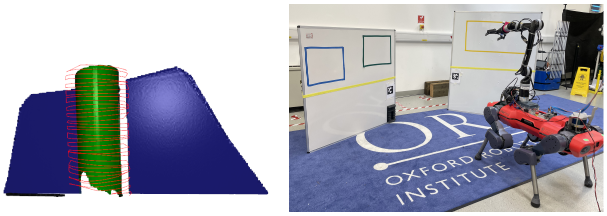
Asymptotically Optimized Multi-Surface Coverage Path Planning for Loco-Manipulation in Inspection and Monitoring

Kim Tien Ly, Matthew Munks, Wolfgang Merkt, Ioannis Havoutis

Abstract*— Regular inspection and monitoring of aging assets are crucial to safe operation in industrial facilities, with remote robotic monitoring being a particularly promising approach for asset inspection. However, vessels, pipework, and surfaces to be monitored can follow complex 3D surfaces, and frequently no 3D as-built models exist. In this paper, we present an end-to-end solution that uses an optimization method for coverage path planning of multiple complex surfaces for mobile robot manipulators. The system includes a two-layer hierarchical structure of optimization: mission planning and motion planning. The surface sequence is optimized with a mixed-integer linear programming formulation while motion planning solves a whole-body optimal control problem considering the robot as a floating-base system. The loco-manipulation system automatically plans a full-coverage trajectory over multiple surfaces for contact-based non-destructive monitoring after unrolling the 3D-mesh region-of-interest selected from the user interface and projects it back to the surface. Our pipeline aims at offshore asset inspection and remote monitoring in industrial applications, and is also applicable in manufacturing and maintenance where area coverage is critical. We demonstrate the generality and scalability of our solution in a variety of robotic coverage path planning applications, including for multi-surface asset inspection using a quadrupedal manipulator.

I. Introduction

Coverage path planning refers to the task of finding a trajectory (or path) that efficiently traverses a particular area while ensuring that the area is sufficiently covered. This can be a vital component for a robotic system performing tasks in a range of applications, for example, in offshore inspection [1], area monitoring [2], search and rescue [3] or landmine scanning [4]. These are scenarios where it is either difficult or dangerous for humans to work on. In the fields of agriculture and horticulture, it is beneficial to use coverage planning for agricultural machines to reduce fuel consumption by optimizing their movements during harvesting or pruning crops [5]. Examples of industry companies that develop robots with coverage techniques are Gray Matter Robotics for material sanding; and Skyline Robotics for window cleaning. Industrial tasks like surface treatment [6] tend to be repetitive and require high precision or consistent performance. Overall, coverage planning is industrially widely applicable and can be significantly more efficient and accurate when performed by robots. In field operations, where robots need to fly over or drive over a field, coverage planning requires considering either obstacles avoidance or terrain inclinations. On the other hand, with articulated robot systems covering a surface, the complication comes from the shape of a surface, robot reachability, and kinematic constraints. These features make a one-fit-all coverage planning system a challenging task to solve in robotics research.



(a) Coverage path (b) Physical robot setup

Fig. 1: Overview of the deployment of our proposed quadrupedal manipulator system: (a) Example of a coverage path on a reconstructed and segmented surface. (b) Physical quadrupedal robot in a multi-surface inspection scenario.

The main challenge of coverage planning is dealing with sets of complex and disjoint surfaces. For example, in real-world applications, pipes are frequently obstructed by valves, or there are many separate pipes in an inspection site requiring complex planning of inspection sequences. This highlights the need for an automated way of computing an optimized coverage path, given a list of surfaces, which can be widely applied to different applications. The optimization-based approach not only saves time and energy but also produces better solutions compared to human performance, in terms of precision and repeatability.

Intuitively, mobile robots are often preferred in large-scale surveillance, where there are many objects to be inspected, in order to achieve full coverage results. Legged robots have been widely used in unstructured environments due to their flexibility and adaptability to different terrains, and their ability to reposition their base to extend the workspace of an on-board manipulator. Therefore, they are an ideal choice for the inspection monitoring task at hand. While coverage tasks require the robot system's mobility and flexibility, we must take into account whole-body planning to guarantee reachability and stability of the robot system.

Figure 1 visualizes an example solution to coverage planning on a vertical pipe and a multi-surface task evaluation setup. In this paper, we address common issues that arise in robots performing coverage paths in remote industrial applications, which include coverage planning on disjoint surfaces, whole-body motion planning on legged manipulators, and autonomy in an unknown environment.

