Enabling Failure Recovery for On-The-Move Mobile Manipulation

Ben Burgess-Limerick[1], Chris Lehnert[1], Jürgen Leitner[2], Peter Corke[1]

***Abstract*— We present a robot base placement and control method that enables a mobile manipulator to gracefully recover from manipulation failures while performing tasks on-the-move. A mobile manipulator in motion has a limited window to complete a task, unlike when stationary where it can make repeated attempts until successful. Existing approaches to manipulation on-the-move are typically based on open-loop execution of planned trajectories which does not allow the base controller to react to manipulation failures, slowing down or stopping as required. To overcome this limitation, we present a reactive base control method that repeatedly evaluates the best base placement given the robot's current state, the immediate manipulation task, as well as the next part of a multi-step task. The result is a system that retains the reliability of traditional mobile manipulation approaches where the base comes to a stop, but leverages the performance gains available by performing manipulation on-the-move. The controller keeps the base in range of the target for as long as required to recover from manipulation failures while making as much progress as possible toward the next objective. See `benburgesslimerick.github.io/MotM-FailureRecovery` for videos of experiments.**

I. Introduction

Performing multi-step mobile manipulation tasks, such as pick-and-place tasks, with the mobile base in motion has been shown to be an effective means of reducing execution time [1]. However, performing tasks on-the-move introduces a new challenge for recovering from failures. In particular, if the manipulation is not completed while the robot transits past the target, the robot will be unable to recover without incurring a significant time cost as it returns to the target.

Existing approaches to manipulation on-the-move are typically based on open-loop execution of a planned trajectory, where a single attempt at the task is planned [1]–[6]. If any failure occurs, such as a poorly executed grasp, a new trajectory is required, which is likely to require significant planning time.

A close analogue to this problem is grasping an object from a moving conveyor, where there is a limited window for completing the grasp. Existing approaches to conveyor grasping typically do not have time to recover from a failed grasp [7]–[9]. A mobile manipulator introduces the possibility to reactively adjust the base motion to ensure that sufficient time is available to complete a manipulation task, while still minimising execution time in multi-step tasks.

A recent reactive approach [10] to manipulation on-the-move has been presented which allows for online repeat attempts in the event of a manipulation failure. In [10], a simple base placement and control system is implemented which does not consider how the base should be best controlled to allow sufficient time for the manipulator to recover from the failure, while also completing the task as quickly as possible.

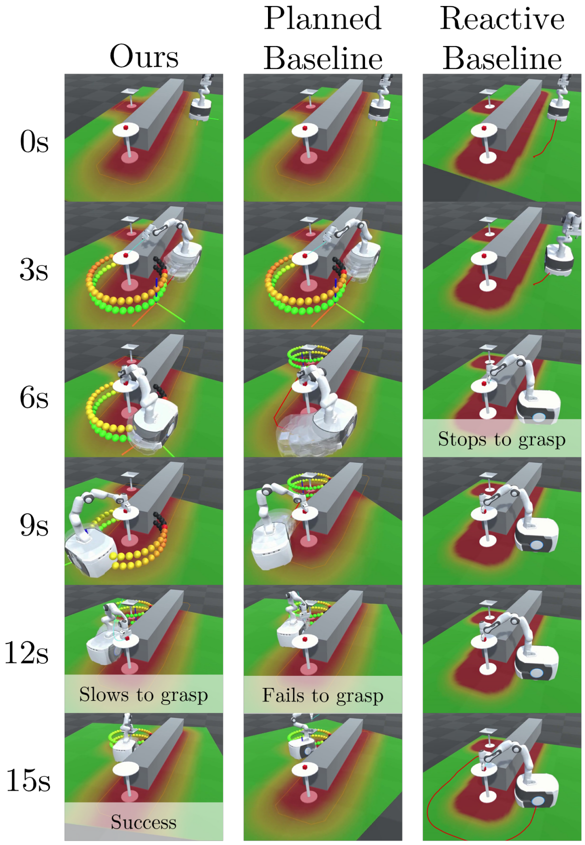


Fig. 1: Frames from video of each method performing a pick-and-place task with a grasp failure delay of 6 s.

