Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 207 (2022) 2335–2344

Procedia Computer Science

www.elsevier.com/locate/procedia

26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

Coordination and path planning of a heterogeneous multi-robot system for sheet metal drilling

Satheeshkumar Veeramani[a,*], Sreekumar Muthuswamy[b], Rossitza Setchi[a]

[a] *Centre for AI, Robotics and Human-Machine Systems, Cardiff University, Cardiff CF24 3AA, UK*

[b] *Centre for AI, IoT and Robotics, Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram, Chennai 600127, India*

Abstract

This paper presents the details of a sub-system developed to address coordination between a serial manipulator robot (machining) and SwarmItFIX robot (fixturing) for a sheet metal drilling process. A heterogeneous multi-robot coordination methodology that has already been demonstrated to be successful in a milling process has been further enhanced here to make it suitable for a drilling process. For the convergence of joint angles in the trajectory planning of the serial manipulator robot, an optimisation-based approach is proposed. The velocity of the tool center point (TCP) is considered to be constant throughout, as it improves the quality of the machining. The SwarmItFIX robot abides by a revised five-step locomotion strategy to traverse between any two support locations. A new time plan that ensures multi-robot coordination has also been proposed in this work. The proposed method has been tested with three different drilling patterns, and the results show that the proposed method computes the trajectory of the serial manipulator, support locations of the SwarmItFIX and locomotion sequence of the base agent accurately.

Keywords:
Robot-assisted sheet-metal drilling; reinforcement learning; heterogeneous multi-robot coordination; robot path planning; SwarmItFIX

Nomenclature

${}^{w}x_0$, ${}^{w}y_0$, ${}^{w}z_0$	Coordinates of the point of origin
${}^{s}x_b$, ${}^{s}y_b$, ${}^{s}z_b$	Coordinates of the serial manipulator's base frame
${}^{s}x_w$, ${}^{s}y_w$, ${}^{s}z_w$	Coordinates of the serial manipulator's w^{th} waypoint
${}^{s}\theta_{k,w}$	k^{th} joint's angle of the serial manipulator at waypoint w
${}^{s}\theta_k$, ${}^{s}d_k$, ${}^{s}\alpha_k$, ${}^{s}a_k$	Denavit-Hartenberg (DH) parameters of the serial manipulator (k = 0 to 6)

* Corresponding author. Tel.: +91 9843623143
E-mail address: veeramanis@cardiff.ac.uk

10.1016/j.procs.2022.09.292

${}^{s}\theta_k^+$, ${}^{s}\theta_k^-$	Upper and lower limits of the k^{th} joint of the serial manipulator
η	Maximum permissible difference in joint angles of the serial manipulator between two consecutive waypoints
${}^{h}x_m$, ${}^{h}y_m$, ${}^{h}z_m$	Coordinates of the m^{th} support location
${}^{b}Gpos_m$	m^{th} goal position of the traversing base agent
${}^{p}\theta_{j,m}$	Angle of the joints of the parallel kinematic machine (pkm) of the SwarmItFIX at the m^{th} support location
${}^{p}\theta_{j,park}$	Angle of the joints of the pkm of the SwarmItFIX at the parking location
${}^{h}\theta_m$	Orientation of the head agent for the m^{th} support location
${}^{safe}d$	Safe distance between the drill position and support location
${}^{ds}x$,${}^{ds}y$	Distance between the drill position and support location on the x- and y-axis
l_{arm}	Length of the serial arm of the SwarmItFIX
$Q({}^{b}s, {}^{b}a)$	Q-value for the state action pair(${}^{b}s, {}^{b}a$) of the base agent
$Q({}^{b}s', {}^{b}a')$	Q-value of the base agent's target state-action pair
${}^{b}\alpha$	Step-size for the base agent
${}^{b}r_t$	Reward value obtained by the computational agent at time step $t + 1$
γ	Discount factor
${}^{b}\pi$, ${}^{b}\pi^*$	Non-optimal and optimal policies of head and base agents
${}^{beg}t_m$, ${}^{end}t_m$	Beginning and end time of drilling at the m^{th} location
${}^{drill}x_m$, ${}^{drill}y_m$	Coordinates of the m^{th} drill location
${}^{prev}x$, ${}^{prev}y$	Coordinate of previous location of the TCP of the serial manipulator robot
${}^{s}x_0$, ${}^{s}y_0$	Home position of the TCP
${}^{tool}v$	Tool feed velocity

1. Introduction and related works

1.1. Robots in manufacturing

Nowadays, mass customisation is considered in manufacturing industries due to an increase in global competition. Products have been increasingly created based on the individual preferences of customers. However customer preferences vary due to each customer's origin, location and culture. This situation forces industries to improve flexibilities in their production layouts. In robot-assisted flexible manufacturing systems (FMS), robots play a major role in various operations, such as fixturing, drilling, inspection, painting and assembly [1, 2]. The main advantage in deploying robots is that the same robot can perform all these tasks just by changing the end-of-arm tooling and the computer control's programming.