A. Related work

1) Robotic inspection and monitoring: Non-destructive evaluation (NDE) has been an effective approach in exam-

All authors are with the Oxford Robotics Institute, University of Oxford. Their respective emails are ktien@robots.ox.ac.uk, matthew.munks@exeter.ox.ac.uk, {wolfgang,ioannis}@robots.ox.ac.uk.

ining material quality in industrial or manufacturing applications such as inspection and monitoring. The advantage of NDE is that it can detect faults such as cracks, cavities and flaws without having to alter them (e.g., by drilling or cutting), which causes no disruption in service. This is especially useful in cases where regular monitoring is needed, e.g., for regular inspection of pressurized vessels used for a significant range of industrial applications. A traditional non-contact approach to non-destructive testing in manufacturing applications is visual inspection, e.g. as presented in [7]. This method can be used on top of other methods to detect visible faults on surfaces quickly, however, is not capable of detecting issues within the material or on the inside/backside of surfaces. Radiography [8] and thermography [9] are other methods that do not require contact with samples, which are applied in checking delamination in composite materials. Non-contact NDE is often fast in data collection in comparison with contact methods, hence, has its own instrumental sensitivity. Meanwhile, for NDE sensors like ultrasonic or electromagnetic, it is essential to ensure surface contact so that they can effectively collect data of the materials. An added benefit of these sensing modalities is that they can sense hidden defects that cannot be inspected through non-contact methods. An example with this approach is an electromagnetic acoustic transducer (EMAT) sensor [1], where the system automatically detects thickness change and builds a thickness map on a steel planar surface.

2) Coverage planning: Many different approaches have been developed to address coverage path planning in 2D spaces using grid-based [10], Morse-based [11] or topological coverage [12]. These methods use cellular decomposition to represent free space, then geometrically plan a path that connects adjacent cells. This task is a variant of the popular Traveling Salesman Problem (TSP) with the condition that agents should visit the neighborhood of cities. [13] introduced a boustrophedon path planner with a TSP approach for flight trajectories covering large fields. A recent survey of robotic coverage path planning is provided in [14]. While 2D coverage planning is commonly used for visual inspection [15] on Unmanned Aerial Vehicle (UAV) with onboard cameras, 3D coverage planning is necessary for contact-based monitoring.

The main challenge in robot coverage planning comes from the complexity or the large size of workspaces. Addressing reachability, [6] developed a mobile manipulator system to spray paint on pre-defined large convex surfaces. The method poses an error to the motion planner and replans if the trajectory is infeasible given the robot base pose. To inspect a large 3D structure, [15] solves a multi-UAV coverage path planning by combining Set Covering Problem and Vehicle Routing Problem. The method models coverage planning with an integer linear programming formulation to minimize the maximum length of each UAV path. Though targeting coverage planning on multiple separate surfaces, our pipeline can also generate a complete coverage trajectory on complex and large 3D objects by considering all surfaces on all sides.

3) Loco-manipulation control: One important factor of mobile manipulator inspection tasks is how to ensure that robots can strictly follow the required end-effector trajectories. Apart from planning the arm, motion planners must compute the robot base's location for achieving desired poses stably. Often, locomotion and manipulation are treated separately in controlling these models due to their complicated dynamics. Inverse reachability maps are introduced in determining base placements by analyzing the reachability and dexterity of manipulators. While collision is generally excluded in this method [16], [17], [18] use inverse Dynamic Reachability Maps [19] for optimal base placement and continuous scene monitoring and replanning to robustly accommodate dynamic changes. Sampling-based approaches [20], on the other hand, can produce collision-free motion planning in dynamic environments. Optimization is an effective tool that has been developed to calculate either base placement [6], [21] or whole-body control signals [22], [23] in loco-manipulation systems, which utilize the manipulation capabilities of mobile robots. While contact forces with the environment during manipulation tasks can cause instability in legged torsos, [24] handled such external disturbances using a trajectory optimization method and optimized whole-body loco-manipulation plans for robustness from unknown and known disturbance directions. The motion planner considers the full dynamics of the platform, therefore, can effectively plan when robot feet break contact with the environment. In [25], the authors introduced a whole-body optimization framework on a wheeled manipulator that targets task-specific constraints. The algorithm generates motion plans from the integrated inputs including state estimation, perception and users. Our work extends this concept to versatile legged mobile manipulators and includes multi-surface coverage path planning, perception, segmentation, and whole-body trajectory planning.

B. Contribution

This paper introduces a novel optimization approach to solving end-to-end loco-manipulation. Our framework integrates environment sensing, coverage path planning, whole-body trajectory planning and execution, allowing the robot to automatically perform coverage path following over multiple surfaces specified from the operator. We effectively combine 3D point cloud reconstruction and segmentation into a perception model to provide environmental awareness, which takes inputs from an RGB-D camera and a user interface. The pipeline includes a two-layer hierarchical structure of optimization: mission planning and motion planning. The former optimizes surface sequencing with a mixed-integer linear programming (MILP) model. Meanwhile, for motion planning, our work utilizes one-step and trajectory optimization algorithms. These are used to compute whole-body motion plans and joint control signals for the robot to follow the generated Cartesian trajectory. Our floating planar-base approach to the torso makes it possible for the system to be applied on either legged or wheeled mobile platforms. The full system proved to be universal and scaleable to a range of coverage path applications.