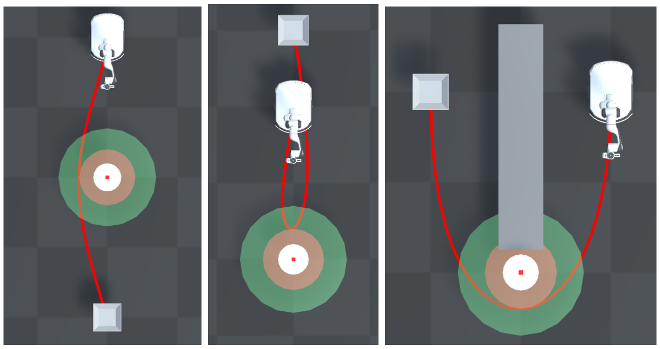
In this work we present a reactive base control system that integrates with the architecture presented in [10] and allows for failure recovery during manipulations performed on-the-move, while also minimising task execution time. The method is investigated in simulated pick-and-place experiments and compared with two baseline approaches representing existing reactive approaches to mobile manipulation and more recent on-the-move methods (Fig. 1).

Our results demonstrate the effectiveness of the proposed approach in handling failures and reducing execution time in multi-step mobile manipulation tasks. The proposed method outperforms traditional approaches to mobile manipulation and more recent on-the-move methods in terms of task completion rate and execution time.

This research was supported by the QUT Centre for Robotics.

[1] Ben Burgess-Limerick, Chris Lehnert, and Peter Corke are with the Queensland University of Technology Centre for Robotics (QCR), Brisbane, Australia `ben.burgesslimerick@qut.edu.au`

[2] Jürgen Leitner is with LYRO Robotics, Brisbane, Australia



(a) Line (b) Turn (c) Obstructed Turn

Fig. 2: Three example pick-and-place scenarios. The green circles represent the area in which the robot can reach the object. The red circles indicate the area in which the robot will collide with the table the object is resting on. The red paths present example trajectories the robot could transit to perform the task.

II. Problem Definition

Graceful recovery from a failure during on-the-move tasks introduces new challenges for mobile manipulator base control. In this work, we focus on pick-and-place tasks where a single object must be grasped and then dropped in a new location. To successfully recover from a failed grasp, the robot must stay within reach of the object until the grasp can be completed. However, there is often an opportunity to improve overall task time by continuing base motion toward the place location for as long as possible. For tasks where multiple objects need to be transported or there are multiple acceptable drop points, the optimal decision may be to change targets entirely. We do not consider these cases in this work.

We identify the three scenarios presented in Fig. 2 as representative of common pick-and-place tasks. These scenarios do not represent an exhaustive list, but illustrate how the arrangement of the environment impacts the amount of time the object will be within reach of the robot as it drives past. For the obstructed turn scenario, there is significant time for the robot to recover from a failed grasp while it drives around the object toward the drop point. By comparison, the turn environment presents a very limited manipulation window. To gracefully recover in all scenarios, the goal for the mobile base should be selected, and adjusted online, such that the object is kept within range until the grasp is completed. We focus only on minor manipulation failures where the grasp can be immediately reattempted. Larger failures, such as the object falling to the floor, require additional recovery behaviours.

III. Related Works

The base-placement problem for mobile manipulation determines a collision-free pose for the robot base from which to perform a manipulation task [11]. Approaches typically aim to generate solutions that are optimal against some other metric such as manipulability [12], [13], or stiffness of the robot [14]. Other approaches aim to minimise task time by calculating base poses from which multiple targets can be reached without repositioning the base between tasks [15], [16]. Time efficiency can be optimised on multi-step tasks by choosing a pose based on where the robot must go after the immediate target [17], [18]. Reactivity can be achieved by frequently recomputing the optimal base pose [17].

Although the system presented in [17] selects optimal base placements considering both the immediate and next target in order to reduce overall execution time for pick-and-place tasks, it does not explore further performance improvements possible by performing the pick and place actions while the robot is still in transit. Instead, the base comes to a stop while the robot performs the manipulation. A benefit of this approach is that in the event of a grasp failure it is trivial for the arm to reset and reattempt the task. By comparison, methods that explore manipulation on-the-move including those presented in [1]–[6] cannot easily reattempt a task in the event of a failure because the robot may be driving away and out of reach of the object.

