Autonomous navigation of quadruped robots for monitoring and inspection of civil infrastructure

Aditya Tandon, Jan Stührenberg, Kosmas Dragos, and Kay Smarsly

Institute of Digital and Autonomous Construction, Hamburg University of Technology, Germany
aditya.tandon@tuhh.de

Abstract

Mobile robots have increasingly been gaining recognition in monitoring and inspection of civil infrastructure, owing to their potential to improve efficiency and accuracy. Specifically, quadruped robots offer enhanced stability and adaptability, rendering the robots ideal candidates for automated monitoring and inspection. To enable quadruped robots to conduct monitoring and inspection of civil infrastructure, the robots must be capable of autonomous navigation. Existing approaches towards autonomous navigation usually incorporate joint-state information to plan motions with whole-body controllers, representing a non-linear, high-dimensional problem that is computationally expensive to solve and is a particular burden when implemented into the robots for real-time navigation. This paper presents an integrated architecture for automated monitoring and inspection that consists (i) of a mission planning framework devised for planning monitoring and inspection tasks and (ii) a robust motion planning framework to enable real-time navigation. Specifically, the motion planning problem is decoupled into high-level task-space and low-level joint-space components, and the motion planning framework, building upon the "Cartographer" simultaneous localization and mapping algorithm, combines the "batch-informed trees" and A* algorithm for global planning and the "timed elastic band" for local planning and obstacle avoidance. For validation, the integrated architecture for automated monitoring and inspection is implemented into quadruped robots, and validation tests are conducted in indoor office environments to be inspected. As a result, it is demonstrated that the quadruped robots navigate safely and collision-free in real time, accommodating both static and dynamic obstacles. Enhancing the efficiency and accuracy of autonomous navigation of quadruped robots in complex and dynamic environments, the architecture is expected to pave the way for future research in robust controller development and 3D-state space planning.

Keywords: Quadruped robots, motion planning, mission planning, batch-informed trees, timed elastic band.

1 Introduction

Degradation of civil infrastructure poses significant risks to public safety. Civil infrastructure therefore requires continuous monitoring and regular inspections to ensure safe operation [1]. Existing manual monitoring and inspection approaches may be time-consuming, labor-intensive, expensive, and dangerous. With labor shortages, rising labor costs, and low productivity in the construction industry [2], monitoring and inspection may become more expensive. Adopting robot-based, automated monitoring and inspection has the potential to enhance productivity, reduce costs, and increase safety [3].

Mobile robots, equipped with sensors, have already been used for monitoring and inspections [4]. However, previous approaches have been dependent on personnel with special training, such as building inspectors and robot operators. Recent advancements in robotics and computing have fueled approaches using mobile robots, including unmanned aerial vehicles and unmanned ground vehicles, in autonomous configurations to perform visual inspections [5]. Unmanned aerial vehicles may, however, be restricted in range and operational flight time because of limited payload and battery capacity. Wheeled or tracked unmanned ground vehicles may be less constrained than unmanned aerial vehicles regarding operation time or payload capacity, but are limited by the terrain. The traversability of construction environments is challenging, as wheeled or tracked robots may be unable to climb staircases, access multi-floor workspaces, or climb over obstacles of varying sizes, such as concrete blocks, pipes, or stacks of construction materials. In contrast to unmanned ground vehicles, legged robots, particularly quadruped robots, can perform complex maneuvers, overcome obstacles, and climb staircases. Quadruped robots outperform both unmanned ground vehicles and unmanned aerial vehicles due to their enhanced mobility, allowing the quadruped robots to navigate difficult terrain and carry heavy payloads. However, to achieve robot-based automated monitoring and inspection, quadruped robots must be capable of reliable autonomous operation.

To enable reliable autonomous operation of quadruped robots, motion planning frameworks capable of real-time environment perception, localization, path planning, obstacle avoidance, and motion control are typically utilized. The goal of motion planning frameworks is to generate robot joint controls that move the robots on collision-free trajectories to target positions. Motion planning for mobile robots has been a subject of comprehensive research [6]. The choice of motion planning frameworks primarily depends on the complexity of tasks to be accomplished and the complexity of the operational environment. The motion planning framework developed in [7] has used a combination of a global planning module that operates on a 3D volumetric map, and a local planning module for navigation and robot traversal. The motion planning framework NeBula, developed in [8], operates across several robot types, including unmanned ground vehicles, unmanned aerial vehicles, and quadruped robots. The NeBula framework includes components for state estimation, mapping, traversability, planning, and communication, and is inclined towards providing platform-agnostic autonomous navigation capabilities. In [9], an autonomous navigation framework, consisting of a locomotion control module, a state estimation module, and a planning module, has been deployed on the MIT Cheetah Vision quadruped robot in outdoor environments. The locomotion module uses a regularized predictive control scheme with a whole-body controller (WBC), which is supplemented by information on robot joint angles and environmental characteristics perceived by the robots. The planning module utilizes an A*-based global planning module and a gait scheduler for high-level planning. However, motion planning is performed in the joint space. In [10], a coupled motion planning method that plans WBC motions consisting of foothold locations and horizontal motions has been presented and deployed on a quadruped robot. The WBC optimizes motion across the quadruped robot while taking into consideration center-of-mass (CoM) locations, foothold locations, robot attitude, and kinodynamic constraints to generate joint torques for robot control.

