

© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

An Approach to Base Placement for Effective Collaboration of Multiple Autonomous Industrial Robots

Mahdi Hassan¹, Dikai Liu¹, Gavin Paul¹ and Shoudong Huang¹

Abstract—There are many benefits for the deployment of multiple autonomous industrial robots to carry out a task, particularly if the robots act in a highly collaborative manner. Collaboration can be possible when each robot is able to autonomously explore the environment, localize itself, create a map of the environment and communicate with other robots. This paper presents an approach to the modeling of the collaboration problem of multiple robots determining optimal base positions and orientations in an environment by considering the team objectives and the information shared amongst the robots. It is assumed that the robots can communicate so as to share information on the environment, their operation status and their capabilities. The approach has been applied to a team of robots that are required to perform complete surface coverage tasks such as grit-blasting and spray painting in unstructured environments. Case studies of such applications are presented to demonstrate the effectiveness of the approach.

I. INTRODUCTION

Industrial robots are used to perform dirty, hazardous and repetitive works [1], which are preferably not carried out by humans. Making an industrial robot autonomous and mobile means achieving greater flexibility and traversability, and hence further helps the recent shift of paradigm from mass production to customized production [1]. By deploying an Autonomous Industrial Robot (AIR) in an unstructured environment, the goal can be to improve the productivity of the intended task and/or to reduce human exposure to potential health hazards. The environment in which the AIR operates can be complex and subject to change prior to each new deployment of the AIR. Various applications can make use of AIRs, examples include grit-blasting, spray-painting, hazardous material handling, pick-and-place of objects and providing lifting or grasping support for human operators.

In order for an AIR to perform a task autonomously, it is required to have enough intelligence to complete each component of the task individually. For example, an autonomous grit-blasting robot [2] that is used to remove rust and other debris from steel surfaces, performs tasks such as exploration and mapping [3], surface-type identification [4], surface segmentation [5], and collision-free motion planning [6] without any involvements of a human operator.

Making use of multiple AIRs can further improve productivity and capacity of the team to carry out the intended task. When the AIRs are performing any part of the intended task, such as finding appropriate base placements

or partitioning surface areas [7] for fair workload division, effective collaboration between the AIRs becomes crucial so as to achieve the team objectives such as minimal overall completion time of the task. In order for collaboration to take place effectively, an appropriate control architecture needs to be devised for the AIRs. The most commonly used control architectures [8] associated with multi-robot systems are centralized, decentralized, hierarchical, and hybrid. The decentralized approach is favored in many applications since each robot is only concerned about its local environment and does not take responsibility for the control of the other robots [8]. Moreover, to achieve the team objectives, effective communication needs to be employed between the robots to share environment and state information.

Appropriate base placement of the AIRs enables efficient collaboration amongst the AIRs. Mitsi et al. [9] developed an optimization based method that combines genetic algorithm, quasi Newton algorithm and constraints handling methods to find the optimal placement of a single manipulator by considering discrete end-effector positions. Vosniakos et al. [10] presented an approach that uses genetic algorithm and considers maximizing manipulability of a single manipulator to deal with the high accuracy required for the milling application. For better evaluation in finding the base placements of the AIRs, it has been shown to be important to incorporate performance measures such as the task-dependent index [11]. An approach is presented in this paper that extends the current works to be applicable to multiple autonomous industrial robots and that can deal with the problem of time deficiency when considering applications that require complete surface coverage such as grit-blasting and spray painting. The approach is also capable of optimizing multiple team objectives, some of which can be related to enhancing the performance of the manipulators attached to the robots, e.g. manipulability measure and torque.

The presented base placement optimization approach can be applied to a range of applications that benefit from the utilization of multiple AIRs. The approach considers a decentralized architecture where each AIR is able to individually explore and create a partial map of the environment, localize itself within the map and determine an appropriate base placement by using the team objectives and the information shared amongst the team of AIRs. It is reasonable to assume that the AIRs can communicate and share information on the environment, their operation status and their capabilities. It needs to be noted that the approach is not limited to the decentralized architecture, and small modifications can make the approach applicable to other control architectures.