We wish to unify the benefits of both on-the-move and stationary grasping approaches. In particular, we desire a system that will perform tasks on-the-move where possible, but slow down or stop when a manipulation failure requires increased time near a target.

IV. Approach

We divide the base control system into two components: a base placement module and a base controller.

A. Base Control

The base controller is a modified version of the Short Term Aborting A* (STAA*) reactive mobile robot controller presented in [19]. This system plans global paths with an A* search through a visibility graph, and generates the optimal path to a given pose, avoiding obstacles in the scene. This global path is intersected with a local grid around the robot to provide an intermediate target for the base. The system generates a plan towards the intermediate target which is aborted either when the goal is reached, or a timer expires. By aborting the search and enforcing an update frequency the system achieves real-time reactive control.

We have introduced several modifications to the controller presented in [19] to improve its performance in a manipulation on-the-move scenario.

1) Goal Orientation: The most important addition to STAA* is the inclusion of an orientation to the goal state. Where STAA* considers driving to a point only, we include orientation which enables poses to be achieved that smoothly connect the immediate target with the next goal.

2) Rotation in Global Planner Cost: The addition of orientation to the goal state requires modification to the node cost computation used in the global A* search. STAA* uses only the cumulative distance between nodes along a path to compute the cost. Instead, we consider a PathRTR metric that estimates the time required to both translate and rotate between nodes along a path. Further explanation of the PathRTR metric is presented in [19]. Fig. 3 illustrates the value of including rotation costs in the global path planner.

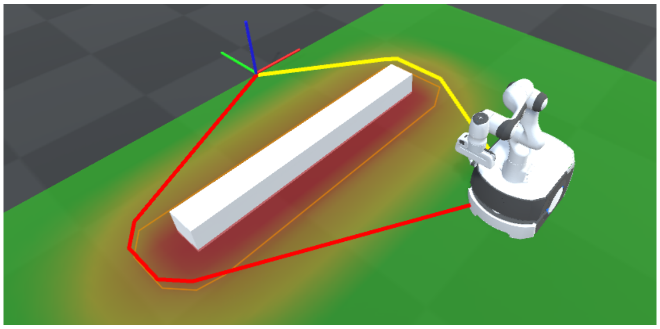


Fig. 3: A comparison of global paths generated by our modified STAA* (path in red) and the original (shown in yellow) for an example goal pose. The red x axis of the target frame represents the desired forward direction.

For the scenario shown, the path in red generated by our modified version encourages the robot to drive a smooth curve around the obstacle connecting the start and end pose. Without rotation costs the shortest path passes the obstacle on the other side and requires significantly more turning.

3) Search Termination Conditions: The implementation of STAA* presented in [19] terminates its search when an explored node is sufficiently close to the goal. To perform manipulation on-the-move, we want to encourage the robot to drive through the current target at high speed. Therefore, we also terminate when the path between a node and its parent passes sufficiently close to the goal.

4) Reduction of Proximity Grid Penalty: STAA* includes a penalty on nodes based on their proximity to obstacles using an inflated occupancy grid. However, to complete mobile manipulation tasks such as picking and placing objects from a table, the robot must necessarily travel close to the table while interacting with objects on it. For example, in Fig. 4, the occupancy grid is represented by the colour of the ground around the robot, with green representing free space, and red representing occupied space. In this case, the target pose for the base from which it can perform the grasp is in a region with moderate proximity. The penalty applied to nodes near obstacles inhibits the exploration of states near the goal. To limit this effect, we reduce the weight of the proximity grid cost based on the expected time to the goal. We scale the grid penalty by $k = \max(0.1, \min(t_h/3, 1))$ where t_h is the estimated time until the goal is achieved.

B. Base Placement

The optimal base placement is selected from a discretised set by evaluating the path cost both from the current robot pose to candidate base placements as well as from the candidates to the drop point. Candidates are evenly spaced around the target object in 10° increments on a circle of radius 0.6 m, for a total of 36 possible base positions. Each position is assigned two possible orientations, the robot's forward vector can be tangential to the circle facing either clockwise or counter-clockwise which gives a total of 72 candidates. In Fig. 4, each coloured sphere represents a candidate, with those in the top ring possessing a counter-clockwise heading, and those in the bottom facing clockwise.