II. Problem formulation

In this section, we discuss three main components of our work: Perception, Mission planning and Motion planning.

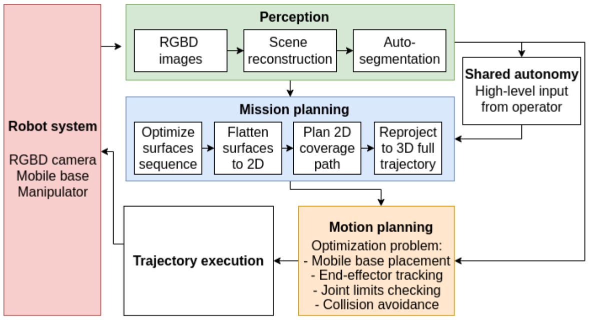


Fig. 2: Design of the proposed perceptive locomanipulation system in multi-surface inspection.

The full design is described in Fig. 2, which computes a complete trajectory over surfaces without any prior information. The goal of our work is to solve an asymptotically optimal coverage strategy over multiple random surfaces for mobile manipulators. The planning is done offline with a static environment that is unknown beforehand and is perceived on the spot. Details of each component are discussed below.

A. Perception

Our perception pipeline includes two main features: reconstruction and segmentation. The module takes input from an onboard camera and displays auto-segmented environmental objects in a user interface. Figure 3 summarizes how our perception pipeline reconstructs and segments the environment to extract visual information.

1) Reconstruction: We use an RGB-D camera on the arm's end-effector to sense the environment, which is also the only sensor used in the system. The robot is commanded to scan the scene from different angles at a fixed location. We reconstruct the environment using Truncated Signed Distance Function (TSDF) volume integration [26] with multiple input frames collected from raw RGB-D data. The method takes transformations of the camera with respect to the world coordinate to compute signed distance and utilizes viewing angles for reconstructing surface normals. The reconstructed point cloud is then downsampled with voxels to reduce processing time in later stages. Due to the noise from the camera and robot transformation during whole-body movement, we add a radius outlier filtering layer that removes all invalid points to finalize our environment point cloud. Fig. 3b shows the reconstructed point cloud from RGB-D camera scanning the environment in Fig. 3a.

2) Segmentation: To assist users in selecting a surface to scan, we segment objects from the reconstructed point cloud by combining Random Sample Consensus (RANSAC) [27] and Euclidean clustering [28]. These methods do not require any known models, which is suitable for processing unstructured environments like remote industrial sites for which no reliable prior map may exist and rapid adaptation using surface scanning is necessary. RANSAC is well-known for its plane segmentation algorithm that determines a plane model by calculating distance error with a hypothesis plane. At first, we use plane segmentation to remove the ground/table and wall from the point cloud. The objects are then clustered based on density with Euclidean algorithm, which can help to segment arbitrary shapes. Segments are then colored and visualized in a user interface (Fig. 3c) and operators can either choose a segmented object or select a custom surface with a bounding box. Users are allowed to specify one or more surfaces for multi-surface planning.

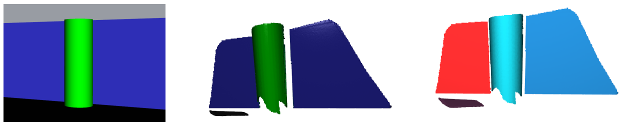

(a) Environment (b) Reconstructed (c) Segmented

Fig. 3: An example of our perception pipeline.

B. Mission planning

1) Sequence planning: The surface sequencing is formulated as a mixed-integer linear programming problem and is an extension of integer linear programming in TSP. Taking each surface as a city in TSP, the problem is similar in the sense that we should find an optimal path through all surfaces and each surface should be visited only once. However, in our case, each surface is configured as a polygon instead of node in maps. Hence, our formulation also includes continuous variables, which are start and end positions on each surface, apart from the integer decision for surface sequencing. Our model will optimize the sequence of surfaces to be scanned in order to minimize energy. The coverage path within each surface does not affect this cost. A task sequence parameterization vector is defined for the decision, which contains the surface, start and end poses on that surface.
Equation (1) models the planning problem with $N(>= 1)$ surfaces where x_{ij} is a binary decision (2) at step i of surface j, which equals to 1 if surface j is to be scanned at step i. S and G are the start and end position matrices (3). $S[,i]$ and $G[,i]$ is the Cartesian start and end positions of the robot's end-effector on the surface-to-scan at step i, which is surface j if $x_{ij} = 1$. The objective function $d()$ is the weighted Euclidean distance between end poses on the previous surface and start poses on the later surface. Constraint (4) and (5) ensures there is only one surface scanned per step and all surfaces are scanned only once. Finally, constraint (6) limits the end-effector start and end poses within the boundary of the surface at that step.