This work is supported by SABRE Autonomous Solutions Pty Ltd and the Centre for Autonomous Systems (CAS) at the University of Technology Sydney, Australia.

¹ Authors are from the Centre for Autonomous Systems (CAS) at the University of Technology Sydney, Australia.

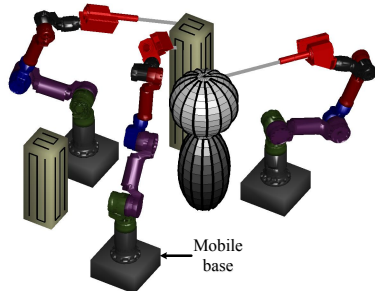
The remainder of this paper is organized as follows. Section II defines the problem. Section III presents the proposed approach and the model for effective collaboration of multiple AIRs by means of base placement optimization, and Section IV presents test results using data collected from a test rig representing a steel bridge maintenance site, and results of a simulated scenario where three robots and three objects are considered. Finally, Section V provides concluding remarks and future work.

II. PROBLEM STATEMENT

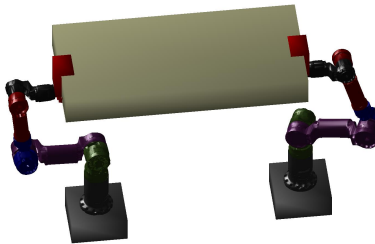
When an AIR is capable of environmental awareness, localization and mapping, situation awareness, and real time motion planning and collision avoidance, the AIR will be able to operate in unstructured, complex and dynamic environments, such as the autonomous grit-blasting robots [2], [3] currently operating in the Sydney Harbour Bridge for bridge maintenance. Generally, in industrial applications such as parts assembly, utilizing a team of robot agents can help to reduce the process cycle time and can expand the capacity of the robots to carry out a task [12].

Two examples of applications where multiple AIRs can be used are shown in Fig. 1. Grit-blasting application is shown in Fig. 1a where three AIRs are collaborating to remove rust or other debris from all surfaces of three different and separated objects. The second example application is shown in Fig. 1b where two AIRs are collaborating to move an object vertically (e.g. a flat plate) while keeping the object horizontal.

When multiple AIRs are used in such applications, each should determine an appropriate base placement both within the environment and relative to the other AIRs. Ideally, the appropriate base placement of each AIR is decided by the AIR. The AIRs communicate with each other and share



(a) Three AIRs performing grit-blasting



(b) Two AIRs lifting an object

Fig. 1: Two example applications

information on the environment and their capabilities, current position and operation status. Therefore, all the AIRs will have the same knowledge of the environment including the global frame. The objective of the AIR team, e.g. minimum makespan or completion time to cover the surfaces of the object (Fig. 1a), or moving the plate while keeping it in the horizontal by applying appropriate lifting force (Fig. 1b), is known to every AIR, and the AIR team will collaborate to achieve the objectives. Having considered the aforementioned requirements, each AIR will then conduct the optimization to select an appropriate base location and orientation prior to executing the intended task.

III. MODELLING AND APPROACH

The process for effective collaboration of multiple AIRs by means of Base Placement Optimization (BPO), i.e. finding appropriate base position and orientation for each AIR is summarized in the flowchart shown in Fig. 2.

The first component shown in the flowchart is exploration for mapping [3], [4] and localization [13], which are autonomously conducted by each AIR prior to the BPO. As a result, each AIR is able to obtain partial information about the environment based on its capabilities and location. The AIRs can be mobile (mounted on a mobile platform) when carrying out the aforementioned tasks; however, for better operation, stability and accuracy in applications such as grit-blasting and spray painting, their base is assumed to be fixed during the task execution. The AIRs are then expected to be able to communicate effectively and share environment information (component 2 of the flowchart) in order for each AIR to have a complete map of the environment. Component 2 also requires the AIRs to share status information, such as their location in the environment, and information about their capabilities, such as speed, properties of the tool attached to the end-effector and their workspace coverage. The team objectives, which are specific to the intended task, are to be known by all AIRs. Component 3 of the flowchart is for each AIR to carry out the optimization and then select the best solution (component 4).