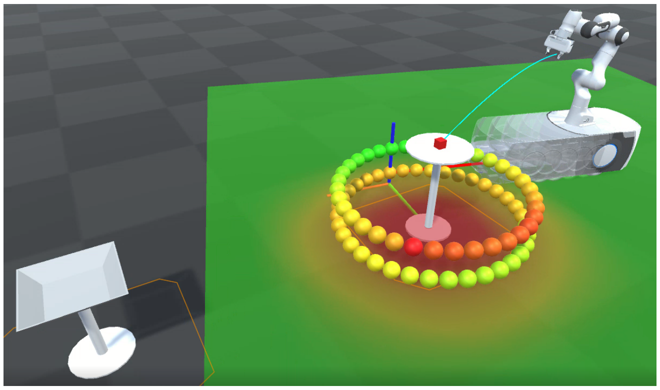


Fig. 4: Illustration of sampled base placement results for a pick-and-place task. Each coloured sphere represents a candidate base placement around the target object. Those in the top ring represent a counter-clockwise heading, while those in the bottom face clockwise.

The path cost for each candidate is evaluated using the global planner from the base control system. The highest-ranked pose from the candidate set is used as the goal for navigation. The result is a method that selects a candidate base placement that is within manipulation range of the target and efficiently connects the current robot pose with the drop pose. For example, the spheres in Fig. 4 are coloured based on their path cost, with green representing the lowest cost. The selected base pose (shown by a target frame) encourages the robot to travel counter-clockwise around the object driving towards the drop point while it completes the grasp.

To enable failure recovery, the desired base placement should be continually updated in response to the current state of the robot. If more time is required to recover from a failed grasp, the system can then reevaluate the optimal base pose which is within manipulation range of the object, but as close as possible to the drop point. We evaluate each of the candidates on every controller step (20 Hz).

V. Baselines

We compare our method with two baselines that represent common approaches to mobile manipulation. All methods share the arm control method presented in [10].

A. Conventional Reactive

The first baseline is representative of several reactive controllers for mobile manipulation such as those presented in [20]–[23]. In this baseline, the base placement method proposes the collision-free pose within manipulation range of the object that is closest to the robot's current position. The base is controlled to this pose using the Timed Elastic Band implementation provided by the ROS Navigation stack[3] which provides reactive control and obstacle avoidance.

B. Planned On-The-Move

The planned baseline is similar to the proposed method but does not recompute the optimal base pose. Effectively, the base follows a precomputed path from the initial state

[3] https://wiki.ros.org/navigation

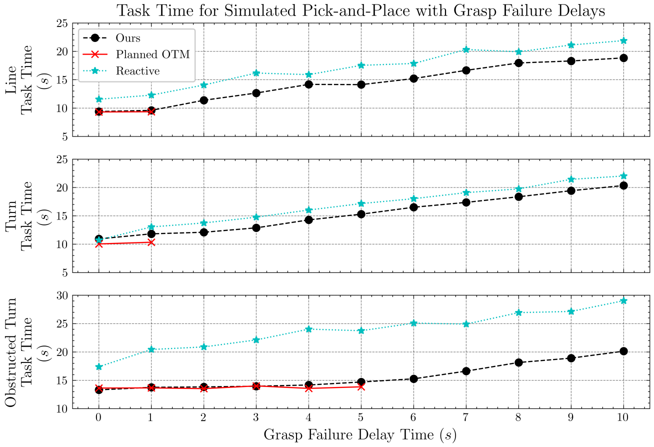


Fig. 5: Results for simulated pick-and-place tasks with varying grasp delays in three different scenarios.

to the drop point. This is representative of planning-based on-the-move approaches such as [1]–[6], [24]. It should be noted that these approaches use a planned trajectory for the arm motion which does not allow for repeat grasp attempts.

