

# A Two-Stage Approach to Collaborative Fiber Placement through Coordination of Multiple Autonomous Industrial Robots

Mahdi Hassan · Dikai Liu · Dongliang Xu

Received: date / Accepted: date

**Abstract** The use of multiple Autonomous Industrial Robots (AIRs) as opposed to a single AIR to perform fiber placement brings about many challenges which have not been addressed by researchers. These challenges include optimal division and allocation of the work and performing path planning in a coordinated manner while considering the requirements and constraints that are unique to the fiber placement task. To solve these challenges, a two-stage approach is proposed in this paper. The first stage considers multiple objectives to optimally allocate each AIR with surface areas, while the second stage aims to generate coordinated paths for the AIRs. Within each stage, mathematical models are developed with several unique objectives and constraints that are specific to the multi-AIR collaborative fiber placement. Several case studies are presented to validate the approach and the proposed mathematical models. Comparison studies using a different number of AIRs and variations of the developed mathematical models are also presented.

**Keywords** Multi-robot fiber placement · Fiber reinforced composites · Multiple autonomous industrial robots · Tool-path allocation · Complete coverage

---

M. Hassan  
Centre for Autonomous Systems (CAS) at the University of Technology Sydney (UTS), Australia  
Tel.: +612-9514-2000  
Fax: +612-9514-3654  
E-mail: Mahdi.Hassan@uts.edu.au

D.K. Liu  
Centre for Autonomous Systems (CAS) at the University of Technology Sydney (UTS), Australia  
E-mail: Dikai.Liu@uts.edu.au

D. Xu  
School of Mechanical and Electronic Engineering, Wuhan University of Technology, 430070 Wuhan, China  
E-mail: xdl556@163.com

## 1 Introduction

When compared to traditional materials such as aluminum and steel, fiber reinforced composites have additional favorable properties such as versatility, strength-to-weight ratio, and stiffness-to-weight ratio [1]. Thus, fiber reinforced composites are suitable for structures in many industries, e.g., aeronautical and automotive industries [2].

Autonomous Industrial Robots (AIRs) can be used to perform fiber placement. An AIR [3] is an industrial robot with or without a mobile platform, that has self-awareness and environmental awareness, and can effectively communicate with other AIRs to share information on aspects such as robots' state or operation status, the sensed environment, etc. If the AIR is attached to a mobile platform, then its definition is the same as the Autonomous Industrial Mobile Manipulator (AIMM) [4]. The mathematical models developed in this paper consider both mobile and immobile platforms.

When conducting robotic fiber placement [5, 2], at first, the fiber strips are pre-impregnated with a proper amount of thermoset resin. The fiber strips are then fed from a supply spool and through several guiding rollers and tensioners to the fiber placement head. The fiber placement head is attached to the end-effector of a robotic manipulator, and its task is to cool the fiber strips, cut them to the correct length and collimate them. Using controlled heating and an elastomeric roller the fiber is then attached to the surface. The heating is used to provide the desired tackiness, and the elastomeric roller is used to place the fiber on the surface with the proper amount of compaction force. The compaction will help prevent gaps and trapped air. Thus, the roller simultaneously lays a number of tows (or fiber strips) on the surface. The number of tows can be adjusted to form a desired tow pattern. A structure constructed using fiber-reinforced composites can be made up of a number of layers, and each layer consists of many tool-paths. The path that the roller

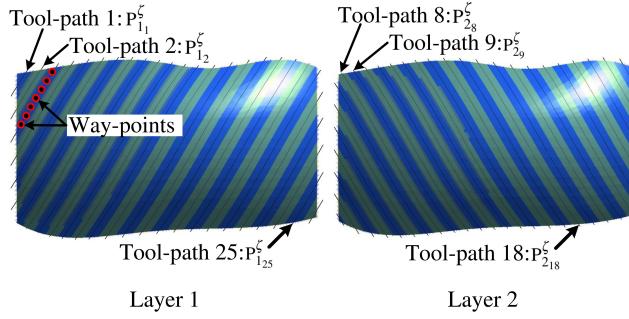


Fig. 1: Two layers of fiber and the associated tool-paths for each layer are shown for an example curved structure.

follows to lay the fiber is defined as a tool-path. Fig. 1 shows two layers and the tool-paths for a curved structure. Finally, the structure is placed in an oven where appropriate heating and pressure is applied to cure the structure.

In this paper, the tool-paths and the way-points along these tool-paths are assumed to be given or can be generated using an existing method [6, 7, 8] as will be explained in the next section. Since this paper focuses on the multi-AIR collaboration aspect of the overall problem, then other well-studied tasks such as generating the tool-paths based on the geometry of the structure and AIRs' base localization are outside the scope of the paper.

Multi-AIR collaborative fiber placement has not received much attention by researchers. Enabling multiple AIRs to collaborate and simultaneously perform fiber placement requires solving several challenges. The challenges addressed in this paper are concerned with the following questions: (1) how would the AIRs divide the work amongst themselves and collaboratively decide on the sequence of the work, (2) how would the AIRs collaborate to achieve optimal outcomes, e.g. in terms of makespan and maneuverability, and (3) how would the AIRs execute the allocated tasks while coordinating their motion to avoid hindering the motion of each other, avoid obstacles, and satisfy the relevant requirements. These challenges have been addressed for various applications; however, when addressing these challenges specifically for the multi-AIR collaborative fiber placement, then new mathematical models need to be devised that consider specific objectives and constraints to satisfy the unique requirements of the fiber placement task.

To solve the above challenges, a two-stage approach is presented where for each stage, a mathematical model is developed that takes into account fiber placement related requirements and constraints. The mathematical models for both stages complement each other in that they both, in a combined effort, aim to optimize the collaboration and productivity of the AIRs for fiber placement.

Figure 2 shows the two stages of the collaborative fiber placement work to be carried by the AIRs. The first stage

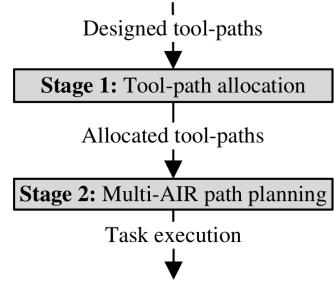


Fig. 2: The two stages considered for multi-AIR collaborative fiber placement.

involves tool-path allocation. The input to the approach is the tool-paths (given) of all layers. The tool-path allocation problem consists of determining: (1) the *number* of tool-paths for each AIR to execute, (2) the *indexes* of the tool-paths that each AIR needs to execute where index of a tool-path is a unique number given to the tool-path for the purpose of identification, and (3) the *executing sequence* of the selected tool-paths for each AIR. Relevant objectives (such as minimal makespan) need to be optimized, and constraints need to be satisfied.

The second stage of the approach is related to the multi-AIR path planning. In this paper, the tool-paths are given and thus, the path planning refers to finding the appropriate joint angles of the AIRs (inverse kinematics) to reach each point in a tool-path such that coordination between the AIRs is achieved, relevant constraints (such as collision avoidance) are satisfied, and relevant objectives (such as minimal change in joints' angles) are optimized. In addition to the general constraints, fiber placement imposes a unique set of constraints that the path planner needs to take into account, e.g., meeting the required compaction force by the roller and operating within the allowable orientation error of the roller with respect to the surface normal. As shown in Fig. 2, the tool-paths allocated to each AIR (output of Stage 1) are the inputs to Stage 2. Some of the objectives and constraints in Stage 1 are designed specifically to make the path planner in Stage 2 more effective. The output of Stage 2 (joints' angles) is then used by a motion controller for task execution.

The rest of the paper is organized as follows. Section 2 discusses related works. Section 3 defines the problem of collaborative fiber-placement by multiple AIRs, and outlines the assumptions considered. Sections 4 and 5 present the mathematical modeling for Stage 1 (tool-path allocation) and Stage 2 (multi-AIR path planning), respectively. Section 6 discusses the optimization process for the two stages. Several case studies are then presented in Section 7 to demonstrate the effectiveness of the approach using different structures. Discussion on computational efficiency and future work are presented in Section 8. Concluding remarks are made in Section 9.

## 2 Related Works

A well-studied task in Robotic Fiber Placement (RFP) is generating the tool-paths that the robot's end-effector (or the roller) follows to lay the fiber. When generating the tool-paths, information such as the geometry of the object, tow width, the number of layers and the acceptable overlap or gap between tows need to be taken into account [6]. This information can help in determining the laying direction of the fiber in each layer, the offset between each tool-path, etc. The problem of generating tool-paths has been widely studied. Thus, tool-paths are assumed to be given in this paper. Li et al. [6] provide a survey of the recent algorithms for RFP path planning. They group the existing algorithms into two main categories: (1) fixed angle path planning where the angle between the laying path and the reference line is conveniently kept the same, and (2) variable angle path planning where the laying angle can change to help the fiber reach its full performance potential.

The following are some literature related to the tool-path planning problem. As part of the overall method in [5], the authors aim to smooth the path, i.e. to determine and smooth the orientation of the tool head along the path whilst meeting the required quality and constraints. Shirinzadeh et al. [1] applied their developed SCAR path planning algorithm to open-contoured structures by formulating a set of surface curves that represent the tool-paths. The algorithm aims to create uniform layers of fiber with no gap or overlap between tows. Yan et al. [8] argue that many of the studies assume a constant value when offsetting a reference tool-path to generate a new tool-path. Thus, Yan et al. [8] consider a varied offset and other improvements to provide a more accurate approach to path generation and to satisfy the required gaps and overlaps between consecutive tows. In [9], a numeric solution to the initial-value problems of systems with first-order ordinary differential equations is used to generate the initial reference path and the offset paths which in turn reduces the computational complexity or time and improves control error. Some new research works focus on utilizing a mesh directly instead of using geometric equations to generate the tool-paths [10, 11].

Besides the problem of generating tool-paths, optimizing RFP process parameters and improvements in the design of RFP tools have also been studied. Aized and Shirinzadeh [12] conduct several experiments to arrive at optimized parameters related to gas torch temperature used to heat the fiber, fiber laying speed and fiber compaction force. Jeffries [13] presents a robot technology and a modular fiber placement head to improve the RFP process. The modular fiber placement head has an automatic, quick-change, tool changer that provides greater flexibility and productivity.

As discussed above, there is a wide range of research works dedicated to tool-path planning and optimizing RFP

processes or tools. Therefore, the work in this paper assumes that the tool-paths and the optimized parameters are the input to the multi-AIR collaborative fiber placement problem.

In this paper, multiple independent AIRs are considered and the AIRs can lay fibers simultaneously and cooperatively while being close to each other. Research on Fiber placement has not focused on the use of multiple AIRs; instead, the limited number of studies on collaborative fiber placement machines focus on developing systems that are coupled together and only consider one robot to lay the fiber. The components of the coupled system mainly help with carrying the mandrel, turning the mandrel, assisting with reachability of the roller, etc. For example, in [14], a collaborative machine is presented which comprises a 6-DoF manipulator, a 6-RSS parallel platform, and a spindle to hold the mandrel. They construct the kinematics of the overall machine, analyze the workspace and prove that the overall machine can enlarge the workspace, simplify trajectory planning, and in-turn improve productivity. Another example of an RFP machine is presented in [5] where the authors take advantage of the additional degree of redundancy to decrease the kinematic loads of the control joints.