Fixtures are mechanical devices that hold a workpiece in a proper orientation at the correct position and guide machining tools in performing operations. Generally, in the total production budget, fixture design alone can contribute up to 20%. A conventional manufacturing system deploys two or more set of fixtures for the production of different products [3], which is expensive and time-consuming as well. To overcome this, industries started utilising robot-assisted fixturing systems called robot fixtureless assemblies (RFAs). RFAs are equipped with the correct fixtures as end-effectors for the corresponding products [4]. Usage of robots in the fixturing process reduces the fixture reconfiguration time, which ultimately improves the production time. Also, costs, such as for hardware of multiple fixtures, labour and reconfiguration in the case of modular fixtures, are reduced to the maximum possible extent. Another advantage is that RFAs do not require any human intervention. With the integration of various sensors in the robot end effectors for sensing various aspects of the workpiece through programmable logic controllers (PLCs) or other types, an RFA system attains more autonomy. Additionally, these sensor-integrated RFAs will also act as "in-line inspection devices", which eliminate a dedicated inspection period. These kinds of efficient systems ultimately increase throughput [5]. The SwarmItFIX (Fig.1) is one such patented RFA setup, which includes mobile robots integrated with swarm techniques, shape adaptability and constrained locomotion. Five representative thin sheet parts used in the production of aircraft model P180 of Piaggio Aero Industries, Italy, were selected as prototypical applications [6]. The REMORA (reconfigurable mobile robot for manufacturing applications) is a quadruped robot that traverses a static platform. Both of these robots are designed exclusively for fixturing workpieces in manufacturing operations, mainly sheet metal workpieces in the automotive and aerospace industries [7]. However, when compared with the SwarmItFIX, the REMORA [8] is designed mainly to support large solid workpieces from a platform possessing a high payload capacity.

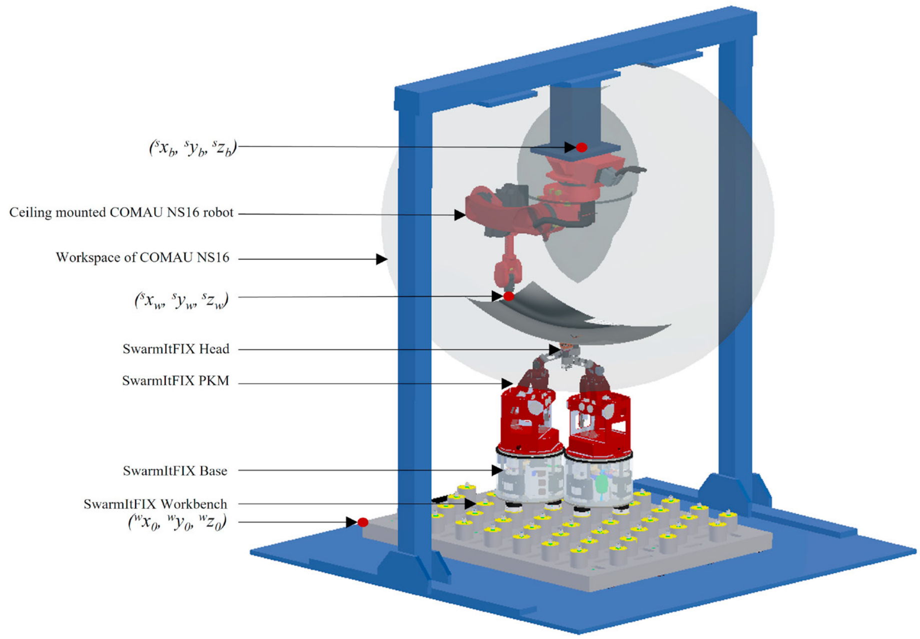


Fig. 1: CAD model of the heterogeneous multi-robot setup for sheet metal drilling

1.2. SwarmItFIX—a mobile robot with discrete and constrained locomotion

In general, it is difficult to make the agents of the multi-robot systems traverse without colliding with the environment, and also with themselves. Thus, an efficient multi-agent path planning model that computes a collision-free sequence of operations performed by the agents is required. Veeramani and Muthuswamy [9] presented a detailed methodology for collision-free path planning of the SwarmItFIX robot. The same authors have also explained other technical details required for planning, which include the work-piece geometry, constraints in the head positioning, various distance parameters for planning the head positions, base position planning and displacement planning. Once the position and orientation of the head and the position of the base are planned between the two adjacent support locations of the SwarmItFIX, the sequence of the locomotion between the two positions can be performed in five steps. For the path planning of base agents, Sagar et al [10] have used various heuristic algorithms and compared each for material handling applications. Finally, the nearest neighbour heuristic algorithm was found to provide better results. The same authors [11] also solved the multi-agent multi-goal problem of the base agents using two formulations (edge-based and vertex-based) of integer linear programming, and the results demonstrated that the edge-based formulation was efficient in terms of makespan. The authors claim that the nearest neighbour with a random insert heuristic approach provides good solution sequences for the SaD (Swing and Dock) locomotion of the mobile base of the SwarmItFIX. Whereas, Zielinska et al [12] used a CSP (constraint satisfaction problem) approach for solving the path planning of a base agent. Though the CSP approach provides decent results, it just computes the goal positions alone for the SwarmItFIX base and not the locomotion sequence/path. It has to plan again for the same conditions a second time, and the computation takes place for the same plan once again. This replanning procedure causes unnecessary detours. Finally, the above methodologies were implemented in machining various aircraft elements, such as an aircraft fuselage, wing and tail fin, using the MRROC++ robot programming framework [13].