Existing motion planning frameworks for quadruped robots incorporate information on robot joint states and terrain that may be used by advanced planning components within the framework, such as foothold contact planning [11] and reachability checking [7], enabling comprehensive motion planning in the joint space. However, the motion problem caused by using advanced motion planning components may be high-dimensional and non-linear, rendering the problem complex and computationally expensive to solve in real-time, particularly on resource-constrained computing devices deployed on quadruped robots. Although WBC approaches have demonstrated promising results [12] on quadruped robots, WBC approaches are computationally complex, with this complexity being hardly justified in standard indoor environments. While outdoor environments may be uncertain, uneven, and complex, standard indoor environments are typically characterized by continuous environments that do not contain large discontinuities. In standard indoor environments without large discontinuities, computation of precise foothold locations may be less crucial, and instead, a gaited strategy that maintains the balance of the robot may be sufficient.

The motion planning frameworks mentioned above are usually either simplistic, with limited scaling with the size and complexity of the environment, or complex, involving motion planning in the joint space. On the one hand, simplistic motion planning frameworks may fail to model the complexity of the robot and the environment; on the other hand, complex motion planning frameworks may attempt to incorporate most aspects of the motion problem, becoming computationally expensive to solve in a reasonable time and inhibiting real-time performance on resource-constrained computing devices deployed on quadruped robots. This paper proposes an integrated architecture for automated monitoring and inspection that includes a motion planning framework suitable for automated monitoring and inspection. The motion planning framework decouples the motion planning problem into two distinct components, (i) a high-level task-space problem, and (ii) a low-level joint-space problem, to ensure effective real-time operation of quadruped robots. The motion planning framework operates through high-frequency communication between the two components and utilizes the high-level interface for robot control, thereby abstracting the low-level control of the robots. In addition to the motion planning framework, a mission planning framework, the second integral part of the integrated architecture, is introduced to enable planning and execution of monitoring and inspection tasks. The mission planning framework provides an integrated task planning system, used to schedule various types of monitoring, inspection, and navigation tasks.

The remainder of the paper is outlined as follows. Section 2 describes the integrated architecture for automated monitoring and inspection. First, the design and implementation of the motion planning framework is covered, followed by the design and implementation of the mission planning framework. Section 3 presents the tests conducted to validate the integrated architecture as well as the results of the validation tests. Finally, Section 4 concludes with a discussion of the results, a summary of the work presented herein, and an outlook on potential future work.

2 An integrated architecture for automated monitoring and inspection

The integrated architecture for automated monitoring and inspection, including the motion planning framework and mission planning framework, is shown in Figure 1. The software setup builds upon a distributed Robot Operating System (ROS) architecture that allows modules to be distributed across robots and computers. Offloading modules to computers reduces the computational load of the robots and facilitates real-time operation. Therefore, in this study, the motion planning framework is implemented on a robot and the mission planning framework, which is not critical for real-time operation, is implemented on a computer. In the following subsections, the motion planning framework and the mission planning framework of the integrated architecture are presented.

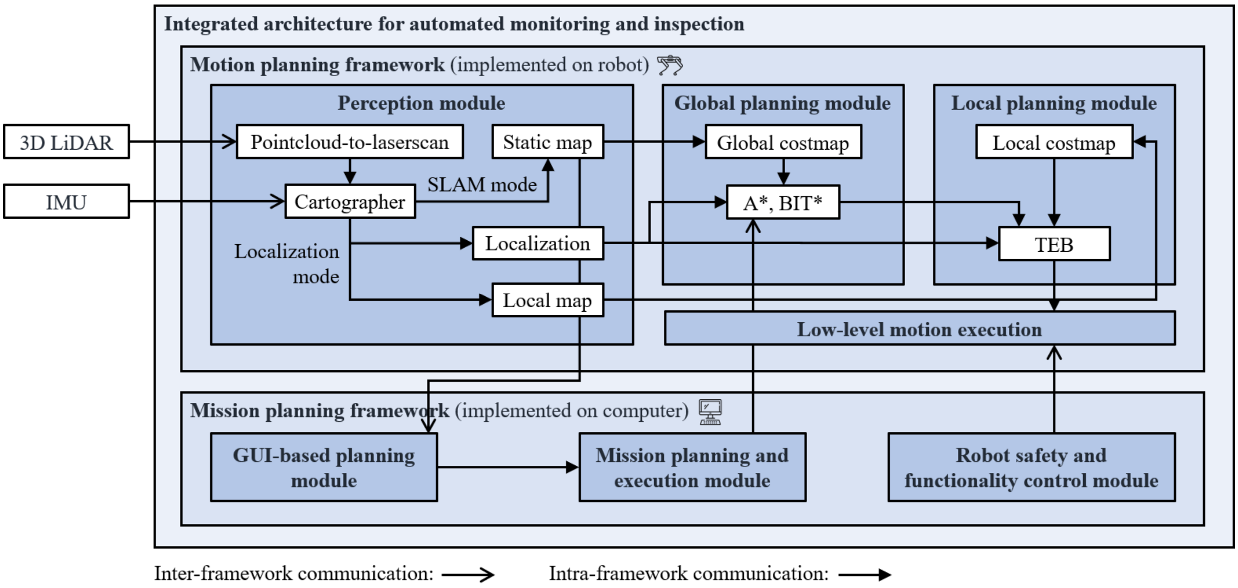


Fig. 1. Integrated architecture for automated monitoring and inspection.