Many research works related to task sequencing, scheduling or allocation exist that can provide inspiration to solving the problem under consideration. The work in [15] provides a survey on robotic task sequencing and scheduling for industrial applications. The following are some example approaches and applications for task sequencing and scheduling. In [16], path planning, job scheduling and collision avoidance of multibridge machining system are solved using a variation of the well-known traveling salesmen problem and population-based incremental learning. Han et al. [17] developed algorithms for allocating asset packages to a set of interdependent tasks that are associated with resource requirements. Later, Han et al. [18] extended the work to handle asset allocation and task planning for multiple agents by devising a blackboard-based collaborative framework. A generic framework for distributed multi-robot cooperation is presented in [19]. The framework is constructed to address a variety of missions that can be simple, complex, have inter-related tasks, or have multi-resource requirements. The tool-path allocation problem presented in this paper is solved with a particular formulation of the mathematical model that is mostly specific to the multi-AIR fiber placement.

Path and trajectory planning for industrial robots or manipulators have been widely studied [20, 21]. Examples include approaches that are based on: artificial force field [22]; optimization methods [23]; vision-based control (camera-space manipulation) [24]; rapidly-exploring random trees and probabilistic roadmaps [25]; screw theory [26]; and recurrent neural network [27]. Control of robotic manipulators have also been widely studied [28, 29, 30]. However, in this paper, the multi-AIR path planning problem (stage

2) is solved with a particular focus on the fiber placement task. That is, the focus has been to develop a mathematical model that considers fiber placement related objectives and constraints while aiming to achieve coordinated execution of the task. The first stage (tool-path allocation) is formulated such that it complements the multi-AIR path planner. The multi-AIR path planning problem is formulated as an optimization problem and can be solved using standard optimization algorithms. The output of the multi-AIR path planner is coordinated AIR poses. This output is then passed to a motion controller for execution.

### 3 Problem Definition

When multiple AIRs are deployed, they are expected to complete the fiber placement task collaboratively and optimally. Therefore, the AIRs are required to:

- equitably divide the tasks by appropriately allocating the tool-paths in all layers amongst themselves,
- determine the execution sequence of the tool-paths such that the potential for hindering the motion of each other is minimized and maneuverability within the workspace is maximized,
- aim to achieve minimal makespan and minimal movement of the joints and the base,
- execute the allocated tool-paths in a coordinated manner,
- avoid self-collision, collision with the environment or the object, and collision with other AIRs, and
- satisfy the requirements inherent to the fiber placement task, such as compaction force and appropriate orientation of the roller to the surface.

To solve the above challenges, appropriate mathematical models need to be devised.

Let  $P = \{P_{Z_I} : Z = 1, 2, \dots, Z^{\text{tot}}; I = 1, 2, \dots, n_Z^{\text{tot}}\}$  be a set containing all the tool-paths in all layers where  $Z$  is the layer index,  $I$  is the tool-path index,  $n_Z^{\text{tot}}$  is the total number of tool-paths in the  $Z$ th layer, and  $Z^{\text{tot}}$  is the total number of layers. Let  $N_Z$  be the total number of tool-paths in all layers, i.e.  $N_Z = \sum_{Z=1}^{Z^{\text{tot}}} n_Z^{\text{tot}}$ . In this work, the tool-paths are assumed to be given. The goal is to deploy  $n$  AIRs and collaboratively execute all  $N_Z$  tool-paths as efficiently as possible. To achieve this goal, the AIRs are allowed to work on different layers. For the example shown in Fig. 1, it is assumed that an AIR can work on a tool-path of the second layer while another AIR is still working on a tool-path of the first layer. However, for an AIR to start on a tool-path, it needs to ensure that all the relevant tool-paths from the preceding layers are executed first. To denote this prerequisite condition in the notation of a tool-path, the notation is modified as follows:  $P_{Z_I}^\zeta$  where  $\zeta$  represents the set of *prerequisite tool-paths*, meaning that  $P_{Z_I}^\zeta$  can be executed only if all tool-paths

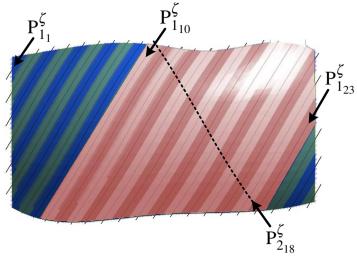


Fig. 3: The tool-paths highlighted in red ( $P_{110}^\zeta$  to  $P_{123}^\zeta$ ) are the prerequisite tool-paths of  $P_{218}^\zeta$ .

in  $\zeta$  (which are from preceding layers) are already executed. As was shown in Fig. 1,  $P_{218}^\zeta$  is the 18th tool-path of the 2nd layer. In Fig. 3, the tool-paths highlighted in red ( $P_{110}^\zeta$  to  $P_{123}^\zeta$ ) are the prerequisite tool-paths of the tool-path  $P_{218}^\zeta$ ; hence,  $\zeta = \{P_{110}^\zeta, P_{111}^\zeta, \dots, P_{123}^\zeta\}$  for the tool-path  $P_{218}^\zeta$ .

To maintain the strength of the fiber, it is expected that the AIRs will lay the fiber for the entire tool-path in one pass. This also eliminates the need for cutting the fiber and starting the pass again with a different trajectory. Hence, one tool-path is allocated to one AIR only.

The tool-path allocation problem is to select a subset of tool-paths for each AIR, i.e. to select  $P_i \subset P (\forall i, i = 1, 2, \dots, n)$  where  $i$  is the AIR index. The tool-paths  $P_i$  need to be sorted in the optimal sequence for execution. The tool-path allocation problem needs to be solved such that collaboration is optimized, productivity is maximized, and multi-AIR path planning is made easier to solve.

The multi-AIR path planning problem in this paper involves finding feasible AIR poses (inverse kinematics) for all way-points of the tool-path while considering the coordination of the AIRs when executing the allocated tool-paths. The path planning must take into account constraints related to the operation of the AIRs (e.g. avoiding collisions and exceeding joints' limits), but also constraints related to fiber placement (e.g. compaction force and acceptable orientation of the roller relative to the surface).

The methodology for carrying out the fiber placement task using multiple AIRs depends on a number of factors including the configuration of the workplace in which the AIRs operate, the size and complexity of the object, and the structure of the AIR itself. Thus, devising a single generic methodology that can handle all variations of the collaborative fiber placement problem may not be optimal for a specific class of problems. To this end, the below assumptions are presented to better define the scope of the work and to clarify the class of problems for which the developed approach applies:

- The target object's position is fixed, and there are no facilities (such as a spindle) to turn the object while fiber placement is being carried out by the AIRs.
- The base of each AIR can be mobile; however, for stability and accuracy, the base needs to be fixed (immobile) when executing the fiber placement task for a tool-path. After completing a tool-path, the base can be moved autonomously or manually to the next optimized base placement for executing the next tool-path.
- If the bases of the AIRs are designed to be permanently fixed at specific locations, then the object needs to be positioned appropriately relative to the AIRs.
- Fiber storage unit for each AIR is installed on the AIR (e.g. at the base of the AIR), and the fiber placement head is fed with fiber from the storage unit when needed. Alternatively, the fiber can be fed from a central unit that is away from the AIRs; however, AIRs' path planning will be more challenging since collision avoidance with the fiber by the AIRs would then be needed.

#### 4 Stage 1: Mathematical Modeling for Tool-path Allocation

This section presents the mathematical modeling for tool-path allocation (i.e. Stage 1 shown in Fig. 2). The aim is to optimize collaboration and productivity of the AIRs and to make the multi-AIR path planning problem (Stage 2) easier to solve. Note that the tool-paths are discretized into waypoints with equal distance between any two adjacent waypoints along a tool-path, as was shown in Fig. 1. At a time step  $\tau_j (j = 1, 2, \dots, n^j)$ , where  $n^j$  is the number of steps to complete the task,  $n$  waypoints from different tool-paths may need to be simultaneously executed by  $n$  AIRs.

##### 4.1 Design Variables

The design variables are:

$$X_{Z_I} \in \{1, 2, \dots, n\}, \quad (1)$$

$\forall Z, I : Z = 1, 2, \dots, Z^{\text{tot}}, I = 1, 2, \dots, n_Z^{\text{tot}}$  where  $Z$  is the layer index and  $I$  is the tool-path index. As an example,  $X_{3,10} = 2$  means that the tool-path  $P_{3,10}^\zeta$  is allocated to the 2nd AIR. Ultimately, the aim is to arrive at values for the design variables that will optimize the following design objectives and constraints.

##### 4.2 Design Objectives

###### 4.2.1 Minimal Makespan

One of the key objectives in deploying multiple AIRs is to minimize the overall completion time of the task (i.e.

makespan). By minimizing the makespan, the idle time of each AIR can be minimized as well. Idle time is associated with a poor allocation of the tool-paths where AIRs have to wait for an excessive amount of time while the prerequisite tool-paths are being executed. Let  $X$  be a set containing all the design variables, i.e.  $X = \{X_{Z_I} : \forall Z, I\}$ . This objective is:

$$\min_X f_t(X) = \max \left( t_1(X), t_2(X), \dots, t_n(X) \right) \quad (2)$$

where completion time  $t_i(X)$  of the  $i$ th AIR can be calculated as:

$$t_i(X) = \sum_{Z=1}^{Z^{\text{tot}}} \sum_{I=1}^{n_Z^{\text{tot}}} \left( \left( \frac{l_{Z_I}}{v_i} + t_i^t + t_i^w \omega_i \right) \alpha(X_{Z_I} \in X) \right), \quad (3)$$

$Z^{\text{tot}}$  is the total number of layers,  $n_Z^{\text{tot}}$  is the total number of tool-paths in the  $Z$ th layer,  $l_{Z_I}$  is the length of the tool-path  $P_{Z_I}^\zeta$ ,  $v_i$  is the constant end-effector speed of the  $i$ th AIR,  $t_i^t$  is a time constant associated with the  $i$ th AIR cutting the fiber at the end of a tool-path and then moving its end-effector (and maybe its base) to start on the next allocated tool-path,  $t_i^w$  is the idle time, and  $\omega_i$  is a weighting factor for increasing/decreasing the significance of  $t_i^w$  in the objective. The idle time is the time that the  $i$ th AIR has to wait prior to starting on a tool-path while all prerequisite tool-paths in  $\zeta$  are being executed. Minimizing the makespan will also minimize the idle time for all AIRs. If the idle time is critical for an AIR (e.g. for a slow AIR), then the weighting  $\omega_i$  can be increased, and vice versa. The term  $\alpha(X_{Z_I} \in X)$  is to ensure that the execution time corresponding to the tool-path  $P_{Z_I}^\zeta$  is only added if  $P_{Z_I}^\zeta$  belongs to the  $i$ th AIR, i.e.

$$\alpha(X_{Z_I} \in X) = \begin{cases} 1 & \text{if } X_{Z_I} = i \\ 0 & \text{if } X_{Z_I} \neq i \end{cases}. \quad (4)$$

###### 4.2.2 Maximal Distance Between AIRs' End-effectors

If at any stage of AIRs' operation, the end-effectors of any two AIRs are close to each other, then either of the two AIRs may have to stop to avoid collisions. To prevent this situation, the distance between the end-effectors of the AIRs at each time step  $\tau_j$  can be maximized. Therefore, this objective is:

$$\max_X f_d(X) = \min_j (d(\tau_j, X)) \quad (5)$$

where  $d(\tau_j, X)$  returns the smallest distance of all distances between the  $n$  waypoints that are to be executed by the  $n$  AIRs at time step  $\tau_j (j = 1, 2, \dots, n^j)$ . That is,

$$d(\tau_j, X) = \min_{i,i} (\|\mathbf{o}_i(\tau_j, X) - \mathbf{o}_i(\tau_j, X)\|) \quad (6)$$

$\forall i, \hat{i} : i = 1, 2, \dots, n; \hat{i} = 1, 2, \dots, n; i \neq \hat{i}$ , where  $\mathbf{o}_i(\tau_j, X) \in \mathbb{R}^3$  and  $\mathbf{o}_{\hat{i}}(\tau_j, X) \in \mathbb{R}^3$  return the position of the way-points that the  $i$ th and the  $\hat{i}$ th AIRs execute, respectively, at time step  $\tau_j$ . This objective will also help the multi-AIR path planner (Stage 2) to be more effective in finding feasible solutions.

