



HAL
open science

Contribution of the omnidirectional autonomous mobile robot to manufacturing systems agility

Jeannette Flayfel, Guillaume Demesure, Bril El-Haouzi

► **To cite this version:**

Jeannette Flayfel, Guillaume Demesure, Bril El-Haouzi. Contribution of the omnidirectional autonomous mobile robot to manufacturing systems agility. 11th Workshop on Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future, SOHOMA 2021, Nov 2021, Cluny, France. hal-03577533

HAL Id: hal-03577533

<https://hal.science/hal-03577533>

Submitted on 16 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Contribution of the Omnidirectional Autonomous Mobile Robot to Manufacturing Systems Agility

Jeannette Flayfel, Guillaume Demesure, Hind Bril El-Haouzi

Abstract: Nowadays, achieving a certain level of agility in a manufacturing system represents a step forward in the direction of Industry 4.0. As material handling is a very important aspect of production systems, the use of Autonomous Mobile Robots (AMR) has started to gain increasing popularity in the manufacturing domain. This paper focuses on two main problems. The first one is the study related to the agility requirements for a continuous changing demand in a shopfloor with focus on the material handling solutions, considered as the best for agility: the Autonomous Mobile Robots. The second problem addressed in this paper is the actual implementation, on a robot prototype with the help of a Particle Swarm Optimization algorithm for the robot path planning and sliding mode control for path tracking.

Keywords: Manufacturing agility, Autonomous mobile robots, Automated Guided Vehicles.

1 Introduction

In a continuously changing world, the idea of constant development is necessary to keep the inertia of persisting, ongoing growth. As in many other domains, the manufacturing domain needs to keep up with versatile demands in order to satisfy its customers' needs. Such needs involve mainly mixed product types and variations in demand. In the past, the availability of products consisted of providing products with no variety in size or packaging. Today, however, companies are promoting a higher product variation in size, type, or package, which require constant changes in manufacturing lines [1]. This first problem requires agility as a "must have" element in manufacturing systems so that the product demand is reached. In general, agility refers to "the successful exploration of competitive bases (speed, flexibility, innovation-proactivity, quality and profitability) through the integration of reconfigurable resources and best practices in a knowledge-rich environment to provide customer-driven products and services in a fast changing market environment" [2].

The second problem which arises along with the need for agility involves implementing the appropriate technology so that a manufacturing line could handle the client's demand. One of the most important competitive bases to improve the agility is the flexibility. The flexibility provides companies with possibilities of adaptation to meet customers order' demands. In that sense, material handling systems are important resource that enhance the flexibility to enable the agility. In order to reach a certain level of agility, the material handling means need to be in accordance with a reconfigurable manufacturing system (RMS) which allows the production to be flexible and responsive enough to meet customers' fluctuating demand as well as the integration of a smaller product family.

The six core features of the RMSs are Modularity, Integrability, Diagnosability, Convertibility, Customization and Scalability. The Convertibility and Customization features of an RMS refer to the ability of a system to be adaptable to new system changes to manufacture a different product demand, respectively the flexibility of the system to produce a part, family, as well as specific parts, leading to a more customized range of products [3]. Hence, the integration of robots come as a solution for the material handling problem which plays a big role in obtaining the convertibility and customization features of an RMS. In particular, the enhancement of new generation of robots with connectivity and smart capability have paved the way for CPS and IoT paradigms that are the main pillars of Industry 4.0.

In this light, the paper focuses on the treatment of the material handling problem which contributes to solving the agility need in manufacturing systems. In [1], it is stated that upgrading the material handling solution could have a positive effect on the manufacturing system's level of flexibility, which is the main issue that this paper will address. The discussed material handling upgrade is the use of Autonomous Mobile Robots (AMR) instead of Automated Guided Vehicles (AGVs) or conveyors. The aim of this study is to demonstrate the contribution of AMRs, especially the omnidirectional wheeled mobile robots, to the system agility and the present an effective implementation to TRACILOGIS test-bed platform.

The rest of the paper is organized as follows. Section 2 gives a better understanding about the study of the related works. Section 3 shows the proposed methodology and Section 4 presents the findings obtained after the execution of the case study along with an evaluation of these results and finally section 5 draws the overall conclusions.