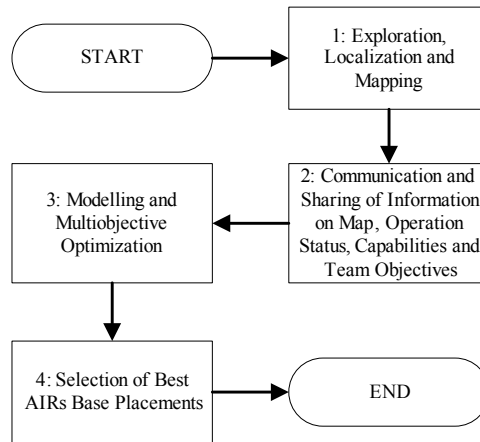


Fig. 2: Flowchart of the process for effective collaboration of multiple AIRs

A. Design Variables

Due to the large number of possible base placements for the AIRs in an environment, the presented approach considers discretizing the search space. Discretization of the search space may produce lower quality results; however in many applications compromising quality in favor of computation time is acceptable if the results are within a predefined threshold. Depending on the application, discrete base positions can be generated using coarse discretization of the search space and then further refined if predefined requirements, such as coverage, are not met.

Fig. 3 shows an example where an AIR considers generating a set of discrete base positions, $B_i = \{b_{i1}, b_{i2}, \dots, b_{im_i}\}$ on the feasible surface of the environment and around the three objects, based on its capabilities and workspace coverage, where i is the AIR index and m_i is the total number of discrete base positions for the i th AIR. From the original set of discrete base positions, each AIR should then determine a subset of discrete base positions, $(B_{FBP})_i \subset B_i$, that have a higher likelihood of producing optimal results. These base positions will henceforth be referred to as the Favored Base Positions (FBPs). Determining the FBPs is dependent on the application, the team objectives and the AIRs capabilities. For example, distance checking between each of the discrete base positions and the obstacles or the target surface can be used to prevent some of the discrete base positions from becoming candidates when they are deemed too close to the environment (shown in Fig. 3 as blue crossed points) or are anticipated to have low coverage (red filled points in Fig. 3). For each FBP, discrete base orientations are also evaluated and the orientation that provides the best result (i.e. best coverage) is selected.

Given $(B_{FBP})_i = \{\beta_{i1}, \beta_{i2}, \dots, \beta_{i(n_F)_i}\}$, $(n_F)_i \leq m_i$, the design variables are $Z_{ij} \in \{0, 1\}$, $1 \leq i \leq n$, $1 \leq j \leq (n_F)_i$ with constraints $\sum_{j=1}^{(n_F)_i} Z_{ij} = 1$ for each $1 \leq i \leq n$. Here $Z_{ij} = 1$ means that the base position $\beta_{ij} \in (B_{FBP})_i$ is selected for the i th AIR. Therefore, for the n deployed AIRs, the ultimate goal is to select n base placements that will meet the team objectives by maximizing the design objective functions in the optimization model.

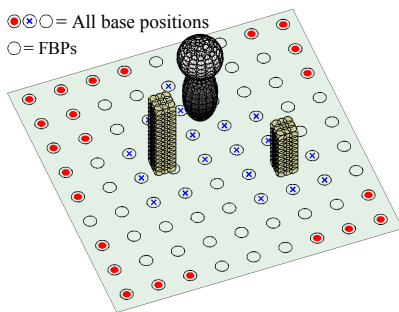


Fig. 3: Discrete base positions and Favored Base Positions (FBPs) of an AIR

B. Design Objectives

AIRs may perform tasks that require the end-effector to act in three different ways: (1) follow predefined and fixed paths during the task execution regardless of the base placement of the AIRs (e.g. paths generated on all surfaces of an object for complete surface coverage in grit blasting or spray painting applications), (2) follow paths that need to be regenerated for each new base placement of the AIRs (e.g. generating shortest path in pick-and-place tasks), and (3) remain fixed during the task execution (e.g. grasping a heavy object so a human can perform drilling or welding operations). With these tasks in mind, the main objectives can be formulated. Note that trajectory planning from the current pose of an AIR to the start point of the path is not considered in this work.