#### 4.2.3 Minimal Distance Between Any Two Consecutive Tool-paths

It is preferred for an AIR to execute as many tool-paths as possible at a base position before moving to a new base position. This will minimize energy consumption and reduce complications associated with the base path planning (further explained in Stage 2). To help an AIR execute as many tool-paths as possible at a base position, the distance between any two consecutive tool-paths that the AIR executes is minimized. Note that minimizing the distance between any two consecutive tool-paths will also minimize the distance between the end of one tool-path and the start point of the next. Thus, the objective is:

$$\min_X f_p(X) = \sum_{i=1}^n \sum_{k=1}^{n_i^p(X)-1} R(\mathbf{P}_{i,k}, \mathbf{P}_{i,k+1}, X) \quad (7)$$

where the function  $R(\mathbf{P}_{i,k}, \mathbf{P}_{i,k+1}, X)$  calculates the maximal distance between the  $k$ th and  $(k+1)$ th tool-paths (i.e. the tool-paths  $\mathbf{P}_{i,k}$  and  $\mathbf{P}_{i,k+1}$ ) that the  $i$ th AIR executes, and the function  $n_i^p(X)$  returns the number of tool-paths allocated to the  $i$ th AIR based on the design variables  $X$ , i.e.

$$n_i^p(X) = \sum_{Z=1}^{Z^{\text{tot}}} \sum_{I=1}^{n_Z^{\text{tot}}} \alpha(X_{Z,I} \in X) \quad (8)$$

where  $\alpha(X_{Z,I} \in X) \in \{0, 1\}$  as defined in Eq. (4).

#### 4.2.4 Maximal Reachability

If the AIRs' bases are permanently fixed at a particular location during the entire fiber placement task, then it is assumed that the object is placed at an appropriate position and orientation relative to the AIRs such that each AIR can cover all tool-paths. However, even if this condition is satisfied, there is still no guarantee that an AIR can reach its allocated tool-paths. This is because the AIRs may hinder the motion of each other if the tool-paths are not allocated appropriately. Thus, the tool-paths need to be allocated to each AIR in a manner that will minimize the distance between each AIR and its allocated tool-paths. This condition will provide greater flexibility for any AIR to maneuver within its workspace without hindering the motion of other AIRs. This objective will help the multi-AIR path planner (Stage 2) to be more effective in finding feasible solutions. If the AIRs'

bases are mobile, then this objective can be discarded. The objective is:

$$\min_X f_r(X) = \sum_{i=1}^n \sum_{j=1}^{n^j} \left\| \mathbf{o}_i(\tau_j, X) - \boldsymbol{\beta}_i^{\text{AIR}} \right\| \quad (9)$$

where  $n^j$  is the total number of steps to complete the fiber placement task,  $\mathbf{o}_i(\tau_j, X) \in \mathbb{R}^3$  returns the position of the way-point that the  $i$ th AIR executes at time step  $\tau_j$ , and  $\boldsymbol{\beta}_i^{\text{AIR}} \in \mathbb{R}^3$  is the fixed base position of the  $i$ th AIR.

### 4.3 Design Constraints

#### 4.3.1 One Tool-path Must be Executed by One AIR only

$P$  was defined as the set that contains the tool-paths for all layers; i.e.  $P = \{P_{Z,I} : \forall Z, I\}$  where  $Z$  is the layer index and  $I$  is the tool-path index. A subset  $P_i \subset P$  is allocated to each AIR based on the values of the design variables where  $i$  is the AIR index. Two subsets of tool-paths associated with any two AIRs must not contain the same tool-path since one tool-path must be executed by one AIR only and partial allocation of a tool-path is not allowed to maintain the strength of the fiber. Thus, the constraint is

$$P_i - P_j = P_i \quad (10)$$

$$\forall i, j : i = 1, 2, \dots, n; j = 1, 2, \dots, n; i \neq j.$$

#### 4.3.2 All Tool-paths Must be Executed

The sum of tool-paths allocated to all AIRs must equal to  $N_Z$  which is the total number of tool-paths in all layers; i.e.

$$\sum_{i=1}^n |P_i| = N_Z \quad (11)$$

where  $|\cdot|$  denotes cardinality.

#### 4.3.3 Prerequisite Tool-paths Must be Executed First

For each of the tool-paths in the set  $\{P_{Z,I}^{\zeta} : Z = 2, 3, \dots, Z^{\text{tot}}; I = 1, 2, \dots, n_Z^{\text{tot}}\}$  there exists a set  $\zeta$  that contains the prerequisite tool-paths. Hence, the use of the superscript  $\zeta$  which is added to the notation to denote that the execution of a tool-path  $P_{Z,I}^{\zeta}$  is subject to the constraint of prerequisite tool-paths which must be executed first.

This constraint was partially accounted for using Objective 1 (minimal makespan). Objective 1 was designed such that by minimizing the makespan, the idle times of the AIRs are also minimized. Recall that the idle time of an AIR is the time the AIR has to wait while other AIRs execute the prerequisite tool-paths. However, since a tool-path and its prerequisite tool-paths can be allocated to the same AIR, then

the poor allocation of the tool-paths may cause an infeasible solution where an AIR is expected to execute a tool-path before executing all prerequisite tool-paths (by the same AIR).

## 5 Stage 2: Mathematical Modeling for Multi-AIR Path Planning

This section presents the mathematical modeling for multi-AIR collaborative path planning (i.e. Stage 2 shown in Fig. 2). The output from tool-path allocation, which is the tool-paths assigned to each AIR and the corresponding way-points, are used as the input to the following mathematical model for multi-AIR path planning. At each time step  $\tau_j$  (henceforth the notation will be reduced to step  $j$ ) the following mathematical model is used to find coordinated AIR poses (joint angles of all AIRs) that execute the corresponding way-points. The coordination between the AIRs is achieved by finding collision-free and feasible AIR poses for all AIRs at each step, as per the below mathematical model. The coordinated AIR poses at all time steps are then passed to a motion controller which is tasked with finding a smooth manipulator motion through the computed AIR poses. Collision avoidance may need to be considered during both multi-AIR path planning and path execution (motion control). The following mathematical model for the multi-AIR path planner considers collision avoidance based on the current environmental information and AIRs states. Without considering collision avoidance during the multi-AIR path planning, the path of the AIRs may not be optimized or feasible during the path execution. During the path execution, the motion controller may need to consider collision detection and avoidance as well based on the updated information of the environment or AIRs' state. This paper assumes that an appropriate motion controller from the existing controllers can be implemented. Note that the AIR's base doesn't move while executing a tool-path; however, at the completion of a tool-path, an appropriate point-to-point path planner may be needed for an AIR to go from a base position to the next within an allocated time ( $t_i^t$  in Eq. (3)). Point-to-point path planning is a well-studied problem and is not considered in the below mathematical model.

### 5.1 Design Variables: AIR Joints' Angles

At each step  $j$ , each AIR has to find a feasible pose corresponding to the way-point that it needs to execute. A feasible AIR pose can be obtained by calculating appropriate joints' angles for the AIR. Hence, the design variables are:

$$\theta_{j,iq}, \forall i, q : i = 1, 2, \dots, n; q = 1, 2, \dots, n_i^q \quad (12)$$

where  $\theta_{j,iq}$  is the joint angle corresponding to joint  $q$  of the  $i$ th AIR at step  $j$ ,  $n$  is the number of AIRs, and  $n_i^q$  is the num-

ber of joints of the  $i$ th AIR. The aim is to arrive at values for the design variables that will optimize the design objective whilst satisfying the relevant constraints. Note that the mathematical model is presented for one step only, hence to obtain a complete path for the AIRs it needs to be repeated  $n^j$  number of times to solve for  $j = 1, 2, \dots, n^j$ .

### 5.2 Design Objectives

Two design objectives are considered; however, only one is needed at each step. The first objective is related to minimizing the change in joints' angles of the AIRs while executing a tool-path. The second objective is only relevant for executing the first way-point of a tool-path, and the aim is to determine a new base position and orientation for the first way-point of a tool-path such that the movement of AIR's base is minimized. The design constraints are the same when using either of the objective function.

#### 5.2.1 Minimal Change in Joints' Angles

To help the controller execute a smooth motion, this objective is designed to minimize the change in joints' angles, i.e.:

$$\min_{\theta} f_a(\theta) = \max_{i,q} \left( \frac{|\theta_{j-1,iq} - \theta_{j,iq}|}{\theta_{iq}^{\text{cap}}} \right) \quad (13)$$

where  $\theta$  is a set containing all the design variables, i.e.  $\theta = \{\theta_{j,iq} : \forall i, q\}$ ,  $\theta_{j-1,iq}$  is the value (angle) of joint  $q$  of the  $i$ th AIR at step  $j - 1$ , and  $\theta_{iq}^{\text{cap}}$  is the maximum rotation capacity of the  $q$ th joint of the  $i$ th AIR. An option for calculating  $\theta_{iq}^{\text{cap}}$  is to consider the worst case scenario where the joint is initially still and then calculating the maximum rotation that the joint can make within a time step given its maximum acceleration and velocity. Thus,  $\theta_{iq}^{\text{cap}}$  acts as a normalization. The optimizer should not necessarily minimize the rotation of the joint that needs to turn more, but rather should minimize the rotation of the joint that is slower or requires a greater acceleration in meeting the desired rotation.

#### 5.2.2 Minimal Movement of AIRs' Bases

If an AIR is equipped with a mobile platform at its base, then at the start of each tool-path, in addition to determining appropriate values for the joints' angles of the AIR, the position and orientation of the mobile base need to be determined. Note that not all 6 degrees of freedom (position along the  $x, y, z$  axes, and orientation about the  $x, y, z$  axes of the base) need to be considered for the base since for most industrial robots the base may not be able to move vertically or make a rotation about the  $x$  and  $y$  axes.

Two options for determining appropriate base position and orientation are: (1) incorporating additional design variables that represent the base position and orientation of the AIRs into the aforementioned design variables (Eq. (12)) and then solving the optimization problem for the first way-point of a path; and (2) using the base placement optimization method presented in [31, 32] which can determine a base placement by considering reachability to all way-points in a tool-path. Note that the base is assumed to be fixed (immobile) when executing the fiber placement task for a tool-path, hence an AIR's base is assumed mobile only after completing a tool-path and moving to the next, as discussed in Section 3. The first option can be faster than the second; however, unlike the second option, it does not guarantee reachability to all way-points. This is because the base position is optimized for the first way-point only, and the same optimized base is assumed to be feasible for executing the rest of the tool-path. If all way-points of a tool-path couldn't be reached at a base position, then the planning process needs to be repeated from the start of the tool-path with a different base position.