2 State of the art

2.1 Material handling in Industry 4.0

The material handling problem generally refers to "the movement, storage, protection and control of materials and products throughout the complete lifecycle of manufacturing, warehousing, distribution and disposal" [4]. Based on this definition, the paper concentrates primarily on the movement of the products, semi-products or any materials inside the manu-

facturing stage of a product lifecycle. The main solutions found in literature that responds to the movement or transportation issues of materials, semi-products, or products refers to conveyors, automated guided vehicles (AGV) and autonomous mobile robots (AMR). The most classic material transportation system among those presented above is conveyors. Mainly, a conveyor is a system of transportation material from one place to another, according to a design layout that has been primarily installed. The main downside of this kind of material transportation solution is lack of flexibility. For this reason, another means of material transport has been introduced, more dedicated to manufacturing systems, namely AGVs. The first introduction of automated guided vehicle dates back to 1955. It represented a step forward in the world of industry, where it was mainly used for material handling. The evolution of the guiding system has gone through different stages including mechanical guidance, optical guidance, inductive guidance, inertial guidance, laser guidance, and finally vision-based guidance which is heavily used today [5].

As a better solution to the material handling problem response, the so-called “upgraded” AGVs are considered. These types of robots are most popularly known under the name of autonomous mobile robots (AMRs) [6]. As by definition, “Autonomous mobile robots are industrial robots that use a decentralized decision-making process for collision-free navigation to provide a platform for material handling, collaborative activities, and full services within a bounded area” [5]. These AMRs are considered to be an “evolution” from the already existent AGVs which bring within implemented technology that gives the robot the capability to deliver better results based on a decentralized decision-making process in comparison with the old AGVs. However, the new capabilities requires better solutions for AMR navigation that also depends on the locomotion. Thus, the first issue regarding the implementation of the AMR is the decision over the appropriate locomotion mean for the robot. When it comes to locomotion, mobile robots could be classified into two categories, such as legged mobile robots and wheeled mobile robots. The legged mobile robots locomotion is inspired from nature, imitating the locomotion system of mammals or reptiles. However, the disadvantages in mechanical complexity and inefficiency for manufacturing systems do not make it a suitable locomotion mean in our case [7]. The wheeled mobile robots on the other hand are more popular and they could be equipped with different wheel types like Conventional wheels, Caster (passive) wheels or omnidirectional wheels. By comparing these three types of wheels [8], despite their manufacturing complexity and sensitivity to rough surfaces, the omnidirectional wheels, such as the Mecanum ones, could be considered as better solution due to their degree of freedom (DoF) when full equipped on AMRs. Indeed, a robot with 2 conventional wheels and one caster wheel (equivalent to a robot with tracks) can navigate longitudinally and can turn (with $DoF = 2$), while the conventional wheels provide AMR with lateral navigation ($DoF = 3$).

The other main issue in AMR implementation refers to navigation tools, especially both trajectory/path planning and tracking control. The path planning problem could be split into two sub-problems, which are known as Global path planning and Local path planning.

The global path planning refers to the calculation of the path that the robot needs to do with the help of a map of the environment which integrates the already existing obstacles. Having this information, the robot is supposed to calculate the optimal path to reach its destination. The local path planning, on the other hand, refers to the recalculation of the route according to the encountering of unexpected obstacles, while keeping the same destination target and an optimal pathway solution. The map is constantly updated with the help of sensors installed on the robot. The recalculated map is supposed to lead the robot in such way that it avoids the newly encountered obstacles and it resembles as much as possible to the pre-calculated path by the global path planner [9]. For the actual path planning, different methods have been proposed such as probabilistic roadmaps, cell decomposition, Voronoi partition, and optimization techniques [10][11]. When manufacturing performances in terms of completion time are required, optimization-based motion planners seem to be appropriate tools [12]. However, the complexity of motion planning problem (NP-Hard) leads to the use of meta-heuristics-like solutions, PSO (Particle Swarm Optimization) algorithm [13], GA (Genetic Algorithm) [14], ACO (Ant Colony Optimization) algorithm [15]. In order to generate appropriate trajectories of the robots, path smoothing techniques, which transform the motion planning problem to parametric optimisation one, are required. The literature provides different solutions [16] and B-spline curves in comparison with other techniques offers interesting properties such as the local control of the trajectory generation (especially to deal with collision avoidance issue).