$$\underset{x,S,G}{min} \sum_{i=1}^{N} \mathrm{d}(S[,i], G[,i-1]) \tag{1}$$

$$\text{s.t. } \forall_{i,j \in 0,..,N-1} : x_{ij} \in \{0,1\} \tag{2}$$

$$S, G = \mathbb{R}^{3\times N} \tag{3}$$

$$\forall_{i \in 0,..,N-1} : \sum_{j=0}^{N-1} x_{ij} = 1 \tag{4}$$

$$\forall_{j \in 0,..,N-1} : \sum_{i=0}^{N-1} x_{ij} = 1 \tag{5}$$

$$\forall_{x_{ij}=1} : S[,i], G[,i] \in \mathbf{B}_j \tag{6}$$

2) Coverage path planning: With each surface on the plan, the system implements the path planner from [13] to autonomously generate a coverage path. This planner processes the coverage path on a 2D surface. Therefore, we need to flatten the 3D point cloud cropped by the Open3D [29] point selector to a polygon. The points in 3D are mapped into a flat 2D space by projecting adjacent mesh vertices into the plane of the current one [30]. We keep a reference of where the points in 2D have come from to allow for an easier transformation of the path back into 3D. The 2D polygon is fed to the coverage path planner along with the specified width of the sensor. Then, we re-project the generated coverage path into 3D with the saved reference. Randomizing positions are added to the computed 2D coverage path to ensure a smooth trajectory after re-projection.

C. Motion planning

We compute motion planning for the whole robot system, which includes both mobile base and manipulator, instead of treating them separately. This approach allows base placement generation integrated in the algorithm and maximizes dexterity in manipulation tasks. We formulate the whole-body motion planning as a nonlinear optimization problem using the Extensible Optimization Toolkit (EXOTica) [31]. The planning scene is defined with a complete loco-manipulation robot model, where the torso is treated as a floating-base system. In our work, the mobile platform is holonomic and does not have any restrictions on translation or rotation, however, these could be added if required. The state of the robot with N-DoF manipulator is described as:

$$x = [q_{base}, q_{manipulator}] \tag{7}$$

$$\text{where } q_{base} = [x_{base}, y_{base}, yaw_{base}] \tag{8}$$

$$q_{manipulator} = [q_1, q_2, q_3, ..., q_N] \tag{9}$$

In this study, we compare two methods of optimization: one-step and trajectory. Details of the algorithms are discussed below.

1) One-step optimization: The whole-body one-step optimization is defined as a quadratic cost problem:

$$\arg\min_{x} f(x)^T Q f(x) \tag{10}$$

with cost function:

$$f(x) = \sum_{i} \rho_i \left\| \Phi_i(x) - y_i^* \right\| \tag{11}$$

where ρ_i is the weight of task i, Φ_i is the mapping function of the task map and y_i^* is the goal reference. The robot is asked to reach desired arm's end-effector position and orientation in the task space generated by the path planning module. The 6-DoF robot arm is reduced to a 5-DoF system where the final joint is ignored since this yaw value does not affect the arm end-effector tracking. We employ the generalized inverse kinematics (IK) solver to compute joint position signals with minimized error with the reference poses, i.e. a multi-objective optimization over a desired set of tasks. Joint limits are checked using the robot model definition in the planning scene. This algorithm computes a single configuration for robot joints at each step on the trajectory and continues to the next stage until the whole trajectory is completed. The optimization loop continuously updates the current state of the robot as the initial state for the next step's optimization problem to avoid local minima and smooth the robot movement. The cost of the optimization problem is used to automatically enable base position constraint, aiming at minimizing base movement and improving scanning time. Accordingly, if the cost is below a pre-defined threshold, the problem will transform to a fixed-base manipulator reaching desired end-effector poses task.

2) Trajectory optimization: For the trajectory optimization, we model the whole-body control for a legged mobile manipulator as a one-step look-ahead nonlinear optimization in time configuration space with constraints:

$$\arg\min_{x,u} f(x,u,t), \tag{12}$$

$$\text{s.t. } h(x,u,t) = 0, \tag{13}$$

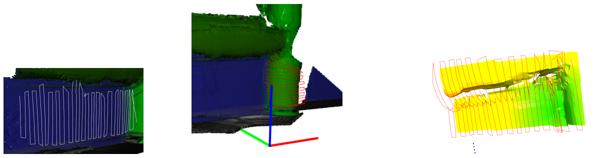
$$g(x,u,t) \leq 0, \tag{14}$$

where u is the control signal. Given the desired end-effector trajectory, equality constraints $h(x,u,t)$ are set as the arm's end-effector position and orientation goal. Meanwhile, inequality constraints $g(x,u,t)$ place limits on joint position and velocity. The trajectory optimization method plans the control signals, considering the next state on the computed path, which ensures smooth movement between steps. We also penalize base movement along the trajectory. We use variable number of timesteps for the trajectory optimization, which is found in the computed path from the path planning module. Instead of generating separate joint positions for each step, this algorithm computes a complete time-indexed joint acceleration trajectory. We use second-order derivatives on cost functions to obtain smooth transitions during horizon planning. A Differential Dynamic Programming (DDP) solver is applied to solve the trajectory optimization, which is efficient in handling timing constraints in the control loop. As the problem is highly nonlinear, this solver is not guaranteed to converge and needs a specified maximum iteration. We relax the constraints during approaching the trajectory to allow the robot more time to settle in the initial step and follow the path afterwards.