Thus, optimization is performed twice for the first point of each tool-path: first to find a feasible base pose (position and orientation) that will result in minimal base movement (using the objective function below), then to find joints angles that will result in minimal joints motion (using Eq. (13)). This objective is:

$$\min_{\vartheta} f_b(\vartheta) = \max_i \left( \max \left( \frac{\|\boldsymbol{\beta}_{i,j-1}^{\text{AIR}} - \boldsymbol{\beta}_{i,j}^{\text{AIR}}(\vartheta)\|}{\beta^{\text{step}}}, \frac{|\psi_{i,j-1}^r - \psi_{i,j}^r(\vartheta)|}{\psi^{r,\text{step}}}, \frac{|\psi_{i,j-1}^p - \psi_{i,j}^p(\vartheta)|}{\psi^{p,\text{step}}}, \frac{|\psi_{i,j-1}^y - \psi_{i,j}^y(\vartheta)|}{\psi^{y,\text{step}}} \right) \right) \quad (14)$$

where  $\vartheta$  is a set of design variables that define the position and orientation (the pose) of the AIRs' base,  $\boldsymbol{\beta}_{i,j-1}^{\text{AIR}} \in \mathbb{R}^3$  is the preceding base position of the  $i$ th AIR,  $\boldsymbol{\beta}_{i,j}^{\text{AIR}}(\vartheta) \in \mathbb{R}^3$  is the new base position,  $\beta^{\text{step}}$  is the maximum translational motion that the base can handle at one time step,  $\psi$  represents the rotation of the base, and  $\psi^{\text{step}}$  is the maximum rotation that the base can handle at one time step where the superscripts  $r, p$ , and  $y$  denote roll, pitch and yaw angles, respectively. This objective is for  $i = 1, 2, \dots, n^m$ ;  $n^m \leq n$ , where  $n^m$  is the number of AIRs that need to move to a new base position at step  $j$ . Although the design variables related to the AIRs' joint angles (expressed in Eq. (12)) are not included in the above objective function (Eq. (14)), they are included in the following constraints functions to ensure that feasible AIR poses can be achieved. Hence, the complete set of design variables are  $\theta = \{\theta, \vartheta\}$ ; however, when using the first objective function (Eq. (13)) the design variables in  $\vartheta$  are discarded. Prior to performing the base optimization for

the start way-point of a tool-path, an additional option is to check whether the tool-path can be executed using the same AIR base pose used for the preceding tool-path.

### 5.3 Design Constraints

#### 5.3.1 Collision Avoidance Between AIRs

Let  $n_i^s$  spheres represent the  $i$ th AIR. Using spheres can be an efficient way of checking collision between AIRs. The number of spheres can be increased to obtain a more accurate representation of the AIR; however, at a cost to computation efficiency since collision checking would need to be performed for a larger number of spheres. An example sphere representation of the arm (manipulator) of an AIR is shown in Fig. 4. The spheres representing an AIR must not collide (overlap) with the spheres representing another AIR. Thus, the constraints are:

$$\|\mathbf{S}_{ip}(\theta_{j,i}) - \mathbf{S}_{i\acute{p}}(\theta_{j,i})\| - (S_{ip}^R + S_{i\acute{p}}^R) \geq 0 \quad (15)$$

$\forall i, i', p, \acute{p} : i = 1, 2, \dots, n; i' = 1, 2, \dots, n; i \neq i'; p = 1, 2, \dots, n_i^s; \acute{p} = 1, 2, \dots, n_{i'}^s$ , where  $\theta_{j,i}$  are a subset of design variable associated with the  $i$ th AIR,  $\mathbf{S}_{ip}(\theta_{j,i}) \in \mathbb{R}^3$  returns the centroid position of the  $p$ th sphere of the  $i$ th AIR based on the design variables  $\theta_{j,i}$  using forward kinematics [33], and  $S_{ip}^R$  is the corresponding sphere radius.

#### 5.3.2 Collision Avoidance Between the Links of Each AIR

Similar to the first constraint, to prevent self-collision the spheres representing an AIR's link must not collide with the spheres representing another link. Thus, the constraints are:

$$\|\mathbf{S}_{ir}(\theta_{j,i}) - \mathbf{S}_{i\acute{r}}(\theta_{j,i})\| - (S_{ir}^R + S_{i\acute{r}}^R) \geq 0 \quad (16)$$

$\forall i, r, \acute{r} : i = 1, 2, \dots, n; r = 1, 2, \dots, n_i^s; \acute{r} = 1, 2, \dots, n_i^s; r \neq \acute{r}$ , where  $\mathbf{S}_{ir}(\theta_{j,i}) \in \mathbb{R}^3$  returns the centroid position of the  $r$ th sphere of the  $i$ th AIR, and  $S_{ir}^R$  is the corresponding sphere radius.

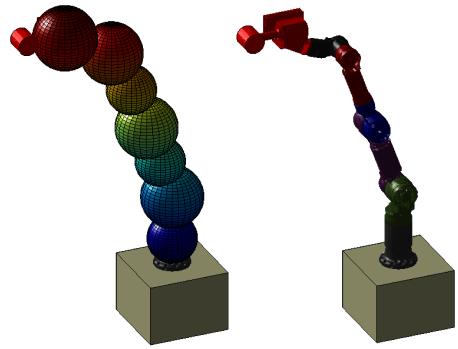


Fig. 4: An example where the manipulator of a 6 DoF industrial robot is represented as 7 spheres.

### 5.3.3 Collision Avoidance with the Environment

An AIR must not collide with any object in an environment. The environment can be represented as a point cloud; hence any point that represents part of the environment must not collide with, or penetrate into, any sphere that represents part of an AIR. Therefore, the constraints are:

$$\|\mathbf{o}_k^{\text{env}} - \mathbf{S}_{ir}(\theta_{j,i})\| - (S_{ir}^R) \geq 0 \quad (17)$$

$\forall k, i, r : k = 1, 2, \dots, n^k; i = 1, 2, \dots, n; r = 1, 2, \dots, n_i^s$ , where  $\mathbf{o}_k^{\text{env}}$  is the  $k$ th point representing part of an environment, and  $n^k$  is the total number of points representing the environment.

### 5.3.4 Compaction Force

To ensure proper attachment of the fiber onto a surface, an appropriate amount of compaction force must be exerted on the fiber which results in the deformation of the roller, as shown in Fig. 5. Let  $\mathbf{T}_i$  be a 4 by 4 homogeneous transformation matrix that represents the position and orientation of the point  $\mathbf{P}_i^{\text{COM}} \in \mathbb{R}^3$  shown in Fig. 5 which is located at the Center Of Mass (COM) of the  $i$ th AIR's roller. Note that the density of the undeformed roller should be uniform so that its COM is at its centroid. In Fig. 5,  $\varphi_i$  is the angle between the surface normal vector and the approach vector in the matrix  $\mathbf{T}_i$  (the vector parallel to the long axis of roller's bracket); and  $r_i$  is the radius of the roller attached to the  $i$ th AIR. Note that the parameters  $\mathbf{T}_i$ ,  $\mathbf{P}_i^{\text{COM}}$  and  $\varphi_i$  are a function of the design variables  $\theta_{j,i}$  (for brevity the notation  $\theta_{j,i}$  is not included), hence these parameters can be calculated using forward kinematics [33]. Let  $\mathbf{P}_i^c \in \mathbb{R}^3$  be a point on the surface of the roller, as shown in Fig. 5. The point  $\mathbf{P}_i^c$  can be calculated as:

$$\mathbf{P}_i^c = \mathbf{T}_i \times \text{Rot}(y, \varphi_i) \times \text{Trans}(0, 0, r_i) \quad (18)$$

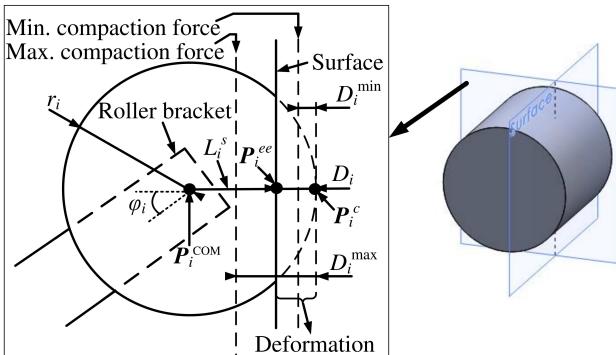


Fig. 5: Cross-section view of the roller, and the parameters related to the compaction force.

where  $\text{Rot}(y, \varphi_i)$  is a 4 by 4 homogeneous transformation matrix that represents a rotation of  $\varphi_i$  degrees about the  $y$ -axis of the roller (the axis joining the two ends of the roller), and  $\text{Trans}(0, 0, r_i)$  is a 4 by 1 matrix representing translation of magnitude  $r_i$  along the  $z$ -axis (i.e. along the approach vector of transformation matrix  $\mathbf{T}_i$ ). When the roller is in contact with the target surface with the appropriate position and orientation relative to the way-point  $\mathbf{o}_i(j)$  at step  $j$ , then the line  $L_i^s$  that connects  $\mathbf{P}_i^c$  to  $\mathbf{P}_i^{\text{COM}}$  is collinear with the line that connects  $\mathbf{o}_i(j)$  to  $\mathbf{P}_i^{\text{COM}}$ . As shown in Fig. 5, let  $\mathbf{P}_i^{\text{ee}} \in \mathbb{R}^3$  be another point on  $L_i^s$  at a distance  $D_i$  away from  $\mathbf{P}_i^c$  and towards  $\mathbf{P}_i^{\text{COM}}$ . The point  $\mathbf{P}_i^{\text{ee}}$  is defined as the end-effector point of the  $i$ th AIR.  $D_i$  defines the amount of deformation on the roller as shown in Fig. 5 and is calculated as  $D_i = \|\mathbf{o}_i(j) - \mathbf{P}_i^c\|$ . Hence,  $\mathbf{P}_i^c$  is always a distance  $r_i$  away from  $\mathbf{P}_i^{\text{COM}}$  irrespective of the deformation, whereas  $\mathbf{P}_i^{\text{ee}}$  accounts for the deformation that is defined using  $D_i$  and is in contact with  $\mathbf{o}_i(j)$  (when other constraints are met). The amount of compaction force  $F_i^c$  is relative to the deformation distance  $D_i$ , hence  $F_i^c = f(D_i)$  where  $f(D_i)$  is a linear or a non-linear function that calculates compaction force based on the distance  $D_i$ . An example of how  $F_i^c$  can be related to  $D_i$  is presented in [2]. Let  $D_i^{\min}$  and  $D_i^{\max}$  be the values of  $D_i$  that correspond to the minimum and maximum allowable compaction force, respectively, as shown in Fig. 5. Thus, the acceptable range of values for  $F_i^c$  and in turn for  $D_i$  need to be determined based on factors such as the type of composite materials used, the shape of the target structure, etc.