1.3. Kinematics of serial manipulators

In general, the kinematics of serial manipulators are classified as forward or direct and inverse or reverse. In forward kinematics, the coordinates of the target position and orientation of the tool are calculated based on known joint variables. With known target coordinates and the orientation of the tool, the joint variables of a serial manipulator can be computed using an inverse kinematics model. Numerous methods have already been explored for addressing the inverse kinematics of serial robot mechanisms. The inverse kinematics can be solved with the help of homogenous transformations. Solvers that read the multidimensional geometric information of a mechanism efficiently address inverse kinematics problems [14]. With the availability of a proper inverse kinematics model, the path planning problem of industrial manipulators can be efficiently solved by a reinforcement learning (RL) framework [15]. [16] shows an efficient path planning model that integrates deep RL and inverse kinematics models for a serial manipulator robot. In inverse kinematics problems of serial manipulators, solvers can produce more than one solution. An optimal one among these solutions can be identified by imposing constraints on the joint angles. The same can be modified further to investigate the joint stiffness and path deviation due to interaction forces [17].

1.4. Markov decision process and reinforcement learning

The Markov decision process (MDP) is an artificial intelligence (AI) technique that addresses sequential decision problems in fully observable stochastic environments with transition and reward functions [18]. An MDP can be defined by a set of variables or tuple $< S, A, T, R, \gamma >$. The outcome of the MDP that guides the actual robot while choosing an action in a real environment is called a policy (π). An MDP problem may have many such policies. However, amongst all the policies, the one that yields the maximum utility value (U) is called the optimal policy, $\pi^*(s)$. These optimal policies can be computed using various methods, including linear programming, dynamic programming and model-free reinforcement learning [19].

RL is an effective technique in the path planning domain, when the environment is known to the robot [20]. The problems modelled as MDPs utilise model-based or model-free RL algorithms for the computation of $\pi^*(s)$. In general, both techniques can be offline or static, where the computed (π^*) will be referred to for performing optimal actions in real time. In both techniques, the computational agents explore the computational environment by performing actions suggested by the current policy, collect the reward and update the utility table. Computational agents initially perform random actions, learn by exploring the environment and finally exploit the environment with the learned information [21]. The skills acquired by the computational agents can be evaluated using the intrinsic motivations in the RL framework. The optimal policy obtained by the simulation can be deployed in robots successfully for their optimal locomotion. By combining asynchronous methodologies with tabular RL algorithms, optimal policies can be obtained much faster [22].

The multi-robot coordination capability could increase the machining workspace and thereby reduce initial investment cost. The literature survey confirms that the coordination among industrial the serial manipulator (machining tool) and the SwarmItFIX robots (fixture) for drilling operation has not been explored yet. Hence this work proposes a planning model to address the identified gap.

2. Proposed multi-robot coordination model

This section presents the proposed method for achieving the coordination between the serial manipulator and SwarmItFIX robots. The serial manipulator performs the drilling on top of the sheet metal and the SwarmItFIX acts as the reconfigurable fixture and supports the sheet metal from the bottom.

2.1. Trajectory of the ceiling-mounted serial manipulator

The six-axis serial manipulator robot (COMAU NS16) is deployed to perform the sheet drilling operation with a drilling bit as its end-of-arm tool. The front-right corner of the workbench of the SwarmItFIX is considered the reference coordinate (${}^w x_0, {}^w y_0, {}^w z_0$). Figure 2 shows the frame assignment of the inverted COMAU NS16 robot, and the DH parameters are listed in Table 1. The objective function for computing the angles of all joints (${}^s\theta_{k,w}$) for the manipulator to reach the w^{th} drill point (${}^s x_w, {}^s y_w, {}^s z_w$) is given in Eq. 1. Eqs. 2, 3 and 4 are the constraints for the joint angle computation.

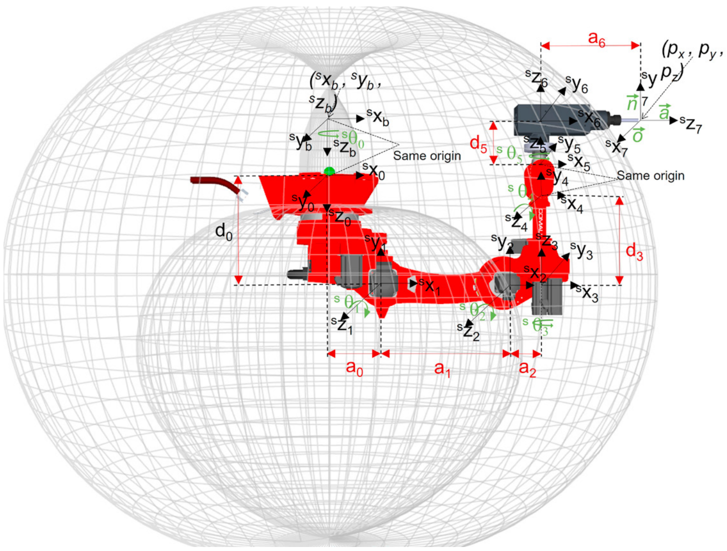


Fig. 2: Frame assignment for the COMAU NS16 (home position)