2.1 Motion planning framework

Since robots operate in a dynamic world, reliable real-time performance of the modules of the motion planning framework must be ensured to avoid collisions. To reduce the computational complexity, the motion planning framework is decoupled into a high-level task-space problem and a low-level joint-space problem. Decoupling the problems results in two problems with reduced complexity, which can be solved independently. The high-level task-space problem is concerned with planning collision-free paths for the robot and to generate target velocities in the task space. Using high-frequency communication, the low-level motion execution employs a gaited strategy to fulfill the target velocities while maintaining robot balance and stability. Since high-level motion planning using full 3D representations in real-time on embedded platforms remains computationally prohibitive [5], a hybrid strategy that approximates the 3D environment in a 2D format is adopted. High-level motion planning in the task space is therefore achieved by representing the quadruped robot as a holonomic robot in the SE(2) state space with a rectangular robot footprint, representing the 2D shape of the physical space occupied by the robot in the environment. Thus, the development of a generalized and modular motion planning framework is achieved. By decoupling the motion planning framework, the low-level controller can operate at high frequencies, ensuring robust and stable locomotion in uneven terrain and in the presence of external disturbances. The ROS navigation stack, used as a basis for the implementation, processes navigation goals and utilizes static and dynamic occupancy grid maps to facilitate autonomous navigation, where a global planning module plans paths globally in the static occupancy grid map and a local planning module plans paths locally in the dynamic occupancy grid to avoid obstacles. In the following paragraphs, the core modules of the framework are presented.

Perception module. Motion planning requires perception of the environment to generate maps of the environment, to localize within the maps, and to detect dynamic obstacles. In this study, perception is granted by a light detection and ranging (LiDAR) sensor, which is the primary sensor employed within the perception module of the motion planning framework, complemented by an inertial measurement unit (IMU). Since motion planning and navigation is performed in the SE(2) state space, 2D laser scan data is employed for perception. However, 2D laser scan data offers only a 2D planar snapshot of the environment at the operational height of the LiDAR and does not effectively capture spatial information in 3D. To leverage 3D spatial information about the environment, 3D point cloud data is converted to 2D laser scan data by projecting the 3D geometry of the environment onto the horizontal 2D plane using the ROS pointcloud_to_laserscan package [13]. The IMU senses fine changes in the attitude and state of the robot to align LiDAR scan data with greater accuracy, aiding the localization and mapping processes, detailed below.

To generate a map of the environment and localize within the map, the simultaneous localization and mapping (SLAM) system "Cartographer" is used [14]. The Cartographer system leverages the 2D laser scan data and the IMU data collected by the perception module. The Cartographer system can be operated in two modes, (i) a SLAM mode, and (ii) a localization mode. The SLAM mode allows generating a static map of the environment, with the robot being controlled manually by an operator to cover the area of interest. The map is stored as an occupancy grid map. Occupancy grid maps represent the environment with a map consisting of an evenly-spaced discrete grid, where each cell represents either occupied or free space for navigation. Once a static map has been generated using the SLAM mode, the localization mode of the Cartographer system is used to localize the robot in the static map.

Global planning module. The global planning module is a critical subset of the motion planning framework for enabling robots to determine optimal trajectories to navigate environments. Specifically, the global planning module identifies sequences of intermediate states that guide the robot from its initial state to a goal state. By searching through the space of all possible states, the so-called "configuration space", the global planning module plans feasible, safe, and optimal paths, by checking against potential collisions with obstacles in the static map and minimizing distance, time, and/or energy. For searching, graph-based or sampling-based path planning algorithms are employed.

The global planning module implements the *nav_core::BaseGlobalPlanner* interface prescribed in the ROS navigation stack. Various path planning algorithms are implemented within the global planning module to enable optimum path planning performance across different operating environments. The *NavfnROS* implementation of the A* graph-based planning algorithm is utilized. A* operates on the discretized and static occupancy grid map. Given an admissible heuristic, A* guarantees an optimal solution. ROS-compatible C++ implementations of sampling-based planning algorithms are utilized from the *OmplPlanner* global planning module [15], based on the Open Motion Planning Library (OMPL) [16]. Implementations of the rapidly-exploring random trees (RRT*) and Informed-RRT* planning algorithms are used as-is, while the *OmplPlanner* is extended to include the batch-informed trees (BIT*) planning algorithm. Sampling-based path planning algorithms operate by randomly sampling the configuration space and constructing a graph by connecting points with edges if they are collision-free. The sampling-based path planning algorithms implemented in this study exhibit asymptotic optimality, i.e. the probability of converging to a solution increases to one as the search time increases. The algorithms are particularly effective for high-dimensional configuration, with runtime independent from the dimension of the configuration space.