2.2 Benefits of AMRs over AGVs in terms of agility

Unlike the AMRs, the well-known AGVs only follow predefined paths, and once they are deviated from their paths, they cannot find them again without external intervention (from AGV central system, usually). AMRs, on the other hand, are capable of moving in any given area without experiencing any incidents. Because an AGV must follow a predefined path, the reconfiguration of the guiding path in an operational area can be very time-consuming. Aside from the benefit of a very quick pathway reconfiguration, AMRs have additional advantages to provide, such as patrolling and collaborating with operators, as well as communicating among themselves, resulting in decentralized control. To complement navigation and the identification of objects, the AMRs give better autonomous operational capabilities to avoid potentially encountered objects in their travel zones in some unpredictable scenarios such as obstacle avoidance and path recalculation [1]. The AGVs need pre-installed guide paths in order to operate along with a centralized control unit, whereas AMRs have a so-called “travel zone” and their control is decentralized. According to [5] after comparing centralized and decentralized control in terms of performances, the decentralized control has a better response in favour of system’s control except for when is performing a single-objective optimization task.

Since the aim of this paper is to demonstrate the contribution of AMRs to the system agility, a small comparison between these material handling solutions is presented here. Agility defined from the perspective of robot transportation problem represents “the ability of a robot system to succeed in an environment of continuous and unpredictable change by reacting efficiently and effectively to changing factors” [17]. Determining the overall measurement of an agile robot system could provide better insights into the real advantages, and most importantly, the real agility level that a robot system has. Agility could be measured through a couple of criteria with three time metrics as follows, the Cycle Time (Cy), which is the period that the robot needs to complete one cycle of an operation or complete one task from beginning to end, the Planning Time (Pl), which is the time spent by the robot to plan the desired task before beginning the action of doing that task and finally the Changeover Time (CO), which is the amount of time it takes a robot to “automate the configuration of the equipment settings for changing over from one product to another” [17].

To be able to calculate these parameters, a small case study for the proof of concept has been conducted in which a comparison of these KPIs was done, by considering the execution of two similar tasks, firstly as executed by a line follower AGV and secondly by an AMR. Here, to conduct our experiments, the AMR has been downgraded to be considered as an AGV. Both robots are supposed to recreate two specific paths, illustrated in Fig. 1 for two different cases. In the first case, the robot needs to go on a trajectory from Machine A to Machine B; in the second case, the robot considers that the manufacturing process has changed so it has to go now on the trajectory from Machine C to Machine D. The distance between Machine A (resp. C) and Machine B (resp. D) is $3m$ (resp. $2m$). The AGV path update is done by the AGV system, while the AMR updates its trajectory. The speed of the robot is considered to be $0.5m/s$ for both. The times measured for those KPIs are shown in Table 1. The Total Time (Tt) presented in the table is the KPI summation as $T = C_y + P_l + C_o$.

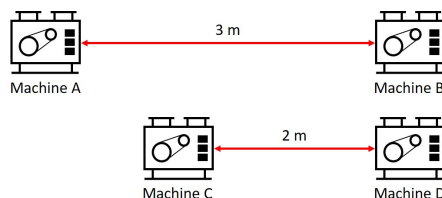


Figure 1: Machines layout for the two proposed cases

Based on the data presented in Tab. 1, the Δ column shows the difference between the times needed for the robots to complete the tasks. The results show a substantial difference using an AMR for the task execution, the advantage being highlighted in the total time needed for the robot to perform a task. Even if the planning time is greater for an AMR compared to AGV (for both Case 1 and Case 2), the big difference is noticed in the Changeover Time where a line-follower AGV would need 72 seconds for Case 1 and 48 seconds

	Case 1: machines <i>A</i> to <i>B</i>					Case 2: machines <i>C</i> to <i>D</i>				
	C_y	P_l	C_o	T	Δ	C_y	P_l	C_o	T	Δ
AGV	6	0	72	78	70.5	4	0	48	52	46.6
AMR	6	1.5	0	7.5		4	1.4	0	5.4	

Table 1: KPIs measurements (in seconds) for an AGV and an AMR

for Case 2 (since the update requires a global update, done by AGV system), but for the AMR the Changeover Time is 0 since there is no need for a pre-settled guideline for the robot functioning. Therefore, the agility performance of the AMR is significantly higher compared with that of an AGV. Moreover, in the previous section, the six core features of RMSs were presented. The Convertibility feature, which is the ability of a system to be adaptable to new changes for manufacturing a different product demand, is in line with the high agility offered by the AMR.

