasks case demonstrate the efficiency and versatility of the proposed method, which can be used for redundant and non-redundant manipulators.

Future works will address the extension and application of the previous framework to a bi-manual mobile manipulator.

References

[1] Q. Zhiang and M.Y. Zhao. "Minimum time path planning of robotic manipulator in drilling/spot welding tasks". In: *Journal of computational design and engineering* 3 (2016), pp. 132–139.

[2] T. Yoshikawa. "Manipulability of Robotic Mechanisms". In: *The International Journal of Robotics Research* 4 (1985), pp. 3–9.

[3] T. Yoshikawa. "Dynamic manipulability of robot manipulators". In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation* 2 (1985), pp. 1033–1038.

[4] P. Chiacchio et al. "Influence of Gravity on the Manipulability Ellipsoid for Robot Arms". In: *Journal of Dynamic Systems Measurement and Control-transactions of The Asme* 114 (1992), pp. 723–727.

[5] P. Chiacchio. "Dynamic manipulability ellipsoid for redundant manipulators". In: *Robotica* 18 (2000), pp. 381–387.

[6] T. Kokkinis and B. Paden. "Kinetostatic performance limits of cooperating robot manipulators using force-velocity polytopes". In: *In Proceedings of the ASME Winter Annual Meeting* (1989), pp. 151–155.

[7] P. Chiacchio, Y. Bouffard-Vercelli, and F. Pierrot. "Force polytope and force ellipsoid for redundant manipulators". In: *Journal of Robotic Systems* 14 (1998), pp. 613–620.

[8] A. Bowling and O. Khatib. "The dynamic capability equations: A new tool for analyzing robotic manipulator performance". In: *Robotics, IEEE Transactions on* 21 (2005), pp. 115–123.

[9] D. Busson. "Management of mobile and redundant manipulators in cluttered and dynamic environment". PhD thesis. Arts et Métiers ParisTech, 2018.

[10] D. Berenson, J. Kuffner, and H. Choset. "An Optimization Approach to Planning for Mobile Manipulation". In: May 2008, pp. 1187–1192.

[11] F. Zacharias et al. "Positioning Mobile Manipulators to Perform Constrained Linear Trajectories". In: Oct. 2008, pp. 2578–2584.

[12] B. Du, J. Zhao, and C. Song. "Optimal Base Placement and Motion Planning for Mobile Manipulators". In: vol. 4. Aug. 2012.

[13] D. Bodily, T. Allen, and M. Killpack. "Motion planning for mobile robots using inverse kinematics branching". In: May 2017, pp. 5043–5050.

[14] R. Malhan et al. "Identifying Feasible Workpiece Placement with Respect to Redundant Manipulator for Complex Manufacturing Tasks". In: May 2019.

[15] S. Ren et al. "A Method for Optimizing the Base Position of Mobile Painting Manipulators". In: *IEEE Transactions on Automation Science and Engineering* PP (Oct. 2016), pp. 1–6.

[16] N. Vahrenkamp, T. Asfour, and R. Dillmann. "Robot placement based on reachability inversion". In: May 2013, pp. 1970–1975. ISBN: 978-1-4673-5641-1.

[17] X. Jingren et al. "Planning a Minimum Sequence of Positions for Picking Parts From Multiple Trays Using a Mobile Manipulator". In: *IEEE Access* PP (Dec. 2021).

[18] M. Rishi and G. Satyandra. "Finding Optimal Sequence of Mobile Manipulator Placements for Automated Coverage Planning of Large Complex Parts". In: Aug. 2022.

[19] G. Sanchez-Ante. "Path Planning Using a Single-Query Bi-directional Lazy Collision Checking Planner". In: 2002, pp. 41–50.

[20] I. A. Şucan, M. Moll, and L. E. Kavraki. "The Open Motion Planning Library". In: *IEEE Robotics & Automation Magazine* 19.4 (Dec. 2012), pp. 72–82.