Table 1: DH parameters of the COMAU NS16

Link (k)	${}^s\theta_k$ (deg.)	${}^s d_k$ (mm)	${}^s a_k$ (mm)	${}^s\alpha_k$ (mm)
0	θ_0	-600	300	-90
1	$\theta_1 - 90$	0	-700	0
2	θ_2	0	-185	-90
3	θ_3	624	0	+90
4	θ_4	0	0	-90
5	θ_5	255	0	0
6 (tool)	-	0	-120	0

d_0 = -600 mm, a_0 = 300 mm, a_1 = -700 mm, a_2 = -185 mm, d_3 = 624 mm, d_5 = 255 mm, a_6 = -120 mm

Objective function:

$$f({}^s x_w, {}^s y_w, {}^s z_w) = Min\left[f({}^s\theta_{k,w}) - \begin{pmatrix} {}^s x_w \\ {}^s y_w \\ {}^s z_w \end{pmatrix} \right] \tag{1}$$

Constraints:

$${}^s\theta_k^- \leq {}^s\theta_{k,w} \leq {}^s\theta_k^+; where, k = 0, 2, 3, 4, 5 \tag{2}$$

$$0 \leq {}^s\theta_{1,w} \leq {}^s\theta_k^+ \tag{3}$$

$${}^s\theta_{k,w-1} \leq {}^s\theta_{k,w} \leq \eta \tag{4}$$

2.2. Motion planning of SwarmItFIX robot

This section presents the mathematical model, which is used for the support location identification of the SwarmItFIX head, goal position computation and path planning of the SwarmItFIX base agent. Figure 3 shows the proposed method for the motion planning of the SwarmItFIX robot.

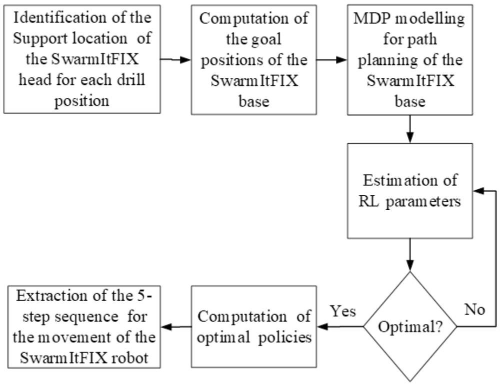


Fig. 3: Method for the motion planning of the SwarmItFIX robot

2.2.1. The revised five-step locomotion

The proposed five-step locomotion strategy given below overcomes the existing replanning and detour issues by computing the locomotion sequence of a base agent instead of just the goal position. Let $({}^{h}x_{m-1}, {}^{h}y_{m-1}, {}^{h}z_{m-1})$ be the coordinate of the $(m-1^{th})$ support location, and ${}^{b}Gpos_{m-1}$ is the corresponding goal position of the base agent. The transition of the head agent between the $(m-1)^{th}$ and m^{th} goal position on the workbench is due to the locomotion base agent. In contrast, the positioning and orientation of the vacuum cup on the support locations are taken care of by the PKM and head agents, respectively. Starting from the $(m-1)^{th}$ supporting position $({}^{h}x_{m-1}, {}^{h}y_{m-1}, {}^{h}z_{m-1}, {}^{b}Gpos_{m-1})$, the SwarmItFIX robot attains the next supporting position $({}^{h}x_{m}, {}^{h}y_{m}, {}^{h}z_{m}, {}^{b}Gpos_{m})$ by performing the revised five-step locomotion strategy given in the following:

i. Release of head agent from $({}^{h}x_{m-1}, {}^{h}y_{m-1}, {}^{h}z_{m-1})$.

ii. PKM agent traverses to keep the vacuum cup at a parking position $({}^{p}\theta_{j,park})$.

iii. Calculate the base agent Locomotion/path $({}^{b}Gpos_{m-1} \rightarrow {}^{b}Gpos_{m})$ and the respective counter-rotation of the PKM.

iv. The PKM agent traverses to keep the vacuum cup at a normal approach away $({}^{p}\theta_{j,m})$ from the m^{th} support location.

v. Head attains $({}^{h}x_{m}, {}^{h}y_{m}, {}^{h}z_{m}, {}^{h}\theta_{m})$.