3 Proposal: agile AMR

As mentioned in Section 2, the advantages that the omnidirectional wheels bring within refer to their capabilities to provide AMR with high degree of freedom ($DoF = 3$) [18]. In this section, we propose to present the benefits of omnidirectional wheels to AMR agility. The objective is to demonstrate that, although using similar tools, these wheels are more suitable for AMRs to contribute to agility.

3.1 Hardware and software environment

The equipment used for the demonstration of the implemented algorithm is a Lego robot with an incorporated Lego Mindstorms EV3 brick, a Raspberry Pi microcontroller board (Fig. 2). For the robot path generation, the Python programming language has been used, more precisely a Python 3.8.3 version and a Spyder IDE. For the robot control implementation, the Simulink graphical programming environment of the MatlabR2021a has been used. The AMR prototype is equipped with four Mecanum wheels, which are omnidirectional.

3.2 Motion planning and tracking strategies

In order to compare AMR with conventional or Mecanum wheels, we used the trajectory planning and tracking tools from [12]. In this article, the PSO algorithm was chosen to implement the motion planning with following advantages: simplicity in algorithm implementation, calculation and conceptual wise, especially compared with other heuristic algorithms approaches; increased quality in the algorithm convergence with results delivered in shorter



Figure 2: The EV3 Lego mindstorms robot, equipped with Mecanum-like omnidirectional wheels.

time compared with other optimization algorithms as well as the capability to deal with continuous variables.

The searching algorithm of the PSO is based on a method of parallel searching. A swarm size is initially given and all the particles fly in the search space, having for purpose to find the global minima (or global maxima). At each t^{th} iteration, the particles move to a new position x_i^{t+1} (Equation 2) according to its velocity (Equation 1)

$$v_i^{t+1} = \omega v_i^t + c_1 r_1^t (p_{best,i}^t - x_i^t) + c_2 r_2^t (g_{best} - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

Where, x_i^t and v_i^t are respectively the position and velocity vectors of particle i at iteration t . $p_{best,i}^t$ is the personal best position of particle i achieved so far and g_{best} is the global best position among all particles. ω is the inertia weight and c_1 and c_2 are positive acceleration constants which are used to level the contribution of the cognitive and social components respectively, r_1^t and r_2^t are random numbers from uniform distribution $U \in (0, 1)$ at time t [13]. The parameters tuning of PSO is done according to Trelea guidelines [19] where $c_1 = c_2 = 1.4$. To determine the number of iterations and the swarm size, some tests have been effectuated by choosing firstly three swarm size dimensions such as 20, 40 and 80 as well as for the number of iterations 50, 100 and 150 epochs. In the first test phase, a total number of 9 parameter combinations has been considered to be tested. For each parameter combination, the python code has been executed 10 times. The performance of a certain parameter combination has been measured according to the computation time and the time needed for the robot to do the path. After testing this first trial set, it has been observed that a lower number of iterations and a bigger swarm size more appropriate for algorithm convergence and has a smaller computation time. Thus, a second set of parameters has been chosen for testing, where the the swarm size takes the values of 120, 150, and 200 and the number of iterations 20, 35 and 50. Again, a total number of 9 parameter combinations has been considered to be tested and the performance of the parameter combination has been measured the same way as before. After this final trial, the best parameter combination

seems to have the number of particles equal with 200 and the number of iterations 20. Finally, the inertia weight ω has also be tuned according to [20], where the values attributed to this parameter they usually linearly decrease from 0.9 to 0.4. As a result, this solution is better than a constant value of $\omega = 0.65$. Therefore, the $\omega = [0.90.80.70.60.50.4]$ vector has been chosen to obtain the values for the inertia weight as the number of epochs decreases.