Fig. 4 shows two AIRs performing the complete surface coverage task of grit-blasting by controlling their end-effectors to follow pre-generated paths created on all surfaces of the I-beam. Discrete points, $O_i = \{o_{i1}, o_{i2}, \dots, o_{iy}\}$ are also generated on the end-effector paths of each AIR (shown in Fig. 4 as connected blue/asterisk markers) where y is the total number of discrete points on the paths of the i th AIR. It can be assumed that for each AIR and at each FBP, the end-effector paths can be generated or predefined if required for the application. In order for an AIR to be able to cover the whole path, it needs to find feasible AIR poses $P_i^f = \{p_{i1}^f, p_{i2}^f, \dots, p_{iy}^f\}$ associated with all discrete points in O_i . In order for an AIR pose p_{ij} to be considered a feasible pose p_{ij}^f , the AIR pose needs to be collision-free and have an acceptable end-effector orientation and position relative to the point o_{ij} where j is the index of the discrete point on the path of the i th AIR.

1) *Objective 1 - Maximize Reachability / Coverage:* This objective is to maximize the number of points along the paths of the n AIRs that can be reached with feasible AIR poses. The higher the proportion of points that can be reached with feasible AIR poses, to the total number of discrete points along a path, equates to a higher likelihood of an acceptable trajectory being found during the task execution.

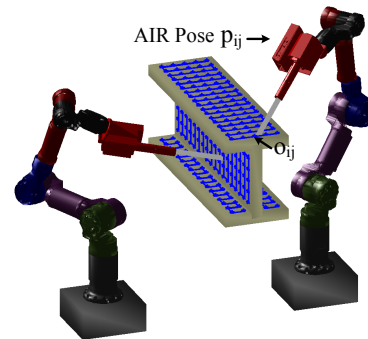


Fig. 4: Feasible poses of two AIRs performing the grit-blasting complete surface coverage task

This objective is to minimize

$$f_1 = \sum_{i=1}^n 1 - \frac{S_i^f}{S_i^T} \quad (1)$$

where S_i^f is the number of points that can be reached with feasible poses of the i th AIR, and S_i^T is the total number of discrete points in the set O_i along the paths of the i th AIR.

2) *Objective 2 - Minimize Makespan:* This objective is to minimize the makespan, i.e. to minimize the overall completion time of the intended task. This objective is particularly important in industrial applications since it aims to reduce the process cycle time. This objective is to minimize

$$f_2 = \max\{t_1, \dots, t_n\} \quad (2)$$

where t_i is the completion time of the i th AIR, which can be calculated based on the end-effector speed of the AIR and the time it takes for the AIR to cover all paths associated with it.

3) *Objective 3 - Maximize Manipulability Measure:* A commonly used performance metric is the manipulability measure, which was first introduced by Yoshikawa [14], and can help obtain a measure for a manipulator or a manipulator pose corresponding to a certain point in the workspace. Thus, this objective is to maximize the manipulability measure of all the AIRs, i.e. to minimize

$$f_3 = \sum_{i=1}^n 1 - \frac{\sum_{j=1}^{S_i^f} W(p_{ij}^f)}{S_i^T} \quad (3)$$

where

$$W(p_{ij}^f) = \sqrt{\det\left(J(p_{ij}^f)J^T(p_{ij}^f)\right)} \quad (4)$$

is the manipulability measure (a value from 0 to 1), J is the Jacobian corresponding to the i th AIR pose $p_{ij}^f = [\theta_1, \theta_2, \dots, \theta_{n_k}]$, j is the discrete point index and θ_1 to θ_{n_k} are the joint angles.