Local planning module. The local planning module of the motion planning framework operates on the coarse path planned by the global planning module. To account for obstacles, particularly dynamic obstacles and obstacles that may have been untracked during mapping, the local planning adapts the path plans for a short time window in the immediate local environment of the robot at high frequencies to facilitate collision-free navigation, while ensuring plans that are generated by the module are executable by the robot controller. The local planning module utilizes sensor data received from the perception module to construct a local map centered around the robot. Local maps enable perception of the immediate environment, track dynamic and static obstacles, and allow the local planning module to generate motion commands and trajectories for collision-free navigation. To ensure minimum distances between robots and obstacles, costmaps are employed. Costmaps supplement the occupancy grid maps by associating travel costs to neighboring occupied cells.

The local planning module uses the timed elastic bands (TEB) approach for local planning [17]. The TEB planning approach distorts existing global plans by counter-balancing virtual attraction forces of target poses and repulsive forces of obstacles. The TEB planning approach generates optimized trajectories with respect to execution time while complying with kinodynamic constraints and maintaining minimum distance when navigating around obstacles. The TEB planning approach involves the minimization of an objective function for online optimization of time-optimal trajectories, while ensuring controller actions do not violate kinodynamic bounds such as maximum velocity and acceleration. Since quadruped robots are holonomic robots and are able to move among translational axes, the TEB local planning module is configured to weaken the penalty for satisfaction of non-holonomic constraints. Thereby, trajectory generation with velocities and accelerations in all directions in the SE(2) state space is promoted.

2.2 Mission planning framework

To enable automated monitoring and inspection, a mission planning framework to schedule various types of monitoring, inspection, and navigation tasks is employed. The mission planning framework integrates a user interface to sequentially plan, execute, and administer sets of monitoring and inspection tasks, hereinafter referred to as “monitoring and inspection missions”, and administers robot-related safety and functionality. The mission planning framework is classified into three modules. First, the mission planning and execution module, followed by the GUI-based planning module, and lastly, the robot safety and functionality control module.

The **mission planning and execution module** models monitoring and inspection missions as hierarchical finite state machines. In particular, mission tasks are modelled as individual states, and transitions between states are defined based on the execution status of the respective tasks. The mission planning and execution module is implemented using the SMACH library in the ROS framework [18]. One waypoint is added for each state to the SMACH state-machine object, and transitions between states are created based on the successful execution status of the states, indicated by successful transitions. Goals are sent to the navigation stack from within each state by implementing an action. The state-machine object waits for completion of the goal before transitioning to the next waypoint. The state machine completes execution when the last waypoint is reached by transitioning to the “final” state.

The **GUI-based planning module** provides a panel in the ROS visualization tool RViz to intuitively create monitoring and inspection missions. Waypoints are added to a monitoring and inspection mission by drawing arrows on the static occupancy grid map created in the SLAM mode of Cartographer. The origin of the arrows corresponds to the target position of the waypoints and the direction of the arrows corresponds to the target heading of the waypoints. Furthermore, the GUI-based planning module allows saving and loading monitoring and inspection missions.

Last, but not least, the **robot safety and functionality control module** is devised to provide the operator with an integrated interface to view and administer control over the operation and motion status of the robot. The operator may pause the execution of a mission by pausing the operation of the robot, ensuring appropriate manual intervention during mission execution.

3 Validation tests and results

The integrated architecture for automated monitoring and inspection proposed in this study is validated in an indoor environment to evaluate functionality and real-time performance. The validation tests are threefold. First, the perception module of the motion planning framework is validated by creating a static occupancy grid map of the environment using the SLAM mode of Cartographer and testing the localization in the static occupancy grid map using the localization mode of Cartographer. Second, the mission planning framework is validated by planning and executing monitoring and inspection missions in the static map. The monitoring and inspection missions are designed as multi-waypoint missions to lastly validate the global planning module and the local planning module of the motion planning framework.

The motion planning framework is implemented into a quadruped robot of type “Intelligent Documentation Gadget” (IDOG), as described in [4]. The validation tests are conducted at the Institute of Digital and Autonomous Construction (IDAC) at Hamburg University of Technology, Germany. The indoor environment features a range of furniture, consisting of large tables, chairs, workstations, small cabinets, and shelves. Notably, movable furniture, such as cabinets and chairs, may alter the spatial layout to a minor extent, thereby constituting static obstacles in the environment. People walking in the office space, not captured during mapping, are considered dynamic obstacles in the environment.

The following subsections provide an overview of the validation tests performed and the results obtained, starting with the perception module, followed by the global planning module, and finally, the local planning module for autonomous navigation capabilities.