VI. Experiments

Experiments are conducted with the three arrangements of pick-and-place locations presented in Fig. 2. Failures were simulated by artificially causing grasp attempts to fail for a set amount of time after the first grasp attempt is made. This allows for investigation of the expected performance when performing tasks of varying complexity, where it may take longer to recover from failures in more challenging tasks. We test each system in each scenario with grasp failure delays between 0 and 10 seconds, where a failure delay of 0 seconds represents success on the first attempt. For each trial we record the total task execution time from the start of the robot motion to the dropping of the object.

VII. Results

Fig. 5 presents the total task execution time for each of the methods in each scenario with varied delays caused by grasp failures. Results are only shown for successful task completions, and the lack of results for the planning baseline at increased failure delay times is indicative of the robot failing to grasp the object while driving past.

In all cases, the on-the-move methods complete the task in less time than the reactive baseline. This is expected, however, it is interesting to note that the magnitude of the improvement is dependent on the task. For example, in the turn task (Fig. 2b), the base placements and subsequent paths taken by the reactive controller are relatively similar to those generated by our method because the optimal path involves driving straight to the target, turning around, and driving straight back. In this case, there is little opportunity for the on-the-move methods to make significant progress toward the next target while performing the manipulation. By contrast, in the task with an obstacle presented in Fig. 2c the robot can drive a significant distance toward the drop point while the object remains within reach.

The execution times for the reactive baseline increase linearly with increasing failure delay time. This is expected because the robot comes to a stop while it attempts to grasp the object and stays stationary until the grasp is completed. The robot can wait indefinitely providing the necessary time to recover from any failure, and consequently, it is successful in all trials.

The planned on-the-move baseline performs similarly to our proposed method when the initially calculated optimal path stays within the manipulation range for longer than the failure delay time. However, for the tasks presented in Fig. 2a and Fig. 2b, this is only the case up to a delay of 1 second. Longer failures result in the robot driving away from the object before the grasp has been completed. For the obstructed turn task (Fig 2c) the path naturally orbits the object for around 5 seconds. Consequently, the planned baseline can successfully complete the task with failure delay times of up to 5 seconds. However, as the delay time increases further, the robot drives away before the task is completed.

Fig. 1 presents snapshots from videos of each system performing a pick-and-place task where the robot takes 6 seconds to recover from a grasp failure. At t = 12s, the proposed method slows the robot to complete the grasp. The planned on-the-move baseline continues moving toward the drop point and is unable to recover from the grasp failure while still in range of the object, resulting in a failed task. The reactive baseline provides sufficient time near the object to recover from the failure, but the base does not make progress toward the drop point while the grasps are attempted, increasing execution time for the pick-and-place task.

The proposed method combines the benefits of both of the baselines. It performs tasks on-the-move with improved execution times when the robot can make progress toward the drop point while attempting to grasp the object. However, by updating the optimal base pose in real-time, the system compensates for delay times that exceed the typical transit time of driving past the object. Instead, the robot will slow down or come to a stop positioned as close to the drop point as possible, allowing time for the grasp controller to recover from failures, grasp the object, and complete the task.

VIII. Conclusion

Our proposed reactive base control system provides a practical solution for handling failures in mobile manipulation tasks performed on-the-move. However, many open questions remain in this space. In this work the robot has no knowledge of how long it might take to recover from a failure. Online prediction of this information could be used to further improve the system. Acceleration limits of the robot also have an important impact on the optimal solution to a given scenario. For example, a robot with a very limited acceleration might achieve improved performance on the presented scenarios by orbiting the object where possible instead of coming to a stop to complete the grasp. These questions present opportunities to further develop mobile manipulation systems that can operate efficiently in complex environments and adapt to uncertainties and failures.

References

[1] S. Thakar, L. Fang, B. Shah, and S. Gupta, "Towards time-optimal trajectory planning for pick-and-transport operation with a mobile manipulator," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 981–987.

[2] S. Thakar, P. Rajendran, A. M. Kabir, and S. K. Gupta, "Manipulator motion planning for part pickup and transport operations from a moving base," *IEEE Transactions on Automation Science and Engineering*, 2020.

[3] S. Thakar, P. Rajendran, V. Annem, A. Kabir, and S. Gupta, "Accounting for part pose estimation uncertainties during trajectory generation for part pick-up using mobile manipulators," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1329–1336.