4) *Objective 4 - Minimize Torque:* Another important measure for the AIRs is the total torque experienced by the joints of each AIR during the task execution. Let p_{ij}^f be a feasible pose corresponding to the i th AIR and the j th discrete point o_{ij} . The maximum torque ratio for the pose p_{ij}^f is the largest torque ratio from all the joints (actuators) of the AIR. This objective is to minimize the sum of maximum torque ratios of the AIR that experiences the most amount of torque. Thus, to minimize

$$f_4 = \max_i \left[\frac{\sum_{j=1}^{S_i^f} \max_k \left| \frac{\tau_{ik}(p_{ij}^f, F_i)}{\tau_{ik}^{cap}} \right|}{S_i^T} \right] \quad (5)$$

where τ_{ik} is the torque experienced by joint k ($k = 1, 2, \dots, n_k$) of the i th AIR due to the end-effector force F_i and the gravitational forces at pose p_{ij}^f . τ_{ik}^{cap} is the torque capacity of k th joint of the i th AIR and n_k is the total number of joints of the i th AIR.

C. Design Constraints

1) *Constraint 1- Distance Between Any Two AIRs:* The distance between any two AIRs should be greater than a predefined threshold, δ_1 . This threshold will prevent the AIRs from being in close proximity to each other and thus, reduce the risk of collision between the AIRs, and avoid any negative effect on the maneuverability of the AIRs during the task execution. δ_1 can be determined based on the workspace of the AIRs. For example, if the AIRs are identical and their workspace boundary can be approximated as a sphere, then δ_1 can be the distance from the base of the AIR to the boundary of the AIR's workspace. The first constraint that needs to be met is

$$\|(\beta_{AIR})_i - (\beta_{AIR})_k\| > \delta_1 \quad (6)$$

$\forall i, k : i = 1, \dots, n, k = 1, \dots, n, i \neq k$ where $(\beta_{AIR})_i$ and $(\beta_{AIR})_k$ are the base positions of the i th and k th AIRs, respectively, and n is the total number of AIRs.

The distance between any AIR and the closest point in the environment has already been considered when selecting FBPs for each AIR, and is not necessary to be considered as an additional constraint.

IV. CASE STUDIES AND SIMULATIONS

A. Case Study 1- Two AIRs Applied in a Steel Bridge Maintenance Environment

In this case study, the grit blasting application is considered. As shown in Fig. 5a, two 6DOF AIRs are deployed to carry out grit-blasting on a test-rig replicating a section of a steel bridge structure. Each AIR is equipped with a nozzle and a sensor on the end-effector. The aggregate of the grit-blasting paths performed by the AIRs needs to result in complete grit-blasting coverage of all targeted surfaces in the environment. During the task execution, for a discrete point o_{ij} on the pre-generated paths, the nozzle of an AIR is to be oriented and positioned within an acceptable range relative to o_{ij} so as to allow the blasting stream exiting the nozzle to aim at and follow the path within an allowable deviation. The data collected from the sensors of the AIRs are used to create the simulated scenario as shown in Fig. 5b. In the simulation, the blue disks are all the discrete points, O_i ($i = 1, 2, \dots, n$) and are used to represent the surfaces. The size of the disks and the overlap between two adjacent disks are dependent on factors such as the AIRs end-effector speed and nozzle properties. In this simulation, the AIRs are considered to be identical and hence, all disks representing the surfaces are the same size for both AIRs. A total of 4130 disks are used to represent the surfaces.