2.2.2. Identification of the support locations of the SwarmItFIX head

The coordinates of the m^{th} support location of the head agent can be computed with the help of geometrical constraints given as under

$$\{{}^{h}x_{m}, {}^{h}y_{m}\} = \{{}^{ds}x + [{}^{safe}d \times \cos(\phi - 90^{o})], {}^{ds}y + [{}^{safe}d \times \sin(\phi - 90^{o})]\} \tag{5}$$

2.2.3. Computation of the goal positions of the SwarmItFIX base

This step computes the ${}^{b}Gpos_{m}$ of the base agent in the workbench for all the respective support locations. The coordinate $({}^{h}x_{m}, {}^{h}y_{m}, {}^{h}z_{m})$ is utilised for the computation of the ${}^{b}Gpos_{m}$. In this regard, initially, a polygon is created with projections of $({}^{h}x_{m}, {}^{h}y_{m}, {}^{h}z_{m})$ on the workbench. The four outer corners of the polygon e_1, e_2, e_3 and e_4 are calculated using Eq.6. Finally, the positions that satisfy the constraints Eq. 7 and 8 are considered as the ${}^{b}Gpos_{m}$ of the m^{th} support location. When more than one solution is identified, the base position that is close to $({}^{h}x_{m}, {}^{h}y_{m})$ becomes the ${}^{b}Gpos_{m}$:

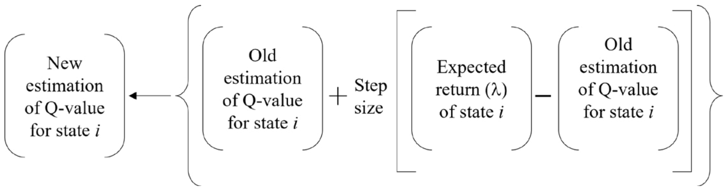
$$e_1 = \begin{Bmatrix} {}^{h}x_m - [l_{arm}], \\ {}^{h}y_m - [l_{arm}] \end{Bmatrix}; e_2 = \begin{Bmatrix} {}^{h}x_m + [l_{arm}], \\ {}^{h}y_m - [l_{arm}] \end{Bmatrix}; e_3 = \begin{Bmatrix} {}^{h}x_m + [l_{arm}], \\ {}^{h}y_m + [l_{arm}] \end{Bmatrix}; e_4 = \begin{Bmatrix} {}^{h}x_m - [l_{arm}], \\ {}^{h}y_m + [l_{arm}] \end{Bmatrix} \tag{6}$$

$$area(e_1, e_2, e_3, e_4) \geq sum(T_1, T_2, T_3, T_4) \tag{7}$$

$$dist(e_1, C_{cd}) \geq l_{arm} \;\&\&\; dist(e_2, C_{cd}) \geq l_{arm} \tag{8}$$

2.2.4. Reinforcement learning-based path planning of the SwarmItFIX base

The problem of identification of the optimal path for the base agent to traverse between (${}^{b}Gpos_{m-1}$ and ${}^{b}Gpos_{m}$) is modelled as a Markov Decision Problem. An efficient RL algorithm that analyses the robot environment and returns the optimal path in the form of a sequence of actions with less computational cost has to be deployed for solving the MDP. Thus, a model-free state-action-reward-state-action (SARSA) on-policy temporal difference (TD) control RL algorithm is chosen in this work, and the RL agent parameters for the SwarmItFIX base path planning problem have been described [21]. The first step in the SARSA algorithm is choosing a random state-action pair (${}^{b}s_i, {}^{b}a_i$) with the state ${}^{b}s_i$ as the initial state of the episode. The Q-value will then be calculated and updated in the state-action matrix $Q({}^{b}s, {}^{b}a)$ against the corresponding initial state-action pair (${}^{b}s_i, {}^{b}a_i$). The general form of the SARSA-TD control algorithm update rule can be written as



The expected return (λ) of the episode is calculated as under

$$\lambda = E_{\pi}\left[\sum_{j=0}^{\infty} \gamma^{j} r_{t+j+1}\right], \tag{9}$$

which can be further simplified as

$$\lambda = E_{\pi}\left[r_{t+1} + \gamma Q(s_{i+1}, a_{i+1})\right] \tag{10}$$

The SARSA algorithm updates the $Q({}^{b}s_i, {}^{b}a_i)$ of the base agent using Eq. 11. Finally, the policy matrix (${}^{b}\pi$) also is updated based on the calculated $Q({}^{b}s, {}^{b}a)$:

$$Q({}^{b}s_i, {}^{b}a_i) \leftarrow Q({}^{b}s_i, {}^{b}a_i) + {}^{b}\alpha\left[{}^{b}r_{t+1} + \gamma Q({}^{b}s', {}^{b}a') - Q({}^{b}s_i, {}^{b}a_i)\right] \tag{11}$$

2.2.5. Time plan

This step explains the time stamp for all the support locations of the SwarmItFIX head. The support engagement time is computed using Eq.12. As already mentioned, velocity of the TCP (${}^{tool}v$) of the serial manipulator is assumed to be constant for better machining quality. Hence, the duration of the support is considered as constant for all the drill locations:

$${}^{eng}t_m = {}^{rel}t_{(m-1)} + \frac{\sqrt{\left({}^{drill}x_m - {}^{prev}x\right)^2 + \left({}^{drill}y_m - {}^{prev}y\right)^2}}{{}^{tool}v}, \tag{12}$$