3.1 Perception module

To validate the perception module, Cartographer is used to create a static map of the environment and subsequently to localize the robot in the static map. The LiDAR is operated at a frequency of 10 Hz and the IMU at 100 Hz. The IDOG is manually controlled, while the SLAM mode of Cartographer creates the occupancy grid map at a resolution of 0.05 m/pixel. The created occupancy grid map, shown in Figure 2, is approximately 780 pixels in length and 400 pixels in width, corresponding to a map size of 39 m × 20 m. After starting the localization mode of Cartographer, the initially unknown location of the IDOG in the map is determined within seconds.

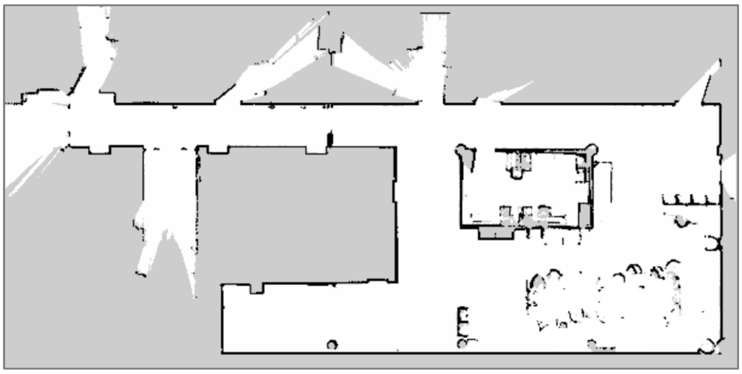


Fig. 2. Occupancy grid map of an indoor environment created by the SLAM mode of Cartographer.

3.2 Global planning

To validate the global path planning module, the mission planning framework is used to create monitoring and inspection missions set up as multi-waypoint missions. The global planning module of the motion planning framework plans paths between the waypoints. The paths are compared based on path length and path planning time. Three challenging waypoints are selected as shown in Figure 3. The first waypoint simulates the challenges encountered when planning long – yet feasible – paths in short time intervals between waypoints that are far apart. The second waypoint simulates the complexities in navigating through intricate environments with large obstacles, narrow corridors, and around corners. The third waypoint involves traversing a narrow, cluttered space with numerous static obstacles. Validation is performed for the BIT*, Informed-RRT*, and RRT* algorithms of the OMPL planner and for the A* algorithm of the NavfnROS planner.

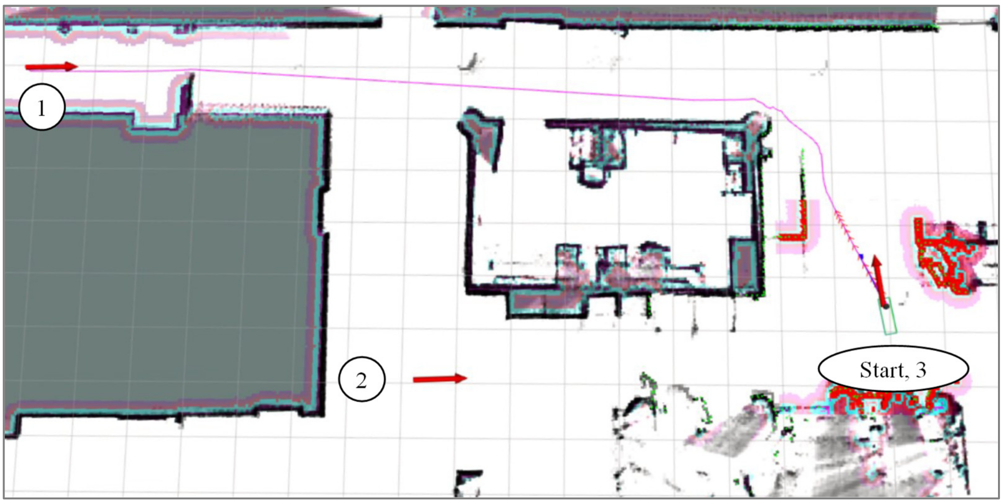


Fig. 3. Multi-waypoint mission, created using the mission planning framework to validate the global planning module.

The validation tests demonstrate that the global planning module is capable of planning valid paths in the indoor environment. All global planning algorithms find a path between the start position and the first waypoint. However, the path produced by RRT* is considered invalid due to its coarse trajectory through cluttered environments, resulting in collisions with furniture. This observation, supported by a comparison of optimal path costs, results in abandoning further path planning tests with RRT*. The paths planned by A*, BIT*, and Informed-RRT* between waypoints 1, 2, and 3 have comparable lengths. However, the key distinction lies in planning time, as shown in Table 1. The A*-based planner demonstrates a planning time an order of magnitude lower than the other sampling-based planning algorithms. Despite a 1-second planning budget for the sampling-based planning algorithms, path lengths generated by the sampling-based planning algorithms exceed the length of the path generated by the A* planning algorithm. The results from the validation tests for the global planning module are tabulated in Table 1.