The circles generated on the floor shown in Fig. 6 are all the discrete base positions created by both AIRs. In the figure, the crossed circles represent discarded base positions due to their proximity to obstacles/objects. The red filled circles represent discarded base positions due to their low coverage of the object, which were determined by obtaining an estimate on the number of targets that can be covered at each of the FBPs, through simple distance calculations

to find the targets that fall inside the workspace boundary of the AIRs. The remaining 30 unfilled circles are the good candidates (FBPs). The spacing and the method for creating the discrete base positions depend largely on the application. For the particular application being considered, it was empirically found that the discrete base positions are best to be spaced at 0.3 meters from each other.

An exhaustive search was considered for this case study for the following reasons: (1) a small number of AIRs are used, (2) the number of FBPs is relatively small, and (3) for the particular AIRs used, no orientation about the vertical axis needs to be considered for base placement since the first joint of the AIRs can do a full rotation about the vertical axis. However, if certain conditions such as a large number of FBPs or an increased number of AIRs cause large computation time, then an appropriate multiobjective optimization algorithm should be used [15].

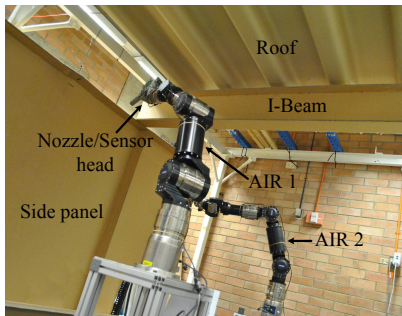
The first step is for each AIR to evaluate its performance against the team objectives at each of its FBPs prior to the optimization. The computation time for evaluating the 30 FBPs shown in Fig. 6 is 14.5 minutes using Matlab. The evaluation information obtained by each AIR is then exchanged between the AIRs, however since the AIRs are the same, then the evaluation is the same. After evaluating performance at each FBP, an exhaustive search was carried out and it took 5.5 seconds using Matlab. The same objectives and constraint functions as those outlined in Section III are used. Since uniform coverage of the surface is required, then the end-effector speed is constant, and hence the completion time t in (2) is calculated as $t_i = l_i v_i^{-1}$ where v_i is the end-effector

speed of the i th AIR, and l_i is the length of the paths that the i th AIR needs to cover. The threshold δ_1 used in (6) is set to 1 meter.

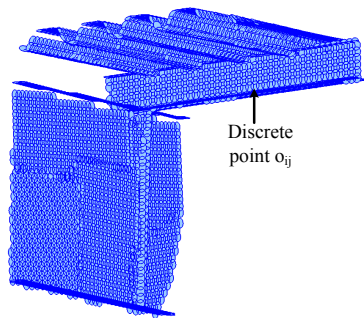
The best solution from the exhaustive search is shown in Fig. 6 where the AIRs are a good distance away from both the environment and each other. Using a lookup table, the AIRs were able to collectively find acceptable poses for 3760 of the 4130 discrete points (blue disks) representing the surfaces. It needs to be noted that some points, such as those at the back of the roof and the I-Beam, can not be reached by any of the AIRs regardless of the base placements selected.

In order for an AIR to find a feasible pose p_{ij}^f for the discrete point o_{ij} , a lookup table can be used, which maybe time efficient since the aim is not finding a feasible trajectory for the task execution. To generate the lookup table, the workspace of an AIR can be decomposed into a number of 3D grids, where each 3D grid is associated with many discrete AIR poses. The AIR poses associated with each 3D grid can be grouped based on the similarity of their end-effector poses. The AIR poses in each group can then be sorted based on manipulability measure, torque or other performance measures used in the team objectives. To find a feasible pose for o_{ij} , the poses in the relevant group of the 3D grid associated with o_{ij} are checked in the sorted order and the first collision-free pose is selected. Quadrees, octrees or similar hierarchical data structures [16] can be used to perform fast searches so as to acquire the relevant data, e.g. to find the 3D grid index which o_{ij} belongs to.