To ensure appropriate compaction force, the following constraints must be satisfied:

$$D_i^{\min} \leq D_i \leq D_i^{\max} \quad (19)$$

$$\forall i : i = 1, 2, \dots, n.$$

### 5.3.5 End-Effector Position Error

The previous constraint (compaction force) can't alone ensure that the end-effector point  $\mathbf{P}_i^{\text{ee}}(\theta_{j,i}) \in \mathbb{R}^3$  is at the desired way-point  $\mathbf{o}_i(j) \in \mathbb{R}^3$  at step  $j$ . Hence, the following constraints must be satisfied:

$$\|\mathbf{P}_i^{\text{ee}}(\theta_{j,i}) - \mathbf{o}_i(j)\| \leq \delta \quad (20)$$

$$\forall i : i = 1, 2, \dots, n, \text{ where } \delta \text{ is a small acceptable deviation.}$$

### 5.3.6 End-Effector Orientation Error

The orientation of the roller attached to the tip of an AIR must meet the desired orientation within a predefined error. Let  $\phi_i^r(\theta_{j,i})$ ,  $\phi_i^p(\theta_{j,i})$  and  $\phi_i^y(\theta_{j,i})$  be the roll, pitch and yaw angles that define the orientation of the roller attached to the  $i$ th AIR. These angles can be calculated for an AIR pose (defined by AIR joints' angles and base pose) using forward kinematics [33]. The difference between these angles and

the angles that define the desired orientation must be within predefined values. Thus, the following constraints must be satisfied:

$$\left| \phi_i^r(\theta_{j,i}) - \phi_i^{r,d}(j) \right| \leq \phi_i^{r,\max} \quad (21a)$$

$$\left| \phi_i^p(\theta_{j,i}) - \phi_i^{p,d}(j) \right| \leq \phi_i^{p,\max} \quad (21b)$$

$$\left| \phi_i^y(\theta_{j,i}) - \phi_i^{y,d}(j) \right| \leq \phi_i^{y,\max} \quad (21c)$$

$\forall i : i = 1, 2, \dots, n$ , where the superscripts  $r, p, y$  represent the roll, pitch, and yaw angles, respectively,  $\phi_i^{\min}$  is the maximum allowable angle error, and the superscript  $d$  represents the desired rotation angle. Note that these errors should be very small (less than or equal to  $1^\circ$  in [5]). However, the fiber placement head can be designed such that it accommodates for larger error in the pitch angle. This is because the roller can be made to rotate along a tool-path even if the heading of the fiber placement head is not completely perpendicular to the surface.

### 5.3.7 Joint Limits

The angle of joint  $q$  of the  $i$ th AIR,  $\theta_{j,iq}$  should be within its limits,  $\theta_{iq}^{\min}$  and  $\theta_{iq}^{\max}$ . Hence, the constraints are:

$$\theta_{iq}^{\min} \leq \theta_{j,iq} \leq \theta_{iq}^{\max} \quad (22)$$

$$\forall i, q : i = 1, 2, \dots, n; q = 1, 2, \dots, n_i^q.$$

## 6 Optimization for the Two Stages

Using the developed mathematical models, the two stages can be solved using many optimization algorithms. In this work standard optimization algorithms such as multi-objective Genetic Algorithm (GA) and gradient optimization are used; hence, not to be considered as a contribution. Thus, the main aim of this section is to demonstrate how the two stages can be solved using existing optimization algorithms.

### 6.1 Optimization for Stage 1 (Tool-path Allocation)

For the tool-path allocation problem (Stage 1), it is vital to obtain a solution as close to the optimal as possible since the objectives used in the mathematical model are critical (e.g. achieving minimal makespan is important) and the performance of the second stage (multi-AIR path planning) is dependent on the output of the first stage. Thus, a global optimization algorithm such as a GA-based optimizer is an appropriate choice as it carries out a wider search than a local optimizer (which depends mostly on a good initial guess).

As an option for optimizing the tool-path allocation (Stage 1), the Multiple Traveling Salesman Problem (MTSP) can

be looked at. In MTSP, a number of salesmen are assigned the task of visiting a number of cities. The task usually involves visiting the cities as quickly as possible and without repeating a visit, however, there are different variations of this problem. In the tool-path allocation problem, the AIRs can represent the salesmen, and the tool-paths can represent the cities that the AIRs need to visit/execute. However, the objectives and constraints used in the mathematical model for tool-path allocation are explicitly designed for the fiber placement problem.

MTSP is considered to be NP-hard [34]. Thus, many researchers have resorted to using metaheuristic optimization techniques such as GA. In order to use GA-based optimization algorithms, an appropriate chromosome representation must be devised for the particular problem under consideration. The work in [35] implements a two-part chromosome representation for solving the MTSP problem. They prove that the two-part chromosome representation reduces redundant solutions and the size of the search space.

In this work, two-part chromosome representation [35] is used; however, some modifications are made to account for the layers and prerequisite tool-paths. Fig. 6 shows an example two-part chromosome representation designed for the specific problem under consideration. The first part of the chromosome shows the permutation of the tool-path indices for all layers. The length of the first part is  $N_Z$  (i.e. the number of tool-paths in all layers). The second part of the chromosome shows the number of tool-paths assigned to each AIR per layer. The length of the second part is  $n \times Z^{\text{tot}}$  (i.e. the number of AIRs multiplied by the number of layers). Based on the example, the first AIR is assigned tool-paths 1 and 2 from layer 1; tool-paths 4, 1, and 2 from layer 2; and tool-paths 2 and 4 from the last layer. The executing sequence of the tool-paths by each AIR is from left to right.

The advantage of this chromosome representation, besides the fact that it reduces redundant solutions, is that it will satisfy the constraints mentioned in Section 4.3. It will ensure feasibility of the solution by enforcing each AIR to execute tool-paths from the first layer to the last (left to right in the chromosome). If a tool-path assigned to an AIR is a prerequisite of another AIR's tool-path, then the prerequisite tool-path must be executed first, which may cause some instances of idle times for a number of AIRs. However, since one of the objective functions is to reduce the makespan which includes idle time, then minimizing makespan will also minimize idle times. The ultimate goal is to arrive at a chromosome that results in the optimal solution for the tool-path allocation problem.

Note that since multiple conflicting objectives are used in the model, then multi-objective GA is used. More specifically, the function *gamultiobj* from the MATLAB optimization toolbox which is based on Non-dominated Sorting Genetic Algorithm II (NSGA-II) [36] is used.

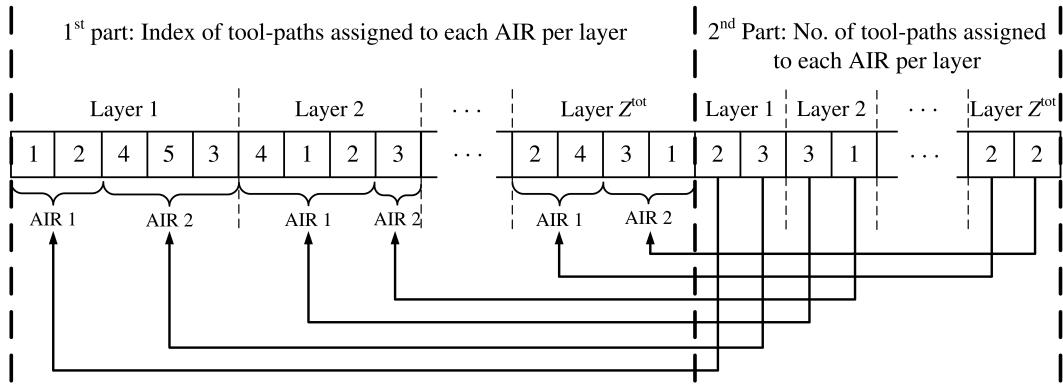


Fig. 6: An example of the implemented two-part chromosome representation.

Let  $\Omega$  be the space in which the design variables coexist. Similarly, let  $\Psi$  be the objective space in which objective functions coexist. Therefore, a point in  $\Omega$  maps to a point in  $\Psi$ . The aim is to obtain Pareto optimal set,  $P^* \in \Omega$ . The solution vectors in  $P^*$  are non-dominated, i.e.

$$P^* := \{x \in \Omega \mid \nexists y \in \Omega \text{ such that } y \succ x\} \quad (23)$$

where  $\succ$  represents dominance. A solution  $x$  dominates a solution  $y$  (i.e.,  $x \succ y$ ) if and only if for all of the objective functions,  $x$  doesn't yield a value worse than  $y$  in  $\Psi$  space and it strictly yields a better value for at least one of the objectives [37]. The corresponding Pareto optimal front ( $PF^*$ ) in  $\Psi$  space is:

$$PF^* := \{F(x) \in \Psi, \forall x \in P^*\} \quad (24)$$

where  $F(x)$  returns the values of the objective functions for a solution  $x$ .

GA-based algorithms require crossover and mutation operators. The crossover operator implemented in this work is based on two-part chromosome crossover (TCX) introduced by Yuan et al. [34]. TCX is made specifically for the two-part chromosome representation, and it is proven to improve solution quality and increase diversity in the second part of the chromosome. As for the mutation operator, it is based on swap mutation which takes a random set of pairs of genes and exchanges the location of genes in each pair. However, each pair is selected from one layer only so as to keep the feasibility of the solution.

## 6.2 Optimization for Stage 2 (Multi-AIR Path Planning)

For the multi-AIR path planning problem, the aim is to find AIR poses at each step such that the constraints outlined in Section 5.3 are satisfied. Using a local optimizer to find solutions that satisfy all constraints is favorable as it provides a

fast computation performance which is helpful since the optimization needs to be repeated for a large number of waypoints (i.e., all way-points for all tool-paths). However, a local optimizer may not find a globally optimum solution in terms of the objective of minimal change of manipulators' joints angles (Eq. (13)). If this objective is critical for an application, then a global optimizer such as GA might be used at a significant cost to computational efficiency as will be discussed in Section 8.

The *fmincon* function from the MATLAB optimization toolbox is used. Since *fmincon* is a local optimizer it may not find a globally optimum solution. This issue can be moderately addressed by selecting a good initial guess for the first way-point of each tool-path. Hence, for the first way-point of each tool-path, GA is used for the following reasons: (1) no initial guess is needed for GA, (2) GA can search more broadly through the space and is a better method for analyzing whether or not a feasible solution exists, and (3) when the model includes determining base positions (Section 5.2.2), GA performs better in finding a feasible solution. For the remaining way-points of a tool-path, the solution from the previous way-point is used as the initial guess for the next. It is important to note that from a practical point of view, at each step, finding a quick solution that reasonably minimizes the change in joints angles and produces a feasible solution (satisfies all the constraints) is more critical than finding a solution that may be globally optimum but takes a relatively long time to obtain.

## 7 Case Studies

Case studies are presented in this section to demonstrate the effectiveness of the approach for different scenarios and conditions. In brief, the following aspects are investigated:

- Case Study 1:
  - Fiber placement on a curved surface
  - Comparison of three solutions from the Pareto front

- A video to show the results
- Achieving optimal makespan
- Checking consistent convergence of the results
- Case Study 2:
  - Fiber placement on a dome-shaped surface
  - Comparisons in terms of makespan
  - A video to show the results
  - Checking consistent convergence of the results
- Case Study 3:
  - Repeating Case Study 2 with fixed AIR bases to investigate the effect of Objective 4 in Eq. (9)
  - Investigating the effect of Objective 3 (minimal distance between consecutive tool-paths - Eq. (7))
  - Investigating the effect of Objective 2 (maximal distance between AIRs' end-effectors - Eq. (5))
  - Investigating the effect of Objective 1 (minimal makespan - Eq. (2))
  - Investigating the performance of the approach with respect to the number of AIRs (2))

Optimization for Stage 1 (tool-path allocation) was performed using the function *gamultiobj* from the MATLAB optimization toolbox. The function *gamultiobj* is based on NSGA-II. Optimization for Stage 2 was mainly done through the *fmincon* function from the MATLAB optimization toolbox. However, as explained in Section 6.2, genetic algorithm (the function *ga* from the MATLAB optimization toolbox) was used for the first way-point of each tool-path. Default optimization parameters were found to be appropriate for the problem under consideration. Optimization was carried out using a PC with Intel Core i5-2400 CPU @ 3.10 GHz. Parallel processing was enabled to run the code using all four cores of the CPU.