III. Evaluation

A. Single-surface planning

This section validates the coverage path planning adaptability on different surface types and their coverage quality. Fig. 4 captures three experiments to validate our work of planning coverage paths on random surfaces. The distance from the surface to the paths is adjustable, which allows different applications for monitoring with visual sensors as well as other, e.g. contact-based, modalities.



(a) Flat surface (b) Cylinder (c) Non-flat surface

Fig. 4: Coverage path planning results on different surfaces. The surfaces have been automatically reconstructed and captured using our method prior to coverage path planning.

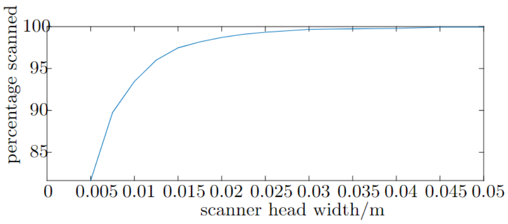


Fig. 5: Coverage results of the path planning algorithm over sensor head width on a flat surface.

In all three experiments, flat contours that are fairly closely aligned are generated. The regions were selected by the user to inspect coverage on different parts. Figures 4a and 4b show the system working on flat and curved surfaces respectively. The path around the vertical cylinder is slightly distorted at the top but this will still allow full coverage. Figure 4c shows an attempt of path planning over such a surface (where the corner formed by the two cylinders and the rear surface is the problematic area). There is a large amount of distortion at that corner, but the algorithm performs better than expected across the remaining portion of the surface.

We evaluated the coverage percentage of the path planning module by looking at adjacent contours in sequence. For each pair of contours, a triangular mesh is generated. This enables us to calculate both the total area to scan, which is the sum of the areas of all the triangles, and the area that the scanner passes over per triangle, which is a trapezium at the bottom and approximated as a triangle at the apex of the triangle. Despite this approximation, it should be suitable for the case where the contours are roughly parallel to each other, which is the case here.

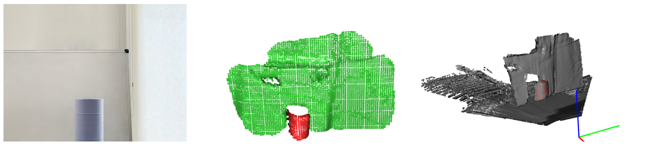
Fig. 5 plots the percentage of the chosen surface scanned against the scanner head width. The contours in these tests are all supposedly $0.03\,\mathrm{m}$ away from each other, i.e. the size of each pixel in 2D space. We can see that the algorithm produces a very high percentage coverage for this width and those above it, as well as doing well on those slightly below it. As the scan quality decreases, the percentage of the surface that is not covered increases. This is due to small kinks in the path, and fewer points at the edges of each scan. This stops the edges of the scan from properly aligning, thus creating a few imperfections in the edges.

Note that Fig. 5 was done using a set of passes over a flat surface, as shown in Fig. 4a. This produced a coverage percentage of $99.6\,\%$. For the cylinders in Figure 4b, we get $96.0\,\%$ coverage. This is due to a slightly higher variance in the spacing of the contours themselves, which in itself is less due to the curvature and has more to do with the length of each contour over the surface. With this percentage of coverage, any flaws in the material that are of finite size are likely to be detected by the system.

We also deployed our perception pipeline with data collected from a physical environment. Fig. 6 shows that our work can successfully reconstruct, segment and plan a coverage path on noisy data in the real world.

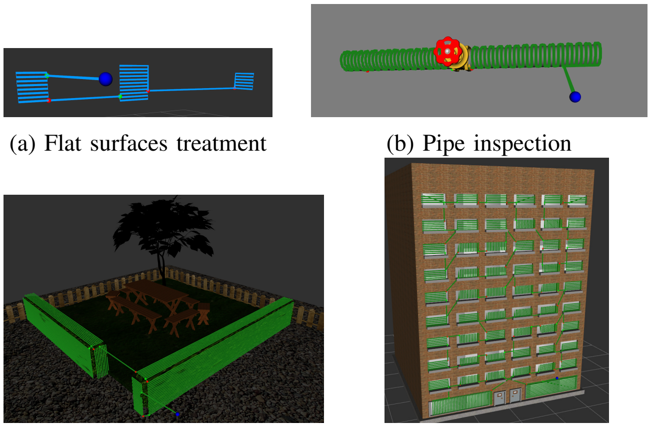
B. Multi-surface mission planning

A set of scenarios are selected to show how our mission planning is universal to many coverage planning applications, as visualized in Fig. 7. In these figures, the blue dot is the current robot's end-effector position while green and red ones represent the computed start and end positions on surfaces. The complete trajectory in Cartesian space, which includes both the surface sequence and coverage paths on all surfaces, is denoted by the blue/green line. This path is the output of the mission planning optimization pipeline. We use Gurobi [32] to solve the optimization problem.