In this case study, the workspace of the AIR is decomposed into 262144 3D grids with each grid containing 67 groups of AIR poses. The AIR poses are grouped based on the similarity of their end-effector poses. Depending on the region of the AIRs workspace in which the 3D grid is in, e.g. dexterous or non-dexterous workspace [17], the number of poses associated with each 3D grid can be different. The number of poses in each group of a 3D grid are chosen to be anywhere from zero (meaning the 3D grid can not be reached with the required end-effectors' orientation) up to a maximum number of three hundred when the 3D grid is in the full dexterous workspace of the AIR.



(a) Two AIRs and the environment



(b) Simulated scenario generated using sensor data

Fig. 5: Real and simulated scenario for case study 1

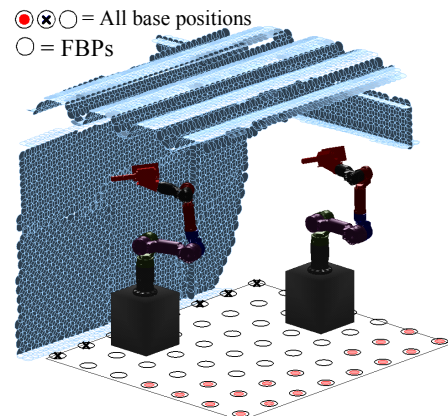


Fig. 6: Placement of the AIRs based on the final solution

Since multiple objectives are considered, then the Pareto front can be obtained [15]. Selection of the final solution from the Pareto front depends on the application. For the particular application being considered, since the top priority is to achieve maximum coverage then from the solutions of the exhaustive search, a small subset of solutions with the highest coverage is firstly chosen. Given that the next priority is to minimize the makespan, a further subset is chosen (from the original subset of solutions) based on the lowest makespan. The final solution can be selected from the second subset based on (a) the manipulability measure to help with maneuverability of the AIRs during the task execution, or (b) the lowest torque if the conditions of the AIRs' joints are critical. In the presented case study, the weighted average of torque and manipulability measure is used to select the final solution.

B. Case Study 2- Three AIRs Grit Blasting Three Objects

A total of three AIRs are considered to grit-blast the three objects shown in Fig. 3, which are separated from each other. The same type of AIRs, settings, and parameters as those mentioned in the first case study are once again used. The final solution shown in Fig. 7 is selected based on the same selection strategy mentioned in the first case study.

The computation time to evaluate the performance of all 33 FBPs is 3 minutes and to carry out the exhaustive search is 4.9 minutes using Matlab. It is clear that as the number of AIRs increase, an exhaustive search becomes inefficient and a multiobjective optimization algorithm, such those outlined in [15], is preferable. The AIRs were able to collectively find acceptable poses for all 664 discrete points (black disks) representing the surfaces of all three objects.

V. CONCLUSION AND FUTURE WORK

This paper presented an approach that optimizes the collaboration of multiple Autonomous Industrial Robots (AIRs) by finding appropriate base placements for the AIRs prior to executing the intended task. A mathematical model was developed by taking into account the capabilities of all AIRs, the shared environment information and robots' operation status to find an appropriate base placement that will improve the team objectives such as minimal makespan and maximum

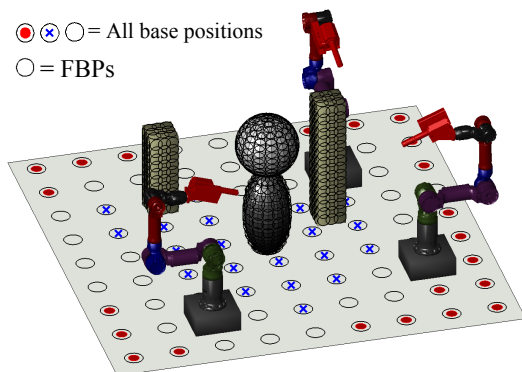


Fig. 7: Placement of the AIRs based on the final solution

coverage. A set of objective and constraint functions are used in the optimization model. Two case studies were presented to demonstrate the effectiveness of the approach for complex applications that require complete surface coverage such as grit-blasting.

Future work will be to extend the presented approach to help with selecting multiple base placements for each of the deployed AIRs, considering that the bases of the AIRs will need to be repositioned any number of times in order to complete the whole task or series of tasks.