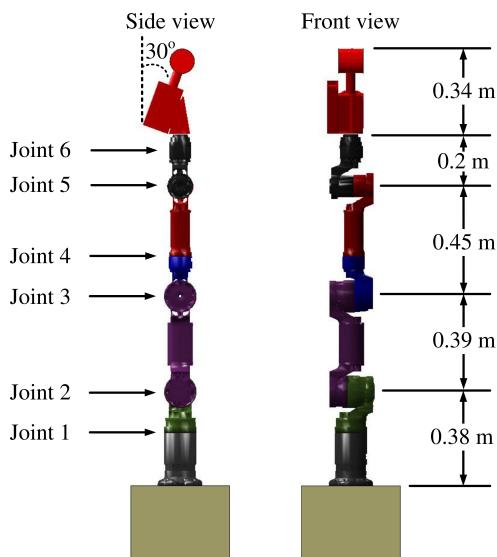


Fig. 7: The AIR used and its dimensions.

Table 1: Parameters and their values.

Parameter	Value	Reference
$v_i$	0.1 m/s	Eq. (3)
$\omega_i$	1	Eq. (3)
$\theta_{iq}^{\text{cap}}$	$1 \forall q$	Eq. (13)
$D_i^{\text{min}}$	0 m	Eq. (19)
$D_i^{\text{max}}$	0.02 m	Eq. (19)
$\delta$	0.0001 m	Eq. (20)
$\phi_i^{r,\text{max}}$	0	Eq. (21)
$\phi_i^{p,\text{max}}$	$45^\circ$	Eq. (21)
$\phi_i^{y,\text{max}}$	0	Eq. (21)

The AIR used in the case studies (shown in Fig. 7) consists of a 6 DoF manipulator mounted on a base platform, and a roller attached to the fiber placement head. The manipulator is modeled upon a real 6 DoF Schunk manipulator. The base is considered to be mobile and omnidirectional, and it is simulated as a  $0.4 \text{ m} \times 0.4 \text{ m} \times 0.3 \text{ m}$  cuboid. The radius and the height of the roller are 0.05 m and 0.1 m, respectively. A total of 6 tows are laid down simultaneously, each tow being 12.7 mm in width which gives a total of 76 mm for all 6 tows. The height from the top surface of the base to the centroid of the roller is 1.74 m. The base of the AIR can't move vertically, neither can it rotate about the x or y axes. The first joint of the AIR, which is closest to the base and rotates about the z-axis, can do a full  $360^\circ$  rotation. The joint limits used in Eq. (22) are  $\pm 180^\circ$ ,  $\pm 110^\circ$ ,  $\pm 105^\circ$ ,  $\pm 175^\circ$ ,  $\pm 102^\circ$ , and  $\pm 175^\circ$  for joints 1 to 6, respectively.

Table 1 lists the value of the parameters used with reference to the corresponding equation. The values are the same for all AIRs (i.e. for all  $i$ ).

## 7.1 Case Study 1: Fiber Placement of a Curved Surface Using Two Mobile AIRs

In this case study, the effectiveness of the overall approach is tested on a curved surface as shown in Fig. 8. Two mobile AIRs are deployed for the fiber placement task (shown in Fig. 9) and they were able to execute all tool-path in all layers while satisfying all constraints. Note that the number of way-points for each tool-path is shown in Fig. 8. A video<sup>1</sup> is provided to show the allocation of tool-paths amongst the AIRs and the output of the collaborative path planner.

For the tool-path allocation (Stage 1), a solution is chosen from the Pareto front. Table 2 shows the values of Objective 1 (makespan based on Eq. (2)), Objective 2 (distance between AIRs' end-effector based on Eq. (5)), and Objective 3 (sum of distances between consecutive tool-paths based on Eq. (7)) for the chosen solution. The table also shows the three best solutions from the Pareto front in terms of Objec-

<sup>1</sup> The video can be viewed at [https://youtu.be/f9SkzJ0z\\_04](https://youtu.be/f9SkzJ0z_04)

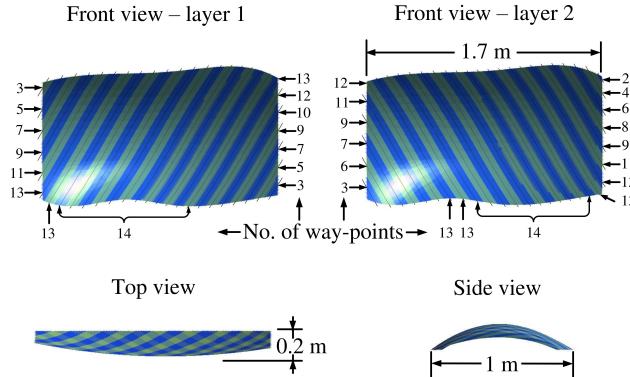


Fig. 8: A curved surface, its dimensions, the tool-paths for each layer, and the number of way-points for each tool-path.

tives 1 to 3. Objective 4 is discarded since AIRs' bases are considered to be mobile (or can be moved manually).

In industrial applications, achieving minimal makespan is vital for productivity. One AIR can execute all tool-paths in all layers with a makespan of 1562 s considering no idle time; thus the optimal solution in terms of makespan for two AIRs is 781 s ( $1562 / 2$ ). Hence, as shown in Table 2, an optimal solution in terms of makespan could be achieved using the proposed mathematical model. However, a solution is chosen that is acceptable in terms of all three objectives with a makespan being worse than the optimal by 6 % (44 s).

The following parameters are chosen specifically for this case study: (1) the value of  $t_i^t$  in Eq. (3) is 19.66 s, (2) the bottom surface of the base platform of the AIRs is 0.06 m lower than the bottom of the target structure, (3) layers 1 and 2 include 26 and 27 tool-paths, respectively, and (4) distance between two way-points along a tool-path is 0.09 m.

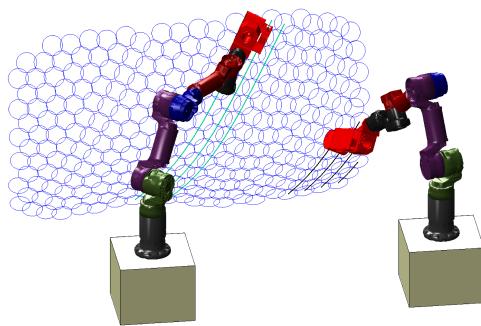


Fig. 9: Two AIRs operating on the curved surface.

Table 2: Solutions from the Pareto front for Case Study 1.

	Obj. 1	Obj. 2	Obj. 3
Chosen solution	825 s	0.85 m	6.41 m
Best in terms of Objective 1	781 s	0.55 m	11.92 m
Best in terms of Objective 2	1312 s	1.66 m	22.10 m
Best in terms of Objective 3	918 s	0.53 m	5.98 m

Figure 10 is generated to show the convergence of the Pareto front over the generations and that the final Pareto front is close to the Pareto optimal front,  $\text{PF}^*$  (explained in Eq. (24)). To show an accurate plot of the convergence, Fig. 10 is constructed using 30 independent optimization runs. More specifically, Fig. 10 is constructed as follows. At first, since the  $\text{PF}^*$  is not known, a reference set is used to approximate the  $\text{PF}^*$  [38] by running the optimization 30 times with a large population (10,000) and a large generation number (10,000). The non-dominated solutions (explained in Section 6.1) from the Pareto fronts of all 30 optimization runs are then used to approximate  $\text{PF}^*$ . Once  $\text{PF}^*$  is obtained, the optimization is repeated 30 times with the default parameters of MATLAB optimization toolbox. Then the Inverted Generational Distance (IGD) [38], which is becoming increasingly popular in recent years, is used to measure how far the Pareto front is from  $\text{PF}^*$  and how it evolves over the generations. The IGD value is calculated as:

$$\text{IGD}(\text{PF}, \text{PF}^*) = \frac{1}{|\text{PF}^*|} \sqrt{\sum_{\mathbf{a} \in \text{PF}^*} \left( \min_{\mathbf{b} \in \text{PF}} \|\mathbf{a} - \mathbf{b}\|^2 \right)} \quad (25)$$

where  $\|\mathbf{a} - \mathbf{b}\|$  is the Euclidean distance between  $\mathbf{a} \in \text{PF}^*$  and  $\mathbf{b} \in \text{PF}$  ( $\text{PF}$  is the Pareto front). Note that  $\text{PF}^*$  and  $\text{PF}$  are normalized using the maximum and minimum values in  $\text{PF}^*$ . A smaller value of IGD indicates a closer distance of  $\text{PF}$  to  $\text{PF}^*$ .

Figure 10 shows that the IGD value does converge and that it is improved by 95 % from the first generation to the last. The IGD value of the 30 optimization runs is calculated at each generation using the current Pareto front at that particular generation, and then the average of the IGD values is taken at each generation to construct the figure. The low IGD value indicates that the Pareto front is close to the  $\text{PF}^*$ . However, increasing the population size can improve the IGD value at the expense of computational efficiency.

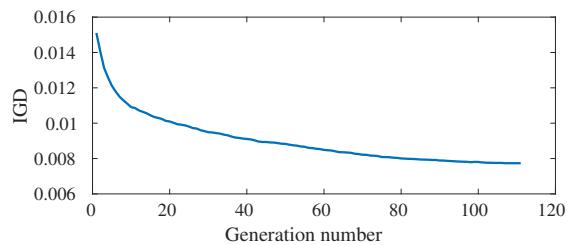


Fig. 10: Convergence of the Pareto front with respect to the Pareto optimal front ( $\text{PF}^*$ ) using the average of 30 independent optimization runs. The Inverted Generational Distance (IGD) is used to measure how far the Pareto front is from the  $\text{PF}^*$  at each generation. The IGD value is improved by 95 % from the first generation to the last.

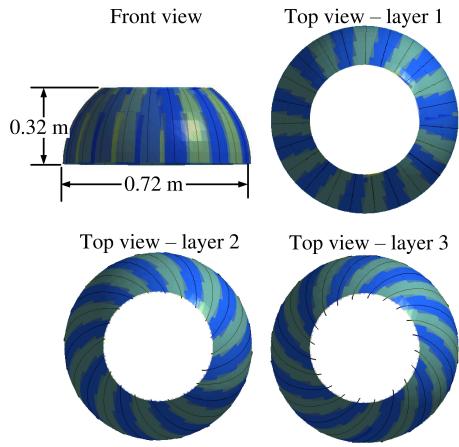


Fig. 11: A dome-shaped surface, its dimensions, and the tool-paths for each layer.

## 7.2 Case Study 2: Fiber Placement of a Dome-Shaped Surface Using Two Mobile AIRs

The purpose of this case study is to test the overall approach again using a different surface, i.e. the dome-shaped surface shown in Fig. 11. Two mobile AIRs are deployed to perform the fiber placement task as shown in Fig. 12. A video<sup>2</sup> is provided to show the results of the tool-path allocation and the multi-AIR collaborative path planning. The chosen solution from the Pareto front has values of 1009 s for Objective 1, 0.41 m for Objective 2, and 9.75 m for Objective 3. In terms of makespan, the chosen solution is 3 % (27 s) worse than the optimal makespan (982 s). Solutions with optimal makespan exist in the Pareto front, however with worse values for Objectives 2 and 3 when compared to the chosen solution. For example, a solution with optimal makespan has values of 0.17 m and 13.8 m for Objectives 2 and 3, respectively. Thus, if the makespan is allowed to be slightly worse

<sup>2</sup> The video can be viewed at <https://youtu.be/jBGh0d1Few>

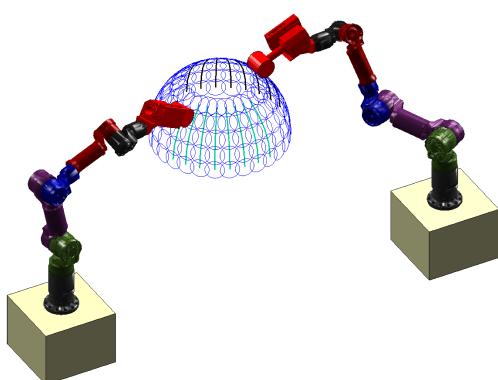


Fig. 12: Two AIRs operating on the dome-shaped surface.