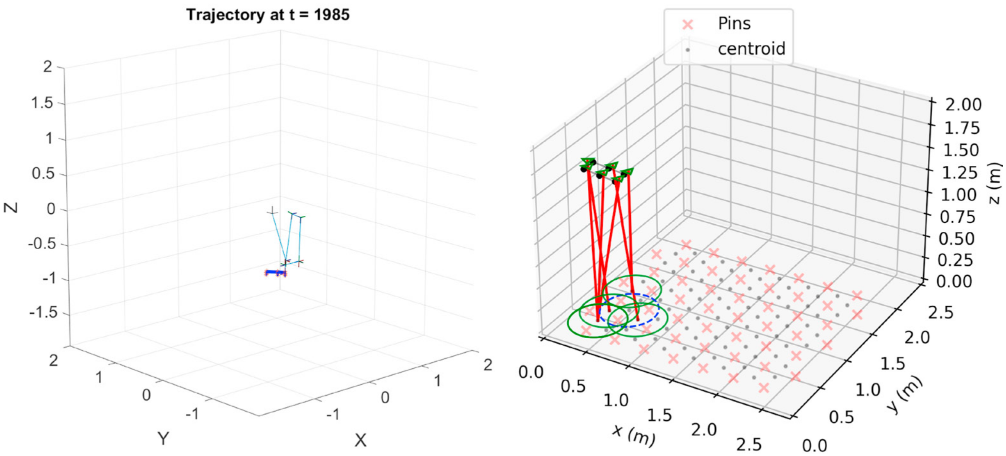
where

$${}^{prev}x,\ {}^{prev}y = \left\{\begin{matrix} {}^{s}x_0,\ {}^{s}y_0, if\ m = 0 \\ {}^{drill}x_{m-1},\ {}^{drill}y_{m-1}, if\ m \neq 0 \end{matrix}\right\} \tag{13}$$

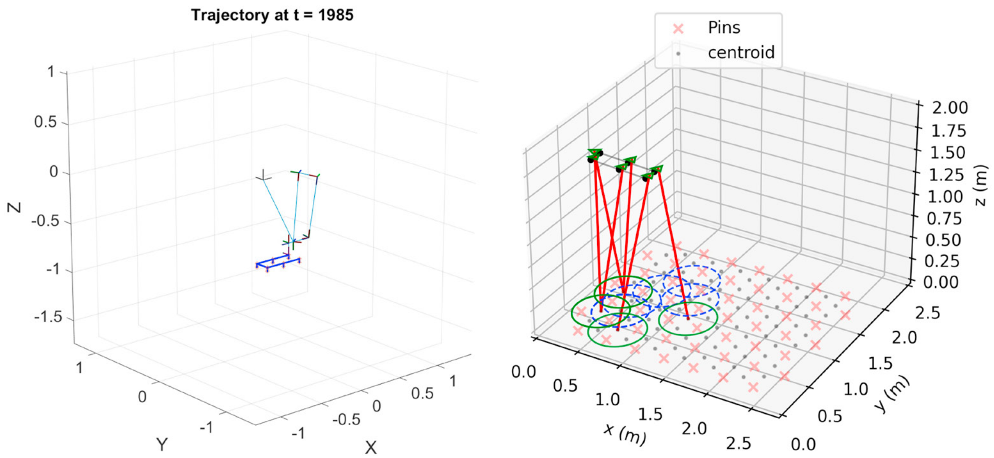
3. Evaluation of the proposed model

This section presents the outcomes of the execution of the proposed model in three different scenarios. The PC used for the computation was an Intel Intel(R) Core(TM) i5-8265U CPU, 1.60 to 1.80 GHz with 16.0 GB RAM, running Windows 10 (64-bit OS). The numerical computation of the COMAU NS16 trajectory planning was performed in the MATLAB Robotics System Toolbox. The reinforcement learning computations of the SwarmItFIX were performed in the Python 3.7 Spyder interface.

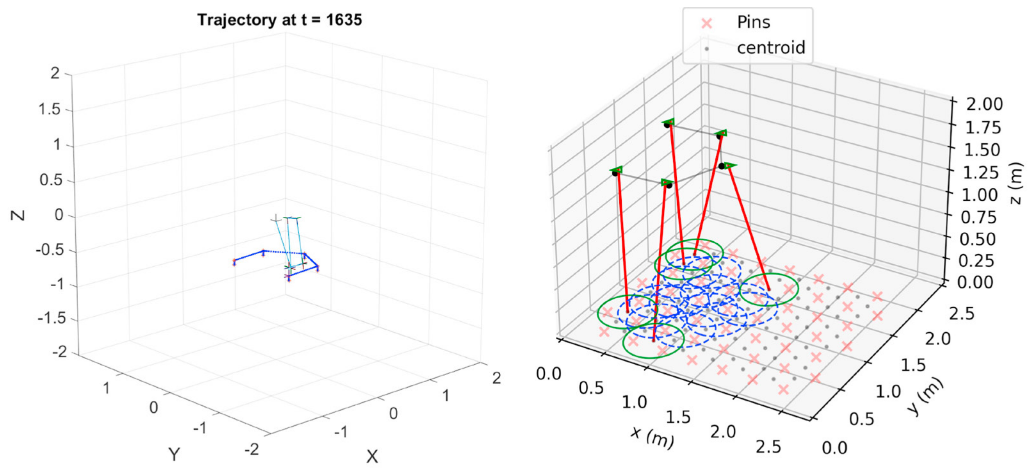
The home position of the COMAU NS16 robot ${}^{s}\theta_{k,home}$ is (180^0, 0^0, -90^0, 0^0, 0^0, 0^0, 0^0) and ${}^{tool}\nu$ is assumed to be 5 mm/s. The reference origin (${}^{w}x_0, {}^{w}y_0, {}^{w}z_0$) is the front-right corner of the workbench, and the base coordinate of the serial manipulator (${}^{s}x_b, {}^{s}y_b, {}^{s}z_b$) is computed from the reference. The proposed model has been tested with three drilling patterns to ensure the ability of the proposed model, and the results are furnished in Fig.4. The three drill patterns were (1) closely arranged with even distances between the points, (2) closely arranged with uneven distance between the points (3) and sparsely arranged drill points. The proposed method successfully computed the trajectory and path of the COMAU NS16 and the SwarmItFIX, respectively, in all three cases. The trajectory of the TCP and joint angles of the COMAU NS16 for the sparsely arranged pattern are presented in Fig.5. It can be observed that joints 1 and 4 together performed the movements of the TCP in the z direction while the drilling took place, which ultimately shares the load and hence avoids jerk and facilitate smoother machining.