(a) Real world (b) Auto-segmented (c) Coverage path

Fig. 6: Perceptive module tested on real environmental data.



(a) Flat surfaces treatment (b) Pipe inspection

(c) Hedge trimming (d) Windows cleaning

Fig. 7: Set of experiments highlighting applicability of our developed method to a variety of multi-surface coverage path planning tasks.

Fig. 7a describes a simple scenario with three separate flat surfaces, which is applicable in material sanding or wall painting. Meanwhile, Fig. 7b shows that our pipeline can be used where the coverage planning is obstructed by a valve and has to follow a cylindrical surface posing challenges for whole-body motion planning, which is typical in pipe inspection tasks. These are different types of 3D surfaces that our system can plan coverage on. Fig. 7c is a hedge-trimming scenario where coverage paths can be planned in different directions. This is an example where the coverage path has to scale to a large 3D structure. Our mobile legged-manipulator system is also suitable for this task as the hedge can be on rough or rocky terrains. Finally, Fig. 7d demonstrates the scalability of the mission planning framework with 56 surfaces in a window cleaning task. Our framework can be extended to consider penalizing motions in different directions (up/down against left/right) for real-world objective functions – even though our legged robot system is not suitable for this scenario, the developed method applies to any floating-base manipulation system, e.g. similar to Skyline Robotics.Our set of experiments can serve as a benchmark for measuring multi-surface coverage path planning for loco-manipulation.

TABLE I: Evaluation of multi-surface mission planning.

Number of surfaces	Planning time (s)	Optimality gap (%)
2	0.021	0
3	0.007	0
5	0.015	0
10	0.23	0
15	16.856	0
20	1000	38.86
25	1000	68.7
30	1000	74.73

Table I shows the results of our multi-surface mission planning over the number of surfaces. We set time limit 1000 s for the planning time. The optimality gap is the expected gap to optimal solution, which is computed by:

$$\frac{|\text{ObjectiveBound} - \text{IncumbentObjectiveValue}|}{|\text{IncumbentObjectiveValue}|}$$

The result shows that our pipeline can obtain globally optimal plans on up to 15 surfaces within 17 s. Longer time limits should be taken to reach optimality with larger state spaces, e.g. the 56-surface window cleaning in Fig. 7d.

C. Pipeline integration

We validated our fully integrated system with one-step and trajectory optimization, and provide a comparison between the two approaches. The mobile manipulator used in both experiments is a lightweight 6-DOF Kinova robot arm mounted on an Anymal C quadruped, as shown in Fig. 1b. We attach an Intel Realsense D435 depth camera to the manipulator's wrist for scene reconstruction and visual sensing.

The testing scenario is a multi-surface inspection task where the robot is asked to scan a region automatically with input surface from the operator. The full pipeline of our system includes the following steps:

1) Scan the surface from a location with different angles.
2) Process collected RGB-D images: reconstruct the surface, filter noises, segment objects.
3) Ask the operator to select one or more objects or custom surfaces to scan.
4) Plan the full coverage path over all selected surfaces.
5) Solve the whole-body optimization problem to get joint positions control signal.
6) Execute the joint trajectory.

Aiming at comparing performance of the two optimization methods, steps 1-4 remain the same while we switch between the one-step and the trajectory mode. The planned coverage trajectory from step 4 is automatically discretized to a number of desired poses midway. The one-step optimization plans control signals for each step on the path. Meanwhile, trajectory planning method defines time-steps needed to perform each pose on the path.

The robot first moves to the desired base pose, then the manipulator is controlled to reach waypoint target positions and orientations accordingly.

In Fig. 8, we compare task costs, which include position, angle alignment and joint limit violation, over steps in the two optimization methods with the same trajectory. All task weights are set the same respectively in both modes. The results show that it takes time for the trajectory planning method to settle on the coverage path, but have better alignment with significantly lower cost at some poses in later steps. Notably, no joint limits are violated in the trajectory optimization. In general, the one-step approach produces more consistent costs along the coverage path in comparison with trajectory optimization. This suggests that separating approach timing from trajectory following could lead to optimal solutions.