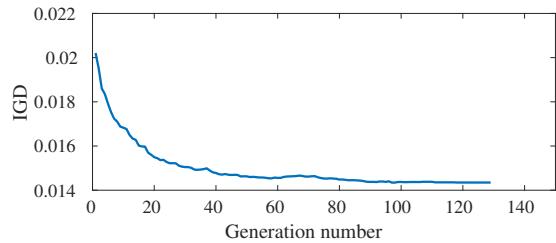


Fig. 13: Convergence of the Pareto front with respect to the Pareto optimal front ( $\text{PF}^*$ ) using the average of 30 independent optimization runs. The Inverted Generational Distance (IGD) is used to measure how far the Pareto front is from the  $\text{PF}^*$  at each generation. The IGD value is improved by 42 % from the first generation to the last.

than the optimal makespan, then solutions that are better in terms of Objectives 2 and 3 can be obtained from the Pareto front which will then help with flexibility and maneuverability during multi-AIR path planning (Stage 2).

The following parameters are chosen specifically for this case study: (1) the value of the parameter  $t_i^t$  in Eq. (3) is 19.72 s; (2) the bottom surface of the base platform of the AIRs is 0.8 m lower than the bottom of the target structure; (3) layers 1, 2 and 3 include 29, 25 and 26 tool-paths, respectively; (4) the number of way-points for all tool-paths in layers 1, 2 and 3 are 5, 6 and 6, respectively; and (5) the distance between two way-points along a tool-path is 0.08 m.

Similar to Case Study 1, Fig. 13 is constructed using 30 independent optimization runs to show that the Pareto front consistently converges over the generations. The IGD value is improved by 42 % from the first generation to the last. The low IGD value indicates that the Pareto front is close to the  $\text{PF}^*$ . Case Study 1 (Section 7.1) and Eq. 25 explain the details of how the IGD value is calculated and how the plot in the figure is generated.

## 7.3 Case Study 3: Comparative Studies

This case study provides comparative studies through: (1) AIRs with and without mobile bases, (2) different combinations of the objective functions in the model, and (3) considering up to 6 AIRs.

**Comparison 1 - considering fixed AIR base:** First, the scenario in Case Study 2 (Section 7.2) is repeated, however now considering immobile (fixed) base for the AIRs. Thus, the fourth objective expressed in Eq. (9), which is maximal reachability through minimizing the distance of the allocated tool-paths to an AIR, is included. For comparison sake, the third objective is discarded. The fixed base position of the AIRs relative to the surface is shown in Fig. 14. A solution from the Pareto front is chosen that has values of

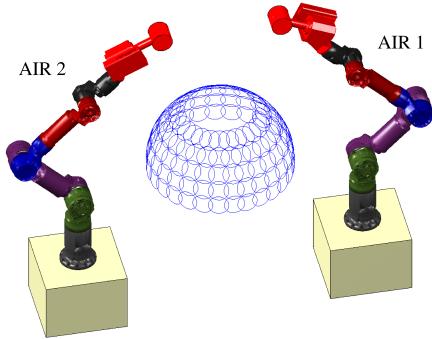


Fig. 14: Location of the two AIRs relative to the dome-shaped surface.

1057 s for Objective 1, 0.32 m for Objective 2, and 2328 m for Objective 4. To compare this solution to a solution from the mathematical model that doesn't include Objective 4, optimization is repeated without including Objective 4 and a solution is chosen from the Pareto front. The selected solution has values of 983 s for Objective 1 and 0.25 m for Objective 2. The benefit of including Objective 4 is that the allocation of the tool-paths is more appropriate for the condition where the AIRs' bases are fixed. That is, as shown in Fig. 15, the way-points for all tool-paths in all layers allocated to each AIR are closer to the corresponding AIR when compared to the solution that doesn't consider Objective 4.

**Comparison 2 - the effect of Objective 3:** Objective 1 is critical since in manufacturing applications it is vital to minimize the makespan for increased productivity. Objective 2 is also critical since it will help the AIRs avoid collisions or the situation where one AIR hinders the motion of another AIR due to the close proximity of the end-effectors. However, Objective 3 is not as critical since it is mainly concerned with reducing the base motion of the AIRs and saving energy. Nonetheless, including this objective is important and beneficial. As a comparison, Case Study 2 is repeated by discarding Objective 3. A solution with values of 983 s for

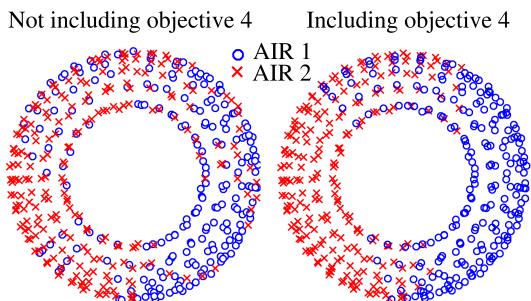


Fig. 15: Top view of the way-points in all tool-path of all layers associated with each AIR, based on the presented solutions in Case Studies 2 and 3.

Objective 1 and 0.25 m for Objective 2 is used. Objective 3 is calculated for this solution, and it was found that the value of Objective 3 is 24.26 m, which is 14.51 m (150 %) larger than the value of the solution obtained in Case Study 2. This significant increase in Objective 3 causes greater base motion of the AIRs since there are fewer instances where consecutive tool-paths can be executed at the same base position (due to the larger distance between consecutive tool-paths), hence the AIRs may have to move more often when executing the tool-paths.

**Comparison 3 - the effect of Objective 2:** Next, Case Study 2 is repeated by only considering Objective 1 (makespan). A solution with the optimal makespan of 982 s is obtained. This solution is then recalculated for Objective 2 to investigate how it compares to the chosen solution in Case Study 2 (where Objective 2 was considered). It was found that the value of Objective 2 for the obtained solution is 0.05 m (which is 0.2 m worse than the chosen solution in Case Study 2), meaning that the end-effectors of the AIRs will be very close to each other at some stage during their operation. This could cause path-planning to halt since the two AIRs may not be able to find collision-free poses.

**Comparison 4 - the effect of Objective 1:** Case Study 2 is repeated by only considering Objective 2 (maximal distance between AIRs' end-effectors). A solution with a value of 0.42 m for Objective 2 was obtained. This solution is then recalculated while accounting for Objective 1 (makespan). It was found that the makespan is 1367 s which is 384 s (28 %) worse than the optimal makespan (982 s).

**Comparison 5 - performance with respect to the number of AIRs:** The aim here is to compare and discuss the performance of the approach with respect to the number of AIRs, particularly by investigating the effect on the makespan objective (E.q. (2)) which is of high importance for productivity. Case studies 1 and 2 are repeated by considering up to 6 AIRs. The solution with the best makespan is selected from the Pareto front for each scenario. Figure 16 shows the actual makespan and the ideal makespan for each scenario. Since it is difficult to calculate the true optimal makespan, a lower bound on the optimal makespan (ideal makespan) is calculated as the time it takes for a single AIR to cover all tool-paths (assuming no idle time) divided by the number of AIRs. The time it takes for a single AIR to cover all tool-path of the surfaces presented in Case Studies 1 and 2 is 1562 s and 1964 s, respectively. From Fig. 16 it is evident that the makespan decreases as more number of AIRs are used; however, the difference to the ideal makespan gradually increases as well. The small gradual increase with respect to the ideal makespan is possibly due to constraints

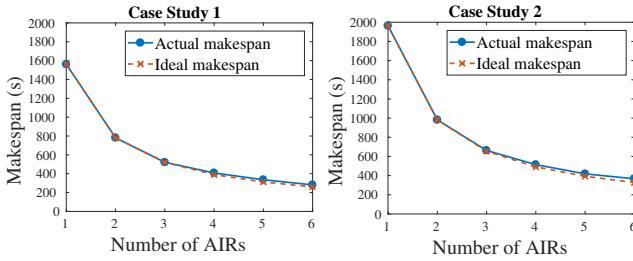


Fig. 16: Actual makespan and ideal makespan considering up to 6 AIRs.

associated with the problem, such as the constraint of satisfying the prerequisite tool-paths which can become harder to satisfy as the number of AIRs increase. Note that the ideal makespan was calculated naively by simply dividing the optimal makespan of a single AIR by the number of AIRs, which may cause an over-optimistic value than the true optimal makespan since it doesn't take into account the constraints of the problem.

Note that if the object under consideration is small, which is the case in the presented case studies, then using more than two AIRs may not be a reasonable option. Although Fig. 16 shows that having more than two AIRs can reduce the makespan, Objective 2 (E.q. (5)) and Objective 3 (E.q. 7) are significantly affected. For example, for the scenario where Case Study 1 is repeated with 3 AIRs, although the makespan is optimal (as shown in the figure), the result of Objective 2 indicates that the end-effectors of the AIRs get to 0.083 m proximity to each other which is 10 times worse than the value of the solution chosen for the case where 2 AIRs were deployed (refer to Table 2).

Relatively large objects may require the deployment of more than three AIRs. However, the tool-paths in large objects are likely to be longer and the assumption that “the base needs to be fixed (immobile) when executing the fiber placement task for a tool-path” is no longer reasonable. Thus, the aim of this particular study was to check the performance of the overall approach when more than two AIRs are deployed; however, the extension of current work for larger objects and considering a mobile base during task execution will be investigated as future work.

## 8 Discussion and Future Work

A two-stage approach to multi-AIR collaborative fiber placement was presented and then tested using several case studies. It is evident from the case studies that the approach can perform well within the scope of the work considered. The assumptions made and the scope of the work was set-out in the problem definition (Section 3) to clarify the advantages and limitations of the approach.

### 8.1 Computation Time

A brief discussion on the computation efficiency of the approach is presented here. On average, the computation time of Stage 1 (tool-path allocation) for Case Studies 1 and 2 is 6 minutes and 8 minutes, respectively. These computation times are acceptable since tool-path allocation is performed off-line. However, there is room for significant improvements since the primary focus of the work in the paper has been on aiming to achieve optimal collaboration and productivity of the AIRs rather than the off-line computational efficiency. Improving the computation time will particularly be of interest when the presented approach is investigated for on-line applications. The period associated with repositioning of AIRs' bases and getting the fiber ready for a new tool-path can be used to perform various computations. Hence, reducing the off-line computation time to fit within this period may enable on-line applications.

In regards to the multi-AIR path planning, on average, it took less than 0.1 s to find a feasible solution at each step using the *fmincon* function of MATLAB optimization toolbox. Since path planning is performed off-line, then this efficiency is acceptable; however, there is room for improvement. For example by further tuning of the optimization parameters, comparing different data structures [39] such as Quadtrees and Octrees for distance queries related to Eq. (16), etc. As mentioned in Section 6.2, GA is used for the first way-point of each tool-path. On average, GA takes less than 5 s to find a feasible solution.

### 8.2 Future Work

As the first work in multi-AIR collaborative fiber placement, the focus has been in developing mathematical models that specifically solve the problem defined in Section 3. The developed models can be of great contribution for scenarios such as those shown in the case studies; however, it will be interesting to study whether there are potentials to apply the models to wider variations of the fiber placement problem. For example, below are some potential improvements, studies, and comparisons that can be carried out as future work:

- Comparing the proposed two-stage approach with a single-stage approach where the models for tool-path allocation and multi-AIR path planning are combined. The single-stage approach may improve optimality, however, may be at a significant cost to computational efficiency.
- Modifying the models such that the AIRs base are allowed to move while operating on a tool-path. For larger and more complex objects, it may not be possible to execute an entire tool-path with a fixed AIR base.
- Investigating the practicality of placing the structure on a spindle to improve AIRs' flexibility and reachability.