(a) Closely arranged drill points

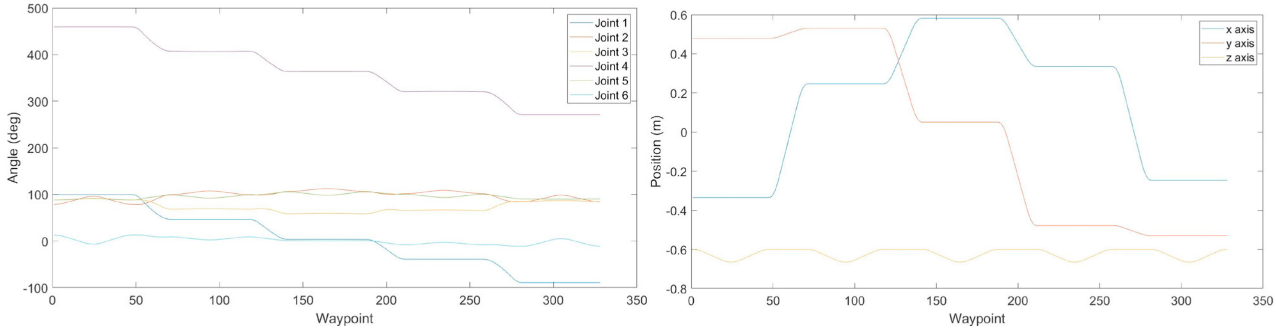


(b) Closely arranged drill points with uneven distances

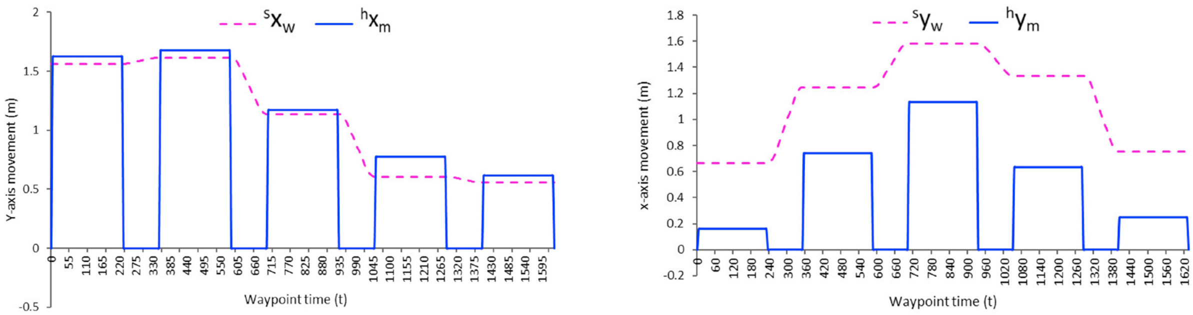


(c) Sparsely arranged drill points

Fig. 4: Trajectory planning of the serial manipulator and the placement of multiple agents of the SwarmItFIX robot for the three different drill patterns



(a) Joint angles and position



(b) Coordination between the TCP and SwarmItFIX head

Fig. 5: Trajectory of the serial manipulator and SwarmItFIX head for the sparsely arranged drill pattern

It can also be seen from Fig.5b that movements of the TCP and the SwarmItFIX head were highly coordinated. Where the SwarmItFIX head attains $(^{h}x_m, ^{h}y_m)$ when the $^{s}x_w$ and $^{s}y_w$ are idle (i.e., the serial manipulator performs drilling in z direction).

4. Conclusions

This study proposes a novel method for heterogeneous multi-robot coordination during sheet metal drilling. To achieve this, an inverse kinematics model of a serial manipulator is coupled with a task planning framework that computes the support locations of the SwarmItFIX. The result demonstrated that the proposed model ensures the successful coordination among the robots for the sheet metal drilling operation in all the tested cases. The proposed method uses geometrical constraints for computation of support locations of the head agent, and RL for locomotion planning of the base agent. For the sparsely arranged drill pattern the agent takes 391.95±6s for learning the sub-optimal policy. Interestingly, for the closely arranged pattern with even and uneven distances, the learning time are 515.8±10s and 504.72±5s respectively. The scope of proposed planners is limited to the path planning of robots with constrained locomotion only. Investigation of machining efficiency and quality, and stiffness of all joints of the COMAU NS16 manipulator in the inverted position is meant for future work.