REFERENCES

- [1] M. Hvilshj, S. Bgh, O. Skov Nielsen, and O. Madsen. Autonomous industrial mobile manipulation (aimm): past, present and future. *Industrial Robot: An International Journal*, 39(2):120–135, 2012.
- [2] D. Liu, G. Dissayanake, P.B. Manamperi, P.A. Brooks, G. Fang, G. Paul, S. Webb, N. Kirchner, P. Chotiprayanakul, N.M. Kwok, and T.R. Ren. A robotic system for steel bridge maintenance: Research challenges and system design. in *Proc. of the 2008 Australasian Conf. on Robotics and Automation*, ACRA, Canberra, Australia, 2008.
- [3] G. Paul, S. Webb, D. Liu, and G. Dissayanake. Autonomous robot manipulator-based exploration and mapping system for bridge maintenance. *Robotics and Autonomous Systems*, 59(78):543–554, 2011.
- [4] G. Paul, D. Liu, N. Kirchner, and G. Dissayanake. An effective exploration approach to simultaneous mapping and surface material type identification of complex three-dimensional environments. *Journal of Field Robotics*, 26(11/12):915–933, 2009.
- [5] N. Kwok G. Paul and D. Liu. A novel surface segmentation approach for robotic manipulator-based maintenance operation planning. *Automation in Construction*, 29(0):136–147, 2013.
- [6] P. Chotiprayanakul, D. Liu, D. Wang, and G. Dissayanake. A 3-dimensional force field method for robot collision avoidance in complex environments. in *Proc. of the 24th Int. Symposium on Automation and Robotics in Construction*, pages 139–145, 2007.
- [7] M. Hassan, D. Liu, S. Huang, and G. Dissayanake. Task oriented area partitioning and allocation for optimal operation of multiple industrial robots in unstructured environments. in *Proc. of the 13th Int. Conf. on Control, Automation, Robotics and Vision*, pages 1184–1189, Marina Bay Sands, Singapore, 2014.
- [8] L. E. Parker. Multiple mobile robot systems. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 921–941. Springer Berlin Heidelberg, 2008.
- [9] S. Mitsi, K. D. Bouzakis, D. Sagris, and G. Mansour. Determination of optimum robot base location considering discrete end-effector positions by means of hybrid genetic algorithm. *Robotics and Computer-Integrated Manufacturing*, 24(1):50–59, 2008.
- [10] G.-C. Vosniakos and E. Matsas. Improving feasibility of robotic milling through robot placement optimisation. *Robotics and Computer-Integrated Manufacturing*, 26(5):517–525, 2010.
- [11] G. Boschetti, R. Rosa, and A. Trevisani. Optimal robot positioning using task-dependent and direction-selective performance indexes: General definitions and application to a parallel robot. *Robotics and Computer-Integrated Manufacturing*, 29(2):431–443, 2013.
- [12] N. Papakostas, G. Michalos, S. Makris, D. Zouzas, and G. Chrysolouris. Industrial applications with cooperating robots for the flexible assembly. *International Journal of Computer Integrated Manufacturing*, 24(7):650–660, 2011.
- [13] L. Carlone, M. Kaouk Ng, J. Du, B. Bona, and M. Indri. Simultaneous localization and mapping using rao-blackwellized particle filters in multi robot systems. *Journal of Intelligent & Robotic Systems*, 63(2):283–307, 2011.
- [14] T. Yoshikawa. Manipulability of robotic mechanisms. *The International Journal of Robotics Research*, 4(2):3–9, 1985.
- [15] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32 – 49, 2011.
- [16] S. Peters. Quadtree- and octree-based approach for point data selection in 2d or 3d. *Annals of GIS*, 19(1):37–44, 2013.
- [17] S. Patel and T. Sobh. Manipulator performance measures - a comprehensive literature survey. *Journal of Intelligent & Robotic Systems*, pages 1–24, 2014.