Table 1. Path-length and planning-time of the global planning algorithms

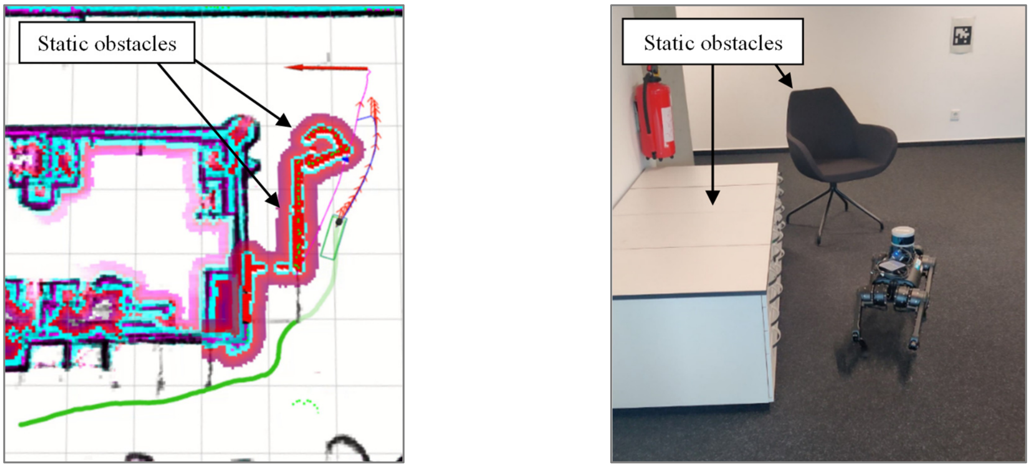
Waypoint	A*		BIT*		Informed-RRT*		RRT*	
	Length (m)	Time (s)	Length (m)	Time (s)	Length (m)	Time (s)	Length (m)	Time (s)
1	18.878	0.082	18.795	1.037	19.732	1.035	26.747	1.034
2	11.544	0.288	11.621	1.051	11.535	1.085	-	-
3	8.815	0.085	8.816	1.032	8.811	1.039	-	-

The results obtained in this study indicate that, although sampling-based planning algorithms represent the state of the art, sampling-based planning algorithms outperform traditional graph-based planning algorithms, such as A*, only in high-dimensional problems. It may also be inferred that the existing representation of the motion planning problem may be low-dimensional, since planning is performed only in 2D. In addition, it must be noted that the size of the validation environment is small, as compared to larger environments in which monitoring and inspection are performed, such as construction or industrial sites. To facilitate future path planning in 3D and to enable operation in large environments, the BIT* planner is integrated into the motion planning framework in addition to the A* algorithm. The A* algorithm will be used for planning in 2D and small environments.

3.3 Local planning

Validation of the local planning module of the motion planning framework is conducted by performing multiple tests. First, the path-tracking capability of the local planning module in the absence of obstacles is ascertained. The tests involve tracking a straight-line path generated by the global planning module and tracking a long global plan to navigate between waypoints while traversing long narrow passageways, maintaining a minimum distance from environment boundaries at all times. The validation tests for the local planning module in the absence of obstacles indicate that the actual trajectory traversed by the robot closely aligns with the global plan. Deviations from the global plan are noted, however, deviations from the global plan do not necessarily reflect poor performance of the local planning module, particularly in regions where minimal distance from environment boundaries must be maintained. However, in some segments of the plan traversed by the robot, slight deviations and a constant offset from the global plan is observed. Overall, the performance of the local planning module in the absence of obstacles is deemed satisfactory.

The functionality and real-time performance of the local planning module is validated against static and dynamic obstacles of variable shapes and sizes, which are likely to appear in indoor environments, despite the mapped environment in which robots operate. In fact, maps may be incomplete or change over time, thereby failing to accurately represent the updated furniture layout. Furniture that has been moved is representative of static obstacles in the environment. Moreover, the performance of the local planning module is also validated against dynamic obstacles, such as persons. The validation tests involve planning global paths using outdated maps of the environment. As the robot traverses the global path, static obstacles are added to the local costmap of the robot, based on sensor data received from the perception module, as illustrated in Figure 4a. It is also observed that the local planning module plans around the obstacle (blue trajectory), although the global plan is in collision with the static obstacle (purple trajectory). As can be seen from Figure 4b, the IDOG is able to successfully avoid the obstacle and complete the global plan.

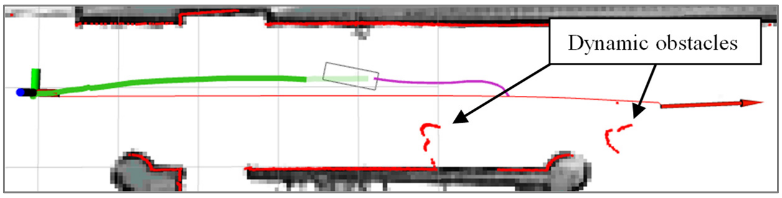


(a) Local costmap with static obstacles.

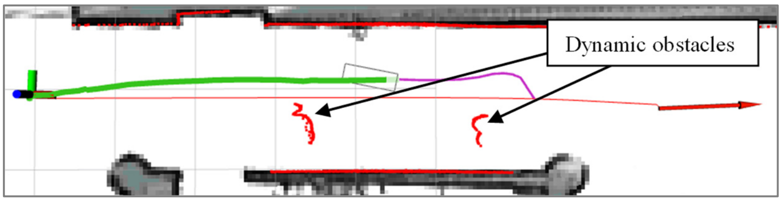
(b) Re planning around a static obstacle.