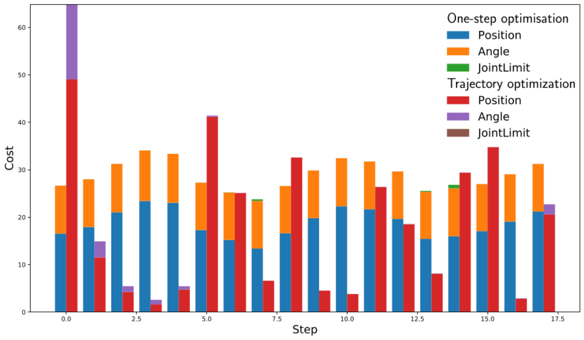


Fig. 8: Detailed task cost comparison between one-step and trajectory optimization.

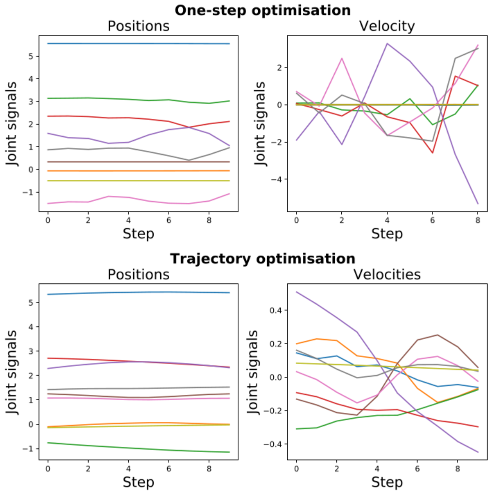
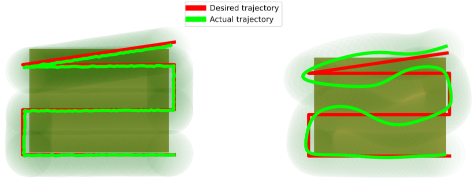


Fig. 9: Joint states comparison between one-step and trajectory optimization.

Fig. 9 depicts the joint states computed from the one-step and trajectory optimization over timesteps. The 9 lines in each subplot represent control signals for our 9-DOF system (3-DOF signal for the base and the rest for the arm). It is clear that joint positions are generally stable in both modes. However, the one-step optimization has sightly more fluctuation in these signals compared to the trajectory approach. In conclusion, the trajectory planning algorithm generates a smoother movement with significantly lower velocity.



(a) One-step optimization (b) Trajectory optimization

Fig. 10: Coverage result of our full pipeline.

Fig. 10 visualizes the coverage path tracking on both motion planning methods. The red lines refer to the generated trajectory from the mission planning module while the green lines denote the performed trajectory from the full pipeline integration. The inspected area is represented by the green faded padding, where the overlapped region gets a darker shade. The radius is the sensor head width sent to the coverage path planning system. In one-step optimization path tracking (Fig. 10a), the performed trajectory is almost the same as the generated one, with its padding proving that the generated trajectory ensures full coverage over the inspected region. The results show that our system successfully follows the coverage trajectory and the pipeline efficiently minimizes overlapping, which optimizes energy in inspection tasks. In the plots, there are observed small gaps that are not covered due to the absolute radius (without overlapping) that we set in the coverage path planning, which can be decreased to compensate errors from robot motion planning and control.

Supplementary material and videos on our experimental evaluation are available at `https://sites.google.com/view/multisurface-coverage-planning`

IV. Discussion

We have proposed an end-to-end optimization-based framework for loco-manipulation systems to perform coverage path planning in inspection and monitoring applications. Our main contribution is the introduction of a two-layer optimization structure for the fully integrated system, where the coverage mission planning also optimizes the start and end positions on the surfaces. Additionally, the perception system automatically reconstructs and segments the environment scene, which intuitively helps users in selecting the inspected surface. Along with that, we developed a full-coverage path planning method that successfully unrolls 3D meshes from a point cloud, generates a coverage 2D path and projects it back to the surface. The pipeline can be universally applied on multiple scenarios and can scale to large state spaces with longer processing time. The set of surface sequencing experiments can also be used as a benchmark for multi-surface coverage planning. Two optimization techniques were evaluated for motion planning, one-step and trajectory optimization planning. Both approaches consider robot reachability, alignment, collision avoidance and capability of base relocation. The results showed that the trajectory optimization produces a smoother movement but with less consistency in poses alignment than the one-step approach. Therefore, the one-step motion solution was preferred if strict requirement of path following (e.g. contact-based applications) was in place. Though, better tuning of trajectory optimization might help this method to produce similar or better alignment with a lower overall cost.

In the future, we would like to integrate the path cost from the coverage path planner into the mixed-integer mission planning objective function. The problem could then be solved with a black-box optimization solver, e.g., [33]. In addition, the proposed framework currently assumes that motion planning can always find a feasible solution that follows the computed Cartesian path resulting from mission planning. Therefore, further development can also involve informing surface sequencing with geometry feasibility and reachability evaluation.