After tackling the motion planning problem, the tracking of the generated trajectory has been investigated. Since the wheels are different (Mecanum), it requires a different kinematic model of the robot. For instance, the kinematic model for a car-like mobile robot with four Mecanum wheels has been implemented with Equation 3 according to [18].

$$\begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} = \frac{R}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ \frac{-1}{a+b} & \frac{1}{a+b} & \frac{-1}{a+b} & \frac{1}{a+b} \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{pmatrix} \quad (3)$$

Where ω_k is the angular velocity for each wheel ($k = 1..4$), R is the radius of the wheel, a and b are respectively the distances on x and y axis from the centre of the robot and the wheel axis, v_x and v_y are the velocity vector components and ω is the angular velocity of the robot. This model is mandatory to design the closed-loop controller based on the inverse kinematic model, which transforms the tracking errors to independent control of each wheel (Eq. 4). After measuring the required parameters to replace in the models, the following values have been obtained: $R = 0.003m$, $a = 0.125m$ and $b = 0.130m$. Then, a sliding mode controller has been chosen for its robust properties, similarly to [12] and the Matlab/Simulink environment is used to tune its parameters, as depicted in Fig. 3.

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{pmatrix} = \frac{1}{R} \begin{pmatrix} 1 & -1 & -(a+b) \\ 1 & 1 & (a+b) \\ 1 & 1 & -(a+b) \\ 1 & -1 & (a+b) \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} \quad (4)$$

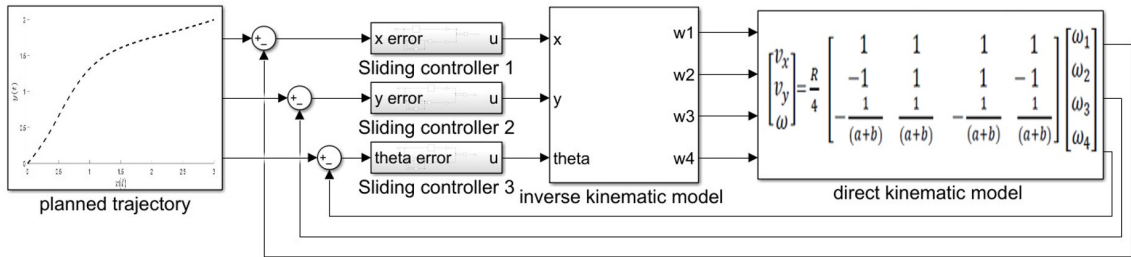


Figure 3: Simulink model used for controller parameter tuning

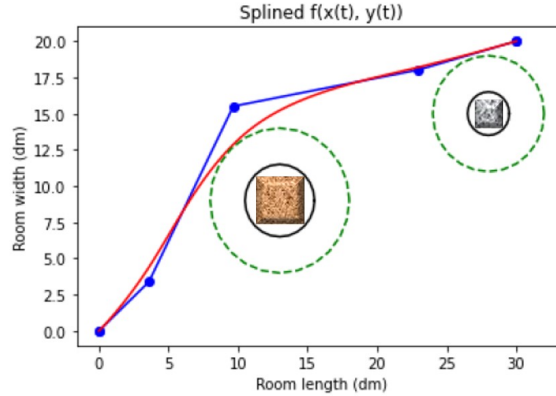


Figure 4: Above view of AMR environment and motion planner results from PSO

3.3 Results and experimentation

To highlight the operational capabilities of the robot after the implementation of the proposed solution to the problem, a study case has been set up inside the laboratory. The environment where the robot has to operate consists of a small surface 3m long and 2m wide with two obstacles, with a short range one. The reference point which helped calculate the position of the obstacles is, for our experiments, situated at $(x, y) = (0, 0)$. Figure 5 shows an above-view of the room layout and the results given by the motion planner after running the Python code with a swarm size = 200 and a number of epochs = 20. It depicts as well the resulted path that the robot needs to track and the already known obstacles. We can notice that this trajectory should be, for AMR manufacturing application as in [12], a last second update to avoid an unpredicted obstacle.

The blue line represent the control polygon which are searched by the algorithm; the red line is the trajectory obtained after applying the spline function to obtain the B-spline curve. The black circles circumscribe the physical obstacles with their corresponding radius and the green dotted circles are the maximum limit that the red line could cross (obstacle radius augmented by the robot size). The provided results shows the effectiveness of the designed algorithm to provide collision-free trajectory.

After running the Simulink model presented in Figure 3, the computed path that the robot is supposed to track, according to the values received from the python code, is plotted in Figure 5.

Figure 6 shows four different pictures of the implemented Lego robot successfully avoiding the obstacle without having to change its orientation. Also, a full demonstration of the robot's performance could be found online¹.