Fig. 4. Demonstration of obstacle-avoidance behaviors against static obstacles.

Furthermore, Figure 5 shows the trajectories re-planned by the motion planning framework to avoid dynamic obstacles. As can be seen from Figure 5, the purple trajectory signifies the local plan, while the red trajectory signifies the global plan. Although the IDOG exhibits reliable obstacle-avoidance behavior, the initial parts of the green trajectory indicate that an offset continues to exist between the real trajectory and the global plan.



(a) IDOG planning around the first dynamic obstacle.

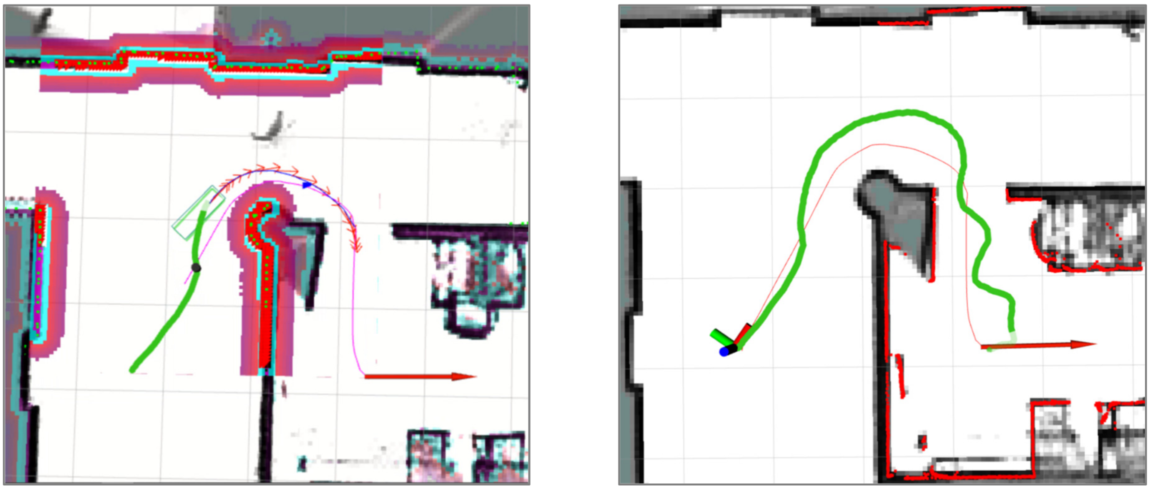


(b) IDOG planning around the second dynamic obstacle.

Fig. 5. Dynamic obstacle-avoidance behavior to avoid multiple persons (dynamic obstacles).

To validate the autonomous navigation capability of the motion planning framework deployed on the IDOG, specific waypoints are selected for navigation through restricted passages. In the tests, waypoints are placed within an office room, requiring traversal through a door passage with a width of 1.05 m, and inside a laboratory room in the office environment, requiring navigation through a door that is 0.8 m wide. The tests aim to ascertain behaviors when navigating around obstacles and through narrow confined spaces simultaneously and necessitate effective global planning as well as robust local planning to adhere to obstacle clearances and to prevent getting stuck in a local minimum.

The validation tests for autonomous navigation demonstrate that the IDOG is able to successfully plan a global path, fine-tune the path locally and navigate in real time, while maintaining enough distance from the boundaries of the door passages of width 1.05 m and 0.8 m, respectively. While the autonomous navigation capabilities of the motion planning framework are satisfactory around obstacles and through narrow pathways, in confined spaces, an oscillatory behavior is observed between the path followed by the IDOG and the global plan (Figure 6); nonetheless, the IDOG recovers and completes the global plan. The origin of the oscillatory behavior requires further investigation to improve stability of autonomous navigation.



(a) Local plan around the corner.

(b) Executed trajectories around the corner.

Fig. 6. Execution of a local plan in confined spaces and around corners.

4 Summary and conclusions

The degradation of civil infrastructure requires continuous monitoring and regular inspections to ensure safe operation, which can be performed by quadruped robots. However, to achieve robot-based automated monitoring and inspection, it is essential that quadruped robots possess the ability to operate autonomously, which requires motion planning frameworks capable of real-time environment perception, localization, path planning, obstacle avoidance, and motion control. The motion planning framework, presented in this study, decouples the motion planning problem into two distinct components, a high-level task-space problem and a low-level joint-space problem for the real-time operation of quadruped robots. The tree-based global planning algorithm BIT*, the A* algorithm, and the timed elastic band local planning approach are proposed for high-level robot control and obstacle avoidance.