[4] R. Colombo, F. Gennari, V. Annem, P. Rajendran, S. Thakar, L. Bascetta, and S. K. Gupta, "Parameterized model predictive control of a nonholonomic mobile manipulator: A terminal constraint-free approach," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 1437–1442.

[5] S. Zimmermann, R. Poranne, and S. Coros, "Go fetch!-dynamic grasps using Boston Dynamics Spot with external robotic arm," in *2021 IEEE International Conference on Robotics and Automation*. IEEE, 2021, pp. 4488–4494.

[6] J. Xu, Y. Domae, W. Wan, and K. Harada, "An optimization-based motion planner for a mobile manipulator to perform tasks during the motion," in *2022 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2022, pp. 519–524.

[7] S. E. Navarro, D. Weiss, D. Stogl, D. Milev, and B. Hein, "Tracking and grasping of known and unknown objects from a conveyor belt," in *ISR/Robotik 2014; 41st International Symposium on Robotics*. VDE, 2014, pp. 1–8.

[8] F. Islam, O. Salzman, A. Agarwal, and M. Likhachev, "Provably constant-time planning and replanning for real-time grasping objects off a conveyor belt," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1370–1384, 2021.

[9] I. Akinola, J. Xu, S. Song, and P. K. Allen, "Dynamic grasping with reachability and motion awareness," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2021, pp. 9422–9429.

[10] B. Burgess-Limerick, C. Lehnert, J. Leitner, and P. Corke, "An architecture for reactive mobile manipulation on-the-move," *arXiv preprint arXiv:2212.06991 (Accepted ICRA2023)*, 2022.

[11] H. Seraji, "Reachability analysis for base placement in mobile manipulators," *Journal of Robotic Systems*, vol. 12, no. 1, pp. 29–43, 1995.

[12] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1970–1975.

[13] S. Jauhri, J. Peters, and G. Chalvatzaki, "Robot learning of mobile manipulation with reachability behavior priors," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8399–8406, 2022.

[14] Q. Fan, Z. Gong, B. Tao, Y. Gao, Z. Yin, and H. Ding, "Base position optimization of mobile manipulators for machining large complex components," *Robotics and Computer-Integrated Manufacturing*, vol. 70, p. 102138, 2021.

[15] F. Paus, P. Kaiser, N. Vahrenkamp, and T. Asfour, "A combined approach for robot placement and coverage path planning for mobile manipulation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 6285–6292.

[16] J. Xu, K. Harada, W. Wan, T. Ueshiba, and Y. Domae, "Planning an efficient and robust base sequence for a mobile manipulator performing multiple pick-and-place tasks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 11 018–11 024.

[17] F. Reister, M. Grotz, and T. Asfour, "Combining navigation and manipulation costs for time-efficient robot placement in mobile manipulation tasks," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9913–9920, 2022.

[18] K. Harada, T. Tsuji, K. Kikuchi, K. Nagata, H. Onda, and Y. Kawai, "Base position planning for dual-arm mobile manipulators performing a sequence of pick-and-place tasks," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 194–201.

[19] M. Missura, A. Roychoudhury, and M. Bennewitz, "Fast-replanning motion control for non-holonomic vehicles with aborting A*," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10 267–10 274.

[20] J. Haviland, N. Sünderhauf, and P. Corke, "A holistic approach to reactive mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3122–3129, 2022.

[21] M. Logothetis, G. C. Karras, S. Heshmati-Alamdari, P. Vlantis, and K. J. Kyriakopoulos, "A model predictive control approach for vision-based object grasping via mobile manipulator," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–6.

[22] P. Arora and C. Papachristos, "Mobile manipulator robot visual servoing and guidance for dynamic target grasping," in *International Symposium on Visual Computing*. Springer, 2020, pp. 223–235.

[23] M. Spahn, B. Brito, and J. Alonso-Mora, "Coupled mobile manipulation via trajectory optimization with free space decomposition," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 12 759–12 765.

[24] W. Shan, K. Nagatani, and Y. Tanaka, "Motion planning for mobile manipulator to pick up an object while base robot's moving," in *2004 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2004, pp. 350–355.