This experiment shows the advantage of omnidirectional wheels and the possibilities to

¹<https://www.youtube.com/watch?v=QkRbZMfBBgk>

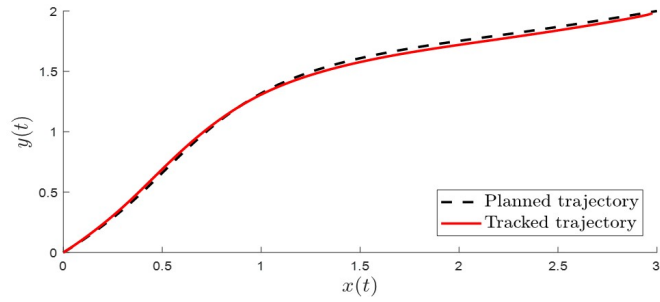


Figure 5: Path tracking results from simulation

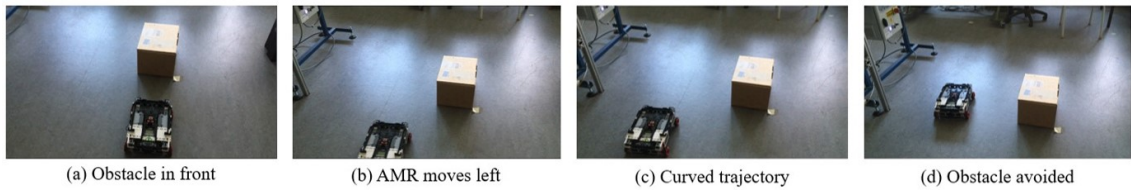


Figure 6: Pictures of AMR avoiding the obstacle

implement them in manufacturing processes. Indeed, by equipping the robot with this type of wheels, not only that we benefit from the high degree of freedom but also, the robot is able to avoid imminent obstacle in a manner that the conventional wheels could not, as shown Figure 7. This ability clearly contribute to agility since the AMR become capable to adapt its plan, without important delay and with few disturbances on others AMRs (or other entities) that may cross the area where the AMR is navigating/manoeuvring.

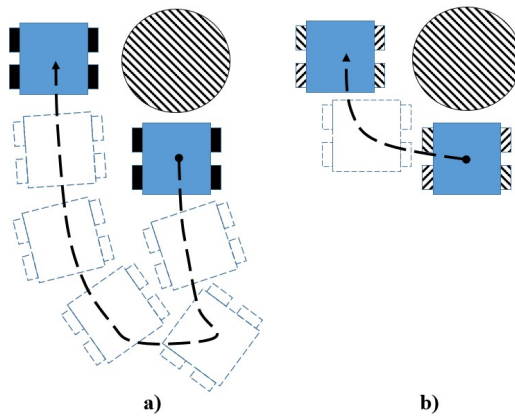


Figure 7: Obstacle avoidance by AMR with a) conventional wheels, b) Mecanum wheels

4 Conclusion

In the manufacturing domain, the constant changes in product demand lead to the need for a manufacturing system which is capable to cope with agility requirements. As a response to this need, a so-called Reconfigurable Manufacturing System (RMS) has been proposed. This system introduces a manufacturing system which is superior in comparison with the previously proposed one in terms of product demand, flexibility, machine structure, product family formation, productivity, systems structure and variety. This paper mainly focused on the implementation of the solution to solve the material transportation problem in a Reconfigurable Manufacturing System. A literature survey has been deducted for this problem, and among the most widely used transportation means in manufacturing, we could enumerate conveyors, AGVs and AMRs. Despite their easy implementation and low cost compared with the other two, the conveyors do not offer the required flexibility to implement a RMS. Even though the AGVs are a more flexible material transportation solution, they still lack primarily the convertibility aspect which AMRs have. To reach the sufficient flexibility that contribute to agility, we focus on the use of omnidirectional wheels for AMRs while implemented the required navigation tools. The two main navigation problems for an AMR represent the path planning and the path tracking. For the path planning, heuristics and meta-heuristics approaches could be found in literature and among them we could enumerate GA, PSO or ACO. The proposed solution to the path planning, which is the PSO implementation, includes the following main advantages, such as easy implementation, low computational cost and simplicity. The biggest issue that has been encountered during this project was manoeuvring of the Lego Robot. As all these robots are only used for educational purposes, their capacities are limited in terms of command precision. As it has been mentioned in Section 4, the tracked trajectory that the robot executes compared with the planned trajectory presents some errors. The lack of robustness in the mechanical assembly of the robot, especially for the assemblage structure of the wheels which do not benefit from any suspension system, affected the route tracking. We could also underline that the robot's final performance is affected by the vibrations and wheel slippage. Regardless of the robot's performances that were limited by external factors, its successful implementation represents a very good lead for future developments in material handling, especially for manufacturing system.