The *integrated architecture for automated monitoring and inspection* proposed in this study has been deployed on the IDOG quadruped robot and a computer, using a distributed software architecture to enable high-level mission planning on the computer and provide autonomous navigation for the IDOG. The *mission planning framework* is deployed on the computer and uses a state machine to plan monitoring and inspection missions in an indoor office environment with static and dynamic obstacles to be completed by the IDOG. The *motion planning framework* deployed on the IDOG is validated by executing the planned monitoring and inspection missions set up as multi-waypoint missions. The results show that the motion planning framework is capable of real-time operation and successfully avoids static and dynamic obstacles. In addition, narrow doors can be traversed, although some undesirable oscillatory behavior of the robot motion is observed. In conclusion, the integrated architecture for automated monitoring and inspection provides a useful foundation for automated monitoring and inspection of civil infrastructure using quadruped robots. The mission planning framework provides a GUI to plan monitoring and inspection missions intuitively, while the motion planning framework reliably plans and controls the motion of the robot in real-time. Future work will investigate and mitigate the oscillatory behavior of the motion planning framework when traversing narrow spaces. In addition, specific monitoring and inspection tasks may be implemented and added to the mission planning framework, to further advance automated monitoring and inspection of civil infrastructure.

Acknowledgments

The authors gratefully acknowledge the support offered by the German Research Foundation (DFG) through grant SM 281/20-1. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DFG.

References

1. Smarsly, K., Law, K. H., & König, M., 2011. Autonomous structural condition monitoring based on dynamic code migration and cooperative information processing in wireless sensor networks. In: Proceedings of the 8th International Workshop on Structural Health Monitoring 2011. Stanford, CA, USA, 09/13/2011.
2. Woetzel, J., Mischke, J., Barbosa, F., Ribeirinho, M. J., Sridhar, M., Parsons, M., Bertram, N., & Brown, S., 2017. Reinventing construction through a productivity revolution, Accessed 03/05/2023, available at: https://www.mckinsey.com/capabilities/operations/our-insights/reinventing-construction-through-a-productivity-revolution.
3. Smarsly, K., & Dragos, K., 2024. Advancing civil infrastructure assessment through robotic fleets. Internet of Things and Cyber-Physical Systems, 4(2024), pp. 138-140.
4. Smarsly, K., Dragos, K., Stührenberg, J., & Worm, M., 2023. Mobile structural health monitoring based on legged robots. Infrastructures, 8(9), 136.
5. Halder, S. & Afsari, K., 2023. Robots in inspection and monitoring of buildings and infrastructure: A systematic review. Applied Sciences, 13(4), 2304.
6. Sánchez-Ibáñez, J. R., Pérez-del Pulgar, C. J., & García-Cerezo, A., 2021. Path planning for autonomous mobile robots: A review. Sensors, 21(23), 7898.
7. Wellhausen, L., & Hutter, M., 2022. Artplanner: Robust legged robot navigation in the field. Field Robotics, 3(1), pp. 413-434, 2023.
8. Morrell, B., et al., 2022. Nebula: Team costar's robotic autonomy solution that won phase ii of darpa subterranean challenge. Field robotics 2, pp. 1432-1506.
9. Dudzik, T., Chignoli, M., Bledt, G., Lim, B., Miller, A., Kim, D., & Kim, S., 2020. Robust autonomous navigation of a small-scale quadruped robot in real-world environments. In: Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems. Las Vegas, NV, USA, 10/25/2020.
10. Mastalli, C., Havoutis, I., Focchi, M., Caldwell, D. G., & Semini, C., 2020. Motion planning for quadrupedal locomotion: Coupled planning, terrain mapping, and whole-body control. IEEE Transactions on Robotics, 36(6), pp. 1635-1648.
11. Stumpf, A., & von Stryk, O., 2022. A universal footstep planning methodology for continuous walking in challenging terrain applicable to different types of legged robots. In: Proceedings of the 2022 International Conference on Robotics and Automation. Philadelphia, PA, USA, 5/23/2022.
12. Liu, M., Qu, D., Xu, F., Zou, F., Di, P., & Tang, C., 2019. Quadrupedal robots whole-body motion control based on centroidal momentum dynamics. Applied Sciences, 9(7), 1335.
13. ros-perception, 2023. pointcloud_to_laserscan. Accessed 12/11/2023, available at https://github.com/ros-perception/pointcloud_to_laserscan.
14. Hess, W., Kohler, D., Rapp, H., & Andor, D., 2016. Real-time loop closure in 2D lidar slam. In: Proceedings of the 2016 International Conference on Robotics and Automation. Stockholm, Sweden, 5/16/2016.
15. ETHZ-RobotX, 2023. smb_ompl_planner. Accessed 12/8/2023, available at https://github.com/ETHZ-RobotX/smb_path_planner/tree/master/smb_ompl_planner.
16. Sucan, I. A., Moll, M., & Kavraki, L. E., 2012. The open motion planning library. IEEE Robotics & Automation Magazine 19(4), pp. 72-82.
17. Rösmann, C., Feiten, W., Wösch, T., Hoffmann, F., & Bertram, T., 2012. Trajectory modification considering dynamic constraints of autonomous robots. In: Proceedings of ROBOTIK 2012; 7th German Conference on Robotics. Munich, Germany, 5/21/2012.
18. Bohren, J. & Cousins, S., 2010. The smach high-level executive. IEEE Robotics & Automation Magazine, 17(4), pp. 18-20.