References

[1] V. Ivan, *et al.*, "Autonomous Non-Destructive Remote Robotic Inspection of Offshore Assets," in *Offshore Technology Conference*, 2020.

[2] S. M. Ahmadi, *et al.*, "Constrained coverage path planning: evolutionary and classical approaches," *Robotica*, vol. 36, no. 6, 2018.

[3] D. Jia, *et al.*, "Coverage path planning for legged robots in unknown environments," in *IEEE SSRR*, 2016.

[4] D. Khudher, "Continuous Landmines Scanning using Legged Robot with Manipulator Arm over Rough Terrain," in *IEEE AIM*, 2017.

[5] I. A. Hameed, "Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain," *J. of Intelligent & Robotic Systems*, vol. 74, no. 3-4, 2014.

[6] N. Dhanaraj, *et al.*, "A Mobile Manipulator System for Accurate and Efficient Spraying on Large Surfaces," in *Int. Conf. on Industry 4.0 and Smart Manufacturing*, 2022.

[7] J. E. See, *et al.*, "The Role of Visual Inspection in the 21st Century," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 61, no. 1, pp. 262–266, 2017.

[8] C. D. Lockard, "Anomaly detection in radiographic images of composite materials via crosshatch regression," Ph.D. dissertation, Mills College, 2015.

[9] C. Maierhofer, *et al.*, "Characterizing damage in CFRP structures using flash thermography in reflection and transmission configurations," *Composites Part B: Engineering*, vol. 57, pp. 35–46, 2014.

[10] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1, 2001.

[11] E. U. Acar, *et al.*, "Morse decompositions for coverage tasks," *IJRR*, vol. 21, no. 4, pp. 331–344, 2002.

[12] S. Wong and B. MacDonald, "A topological coverage algorithm for mobile robots," in *IEEE IROS*, 2003.

[13] R. Bähnemann, *et al.*, "Revisiting boustrophedon coverage path planning as a generalized traveling salesman problem," in *Field and service robotics*. Springer, 2021, pp. 277–290.

[14] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[15] W. Jing, *et al.*, "Multi-uav coverage path planning for the inspection of large and complex structures," in *IEEE IROS*, 2020.

[16] A. Makhal and A. Goins, "Reuleaux: Robot Base Placement by Reachability Analysis," in *IEEE IRC*, 01 2018, pp. 137–142.

[17] J. Dong and J. C. Trinkle, "Orientation-based reachability map for robot base placement," in *IEEE IROS*, 2015.

[18] W. Merkt, *et al.*, "Robust shared autonomy for mobile manipulation with continuous scene monitoring," in *IEEE CASE*, 2017.

[19] Y. Yang, *et al.*, "iDRM: Humanoid motion planning with realtime end-pose selection in complex environments," in *IEEE Humanoids*, 2016.

[20] ——, "Planning in Time-Configuration Space for Efficient Pick-and-Place in Non-Static Environments with Temporal Constraints," in *IEEE Humanoids*, 2018.

[21] Q. Yu, *et al.*, "Base position optimization for mobile painting robot manipulators with multiple constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 56–64, 2018.

[22] C. D. Bellicoso, *et al.*, "ALMA - Articulated Locomotion and Manipulation for a Torque-Controllable Robot," in *IEEE ICRA*, 2019.

[23] P. Ewen, *et al.*, "Generating Continuous Motion and Force Plans in Real-Time for Legged Mobile Manipulation," in *IEEE ICRA*, 2021.

[24] H. Ferrolho, *et al.*, "RoLoMa: Robust Loco-Manipulation for Quadruped Robots with Arms," *arXiv:2203.01446*, 2022.

[25] W. Merkt, *et al.*, "Towards shared autonomy applications using whole-body control formulations of locomanipulation," in *IEEE CASE*, 2019.

[26] Q.-Y. Zhou and V. Koltun, "Dense Scene Reconstruction with Points of Interest," *ACM Transactions on Graphics*, vol. 32, 07 2013.

[27] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. ACM*, jun 1981.

[28] M. Ester, *et al.*, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Int. Conf. on Knowledge Discovery and Data Mining*. AAAI Press, 1996.

[29] Q.-Y. Zhou, *et al.*, "Open3D: A Modern Library for 3D Data Processing," *arXiv:1801.09847*, 2018.

[30] L. Fraser, "How to flatten 3D object surface into 2D array?" 2017.

[31] V. Ivan, *et al.*, *EXOTica: An Extensible Optimization Toolset for Prototyping and Benchmarking Motion Planning and Control*. Springer, 2019.

[32] B. Bixby, "The Gurobi optimizer," *Transp. Research Part B*, 2007.

[33] R. Hamano, *et al.*, "CMA-ES with Margin: Lower-Bounding Marginal Probability for Mixed-Integer Black-Box Optimization," *arXiv:2205.13482*, 2022.