- Carrying out a detailed study on computation complexity and time efficiency of the approach, e.g. by tuning the parameters in the optimization algorithms. As discussed previously, reducing the computational time may help with enabling the approach for on-line application.
- Incorporating other objectives such as energy minimization, torque minimization, and manipulability maximization and analyzing their effect on the performance.
- Estimating the time it takes for an AIR to move from one base position to the next while performing the optimization for the first stage of the approach so as to improve optimality. That is, estimating the value of  $t_i^l$  in Eq. (3) rather than using a constant value that is conservative.

## 9 Conclusion

Two challenges related to multi-AIR collaborative fiber placement were addressed in this paper. The first challenge is concerned with tool-path allocation which is defined as the problem of allocating a subset of tool-paths to each AIR and simultaneously determining the visiting sequence of the allocated tool-paths for each AIR. A tool-path is a path that the roller attached to the end-effector of an AIR needs to follow so as to lay the fiber. The second challenge, which is multi-AIR path planning, involves finding feasible AIR poses (inverse kinematics) for all way-points of a tool-path while considering the coordination of the AIRs when executing the fiber placement task. To solve these challenges, a two-stage approach was presented where Stage 1 is concerned with tool-path allocation and Stage 2 is concerned with multi-AIR path planning. Stage 1 of the approach (tool-path allocation) consists of a mathematical model which considers objectives related to productivity (e.g. minimal makespan) and those related to multi-AIR collaboration (e.g. maximal distance between the end-effectors' of the AIRs). The output from Stage 1, which is the tool-paths allocated to each AIR, is then used in the multi-AIR path planner (Stage 2). The multi-AIR path planner consists of a mathematical model that considers fiber placement related constraints such as compaction force and end-effector roller's orientation, but also general constraints such as collision avoidance and AIR joints' limits. The objective of the multi-AIR path planner is to achieve a smooth motion (by minimizing the change in joints' angles at each step) whilst satisfying all constraints. Several case studies were presented to prove the effectiveness of the approach for multi-AIR fiber placement.

## 10 Acknowledgments

This work is supported by the Centre for Autonomous Systems (CAS) at the University of Technology Sydney (UTS).

## References

1. Shirinzadeh, B., Cassidy, G., Oetomo, D., Alici, G., Ang, M. H.: Trajectory generation for open-contoured structures in robotic fibre placement. *Robotics and Computer-Integrated Manufacturing*, 23(4), 380 – 394 (2007)
2. Shirinzadeh, B., Alici, G., Foong, C. W., Cassidy, G.: Fabrication process of open surfaces by robotic fibre placement. *Robotics and Computer-Integrated Manufacturing*, 20(1), 17 – 28 (2004)
3. Hassan, M., Liu, D., Paul, G.: Modeling and stochastic optimization of complete coverage under uncertainties in multi-robot base placements. In: International Conference on Intelligent Robots and Systems (IROS), pp. 2978–2984 (2016)
4. Hvilsted, M., Bogh, S., Skov Nielsen, O., Madsen, O.: Autonomous industrial mobile manipulation (AIMM): past, present and future. *Industrial Robot: An International Journal*, 39(2), 120–135 (2012)
5. Debout, P., Chanal, H., Duc, E.: Tool path smoothing of a redundant machine: Application to automated fiber placement. *Computer-Aided Design*, 43(2), 122 – 132 (2011)
6. Li, L., Xu, D., Wang, X., Tan, M.: A survey on path planning algorithms in robotic fibre placement. In: Chinese Control and Decision Conference (CCDC), pp. 4704–4709 (2015)
7. Hu, B., Xu, D.-l.: Fiber placement path planning for open surfaces based on traversal method. *Fiber Reinforced Plastics*, 6, 005 (2014)
8. Yan, L., Chen, Z. C., Shi, Y., Mo, R.: An accurate approach to roller path generation for robotic fibre placement of free-form surface composites. *Robotics and Computer-Integrated Manufacturing*, 30(3), 277 – 286 (2014)
9. Xiaoping, W., Luling, A., Liyan, Z., Laishui, Z.: Uniform coverage of fibres over open-contoured freeform structure based on arc-length parameter. *Chinese Journal of Aeronautics*, 21(6), 571 – 577 (2008)
10. Bruyneel, M., Zein, S.: A modified fast marching method for defining fiber placement trajectories over meshes. *Computers & Structures*, 125, 45 – 52 (2013)
11. Li, L., Wang, X., Xu, D., Tan, M.: Path planning of airfoil surface for robotic fibre placement. In: 2015 International Conference on Advanced Mechatronic Systems (ICAMechS), pp. 316–321 (2015)
12. Aized, T., Shirinzadeh, B.: Robotic fiber placement process analysis and optimization using response surface method. *The International Journal of Advanced Manufacturing Technology*, 55(1), 393–404 (2011)
13. Jeffries, K. A.: Enhanced robotic automated fiber placement with accurate robot technology and modular

- fiber placement head. *SAE International Journal of Aerospace*, 6(2013-01-2290), 774–779 (2013)
14. Zhang, X., Xie, W.-F., Hoa, S. V.: Modeling and workspace analysis of collaborative advanced fiber placement machine. In: ASME 2014 International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers (2014)
  15. Alatartsev, S., Stellmacher, S., Ortmeier, F.: Robotic task sequencing problem: A survey. *Journal of Intelligent & Robotic Systems*, 80(2), 279–298 (2015)
  16. Li, J., Meng, X., Zhou, M., Dai, X.: A two-stage approach to path planning and collision avoidance of multibridge machining systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, PP(99), 1–11 (2016)
  17. Han, X., Bui, H., Mandal, S., Pattipati, K. R., Kleinman, D. L.: Optimization-based decision support software for a team-in-the-loop experiment: Asset package selection and planning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(2), 237–251 (2013)
  18. Han, X., Mandal, S., Pattipati, K. R., Kleinman, D. L., Mishra, M.: An optimization-based distributed planning algorithm: A blackboard-based collaborative framework. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(6), 673–686 (2014)
  19. Sariel-Talay, S., Balch, T. R., Erdogan, N.: A generic framework for distributed multirobot cooperation. *Journal of Intelligent & Robotic Systems*, 63(2), 323–358 (2011)
  20. Gasparetto, A., Boscaroli, P., Lanzutti, A., Vidoni, R.: Trajectory planning in robotics. *Mathematics in Computer Science*, 6(3), 269–279 (2012)
  21. Latombe, J.-C.: Robot motion planning, vol. 124. Springer Science & Business Media (2012)
  22. Chotiprayanakul, P., Liu, D., Wang, D., Dissanayake, G.: A 3-dimensional force field method for robot collision avoidance in complex environments. In: International Symposium on Automation and Robotics in Construction (ISARC), pp. 19–21 (2007)
  23. To, W. K., Paul, G., Kwok, N. M., Liu, D.: An efficient trajectory planning approach for autonomous robots in complex bridge environments. *International Journal of Computer Aided Engineering and Technology*, 1(2), 185–208 (2009)
  24. Bonilla, I., Mendoza, M., Gonzalez-Galvn, E. J., Chavez-Olivares, C., Loredo-Flores, A., Reyes, F.: Path-tracking maneuvers with industrial robot manipulators using uncalibrated vision and impedance control. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), 1716–1729 (2012)
  25. LaValle, S. M.: Planning algorithms. Cambridge university press (2006)
  26. Liu, Z., Xu, J., Yang, C., Zhao, Y., Zhang, T.: A TE-E optimal planning technique based on screw theory for robot trajectory in workspace. *Journal of Intelligent & Robotic Systems* (2017)
  27. Li, S., He, J., Li, Y., Rafique, M. U.: Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective. *IEEE Transactions on Neural Networks and Learning Systems*, 28(2), 415–426 (2017)
  28. He, W., Ouyang, Y., Hong, J.: Vibration control of a flexible robotic manipulator in the presence of input deadzone. *IEEE Transactions on Industrial Informatics*, 13(1), 48–59 (2017)
  29. He, W., He, X., Zou, M., Li, H.: PDE model-based boundary control design for a flexible robotic manipulator with input backlash. *IEEE Transactions on Control Systems Technology*, pp. 1–8 (2018)
  30. Gao, H., He, W., Zhou, C., Sun, C.: Neural network control of a two-link flexible robotic manipulator using assumed mode method. *IEEE Transactions on Industrial Informatics*, pp. 1–1 (2018)
  31. Hassan, M., Liu, D., Paul, G., Huang, S.: An approach to base placement for effective collaboration of multiple autonomous industrial robots. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3286–3291 (2015)
  32. Hassan, M., Liu, D., Paul, G.: Collaboration of multiple autonomous industrial robots through optimal base placements. *Journal of Intelligent & Robotic Systems* (2017)
  33. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: Robotics: modelling, planning and control. Springer Science & Business Media (2010)
  34. Yuan, S., Skinner, B., Huang, S., Liu, D.: A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. *European Journal of Operational Research*, 228(1), 72–82 (2013)
  35. Carter, A. E., Ragsdale, C. T.: A new approach to solving the multiple traveling salesperson problem using genetic algorithms. *European Journal of Operational Research*, 175(1), 246–257 (2006)
  36. Deb, K., Goel, T.: Controlled elitist non-dominated sorting genetic algorithms for better convergence. In: Zitzler, E., Thiele, L., Deb, K., Coello Coello, C., Corne, D. (eds.), Evolutionary Multi-Criterion Optimization, vol. 1993 of Lecture Notes in Computer Science, pp. 67–81, Springer Berlin Heidelberg (2001)
  37. Riquelme, N., Lcken, C. V., Baran, B.: Performance metrics in multi-objective optimization. In: 2015 Latin American Computing Conference (CLEI), pp. 1–11 (2015)
  38. Ishibuchi, H., Masuda, H., Tanigaki, Y., Nojima, Y.: Difficulties in specifying reference points to calculate

- the inverted generational distance for many-objective optimization problems. In: 2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM), pp. 170–177 (2014)
39. Peters, S.: Quadtree- and octree-based approach for point data selection in 2D or 3D. *Annals of GIS*, 19(1), 37–44 (2013)

**Mahdi Hassan** received his Mechanical Engineering degree with first class honours in 2013 from the University of Technology Sydney (UTS). He undertook his Ph.D. studies at UTS Centre for Autonomous Systems (CAS) and received his Ph.D. early 2018. Since early 2017, Dr. Mahdi Hassan has been working as a research associate at CAS. His research is in robotics; and his research interests include adaptive and cooperative complete coverage in complex 3D environments; autonomous industrial robots; path, motion and task planning; multi-robot collaboration and coordination; and optimization-based algorithms.

**Dikai Liu** received his Ph.D. degree in 1997. His main research interest is robotics including exploration, motion planning, robot teams, and physical human-robot interaction. Professor Liu has been developing novel methods and algorithms that enable robots to operate in unstructured and complex 3D environments autonomously or collaboratively with human users. Example robotic systems he developed and practically deployed include autonomous grit-blasting robots for steel bridge maintenance, assistive robots for human strength augmentation in industrial applications and bio-inspired climbing robots for steel structure inspection.

**Dongliang Xu** Ph.D., is born in January 1970. He is currently working in the school of mechanical and electrical engineering, Wuhan University of Technology, China. He is mainly engaged in fiber winding and placement technology and equipment research.