Acknowledgements

The support received from Prof. Matteo Zoppi, Prof. Rezia Molfno and Dr. Keerthi Sagar of PMAR Research Group @ the University of Genoa, Italy is greatly appreciated. The authors would also like to acknowledge the support extended by the Centre for Artificial Intelligence, Robotics and Human-Machine Systems (IROHMS), operation C82092, partially funded by the European Regional Development Fund (ERDF) through the Welsh Government.

References

[1] Zoppi M., Molfino R., and Zlatanov D. (2013) "Bench and method for the support and manufacturing of parts with complex geometry." *US Patent* **US8495811B2**.

[2] Olabi A., Béarée R., Gibaru O., and Damak M. (2010) "Feedrate planning for machining with industrial six-axis robots." *Control Engineering Practice* **18** (5): 471–482.

[3] Tohidi, H. and AlGeddawy T. (2016) "Planning of modular fixtures in a robotic assembly system." *Procedia CIRP* **41**: 252–257.

[4] de Leonardo, L., Zlatanov D., Zoppi M., and Molfino R. (2013) "Design of the locomotion and docking system of the swarmitfix mobile fixture agent." *Procedia Engineering* **64**: 1416–1425.

[5] Foumani M., Smith-Miles K., and Gunawan I. (2017) "Scheduling of two-machine robotic rework cells: In-process, post-process and in-line inspection scenarios." *Robotics and Autonomous Systems* **91**: 210–225.

[6] Molfino R., Zoppi M., and Zlatanov D. (2009) "Reconfigurable swarm fixtures." *2009 ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots*, London, UK, 730-735.

[7] Veeramani S., Muthuswamy S., Sagar K., and Zoppi M. (2020) "Artificial intelligence planners for multi-head path planning of SwarmItFIX agents." *Journal of Intelligent Manufacturing* **31**(4): 815–832, 2020..

[8] Yang, H., Krut S., Baradat C., and Pierrot F. (2011) "Locomotion approach of remora: A reconfigurable mobile robot for manufacturing applications." *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, CA, USA, 5067-5072.

[9] Veeramani S., and Muthuswamy S., (2021) "Hybrid type multi-robot path planning of a serial manipulator and SwarmItFIX robots in sheet metal milling process." *Complex & Intelligent Systems*, 1–18.

[10] Sagar K., Zlatanov D., Zoppi M., Nattero C., and Muthuswamy S. (2016) "Path planning of a material handling agent with novel locomotion." *Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, **50145**.

[11] Sagar K., Zlatanov D., Zoppi M., Nattero C., and Muthuswamy S. (2016) "Multi-goal path planning for robotic agents with discrete-step locomotion." *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, **58172**.

[12] Zielinska T., Kasprzak W., Szynkiewicz W., and Zieliński C. (2014) "Path planning for robotized mobile supports." *Mechanism and Machine Theory* **78**: 51–64.

[13] Zielinski, C. and Winiarski T. (2010) "Motion generation in the MRROC++ robot programming framework." *The International Journal of Robotics Research* **29**(4): 386–413.

[14] Husty, M. L., Pfurner M., and Schröcker H.-P. (2007) "A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator." *Mechanism and Machine Theory* **42**(1): 66–81.

[15] Roveda, L., Maskani J., Franceschi P., Abdi A., Braghin F., Tosatti L. M., and Pedrocchi N. (2020) "Model-based reinforcement learning variable impedance control for human-robot collaboration." *Journal of Intelligent & Robotic Systems* **100**(2): 417–433.

[16] Zhong, J., Wang T., and Cheng L. (2021) "Collision-free path planning for welding manipulator via hybrid algorithm of deep reinforcement learning and inverse kinematics." *Complex & Intelligent Systems* 1–14.

[17] Cordes, M., and Hintze W. (2017) "Offline simulation of path deviation due to joint compliance and hysteresis for robot machining." *International Journal of Advanced Manufacturing Technology* **90**(1): 1075–1083.

[18] Veeramani S., Muthuswamy S., Sagar K., and Zoppi M. (2019) "Multi-head path planning of SwarmItFIX agents: A Markov decision process approach." *Proceedings of the IFToMM WC on Mechanisms and Machine Science* **73**, Krakow, Poland, 2237-2247.

[19] Wang Y. and de Silva C. W. (2009) "Sequential Q-learning with kalman filtering for multirobot cooperative transportation." *IEEE/ASME Transactions on Mechatronics* **15**(2): 261–268.

[20] Sutton R. S., and Barto A. G. (2018) Reinforcement learning: An Introduction *(2nd ed.)*, MIT press.

[21] Veeramani S., and Muthuswamy S. (2022) "Reinforcement learning based path planning of multiple agents of SwarmItFIX robot for fixturing operation in sheetmetal milling process." *Proc IMechE, Part B: Journal of Engineering Manufacture*.

[22] Zhao X., Ding S., An Y., and Jia W. (2018) "Asynchronous reinforcement learning algorithms for solving discrete space path planning problems." *Applied Intelligence* **48**(12): 4889–4904.