References

- [1] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa, J. O. Strandhagen, Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics, *Annals of operations research* (2020) 1–19.
- [2] Y. Y. Yusuf, M. Sarhadi, A. Gunasekaran, Agile manufacturing:: The drivers, concepts and attributes, *International Journal of production economics* 62 (1999) 33–43.
- [3] M. Bortolini, F. G. Galizia, C. Mora, Reconfigurable manufacturing systems: Literature review and research trend, *Journal of manufacturing systems* 49 (2018) 93–106.
- [4] O. K. Efthymiou, S. T. Ponis, Current status of industry 4.0 in material handling automation and in-house logistics, *International Journal of Industrial and Manufacturing Engineering* 13 (2019) 1370–1374.

- [5] G. Fragapane, R. de Koster, F. Sgarbossa, J. O. Strandhagen, Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda, *European Journal of Operational Research* 294 (2021) 405–426.
- [6] G. Demesure, H. Bril El-Haouzi, B. Iung, Mobile-agents based hybrid control architecture—implementation of consensus algorithm in hierarchical control mode, *CIRP Annals* 70 (2021) 385–388.
- [7] J. Rigelsford, Introduction to autonomous mobile robots, *Industrial Robot: An International Journal* (2004).
- [8] K. Shabalina, A. Sagitov, E. Magid, Comparative analysis of mobile robot wheels design, in: 2018 11th International Conference on Developments in eSystems Engineering (DeSE), IEEE, pp. 175–179.
- [9] P. Marin-Plaza, A. Hussein, D. Martin, A. d. l. Escalera, Global and local path planning study in a ros-based research platform for autonomous vehicles, *Journal of Advanced Transportation* 2018 (2018).
- [10] E. Masehian, D. Sedighizadeh, Classic and heuristic approaches in robot motion planning—a chronological review, *World Academy of Science, Engineering and Technology* 23 (2007) 101–106.
- [11] I. A. Sukan, M. Moll, L. E. Kavraki, The open motion planning library, *Robotics & Automation Magazine, IEEE* 19 (2012) 72–82.
- [12] G. Demesure, M. Defoort, A. Bekrar, D. Trentesaux, M. Djemai, Decentralized motion planning and scheduling of agvs in an fms, *IEEE Transactions on Industrial Informatics* 14 (2017) 1744–1752.
- [13] H. S. Dewang, P. K. Mohanty, S. Kundu, A robust path planning for mobile robot using smart particle swarm optimization, *Procedia Computer Science* 133 (2018) 290–297. International Conference on Robotics and Smart Manufacturing (RoSMa2018).
- [14] C. Lamini, S. Benhlima, A. Elbekri, Genetic algorithm based approach for autonomous mobile robot path planning, *Procedia Computer Science* 127 (2018) 180–189.
- [15] Z. Nie, H. Zhao, Research on robot path planning based on dijkstra and ant colony optimization, in: 2019 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), IEEE, pp. 222–226.
- [16] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, C.-C. Peng, Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges, *Sensors* 18 (2018) 3170.
- [17] A. Downs, Z. Kootbally, W. Harrison, P. Pilliptchak, B. Antonishek, M. Aksu, C. Schlenoff, S. K. Gupta, Assessing industrial robot agility through international competitions, *Robotics and Computer-Integrated Manufacturing* 70 (2021) 102113.
- [18] E. Maulana, M. A. Muslim, V. Hendrayawan, Inverse kinematic implementation of four-wheels mecanum drive mobile robot using stepper motors, in: 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA), IEEE, pp. 51–56.
- [19] I. C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, *Information Processing Letters* 85 (2003) 317–325.
- [20] R. Eberhart, J. Kennedy, Particle swarm optimization, in: *Proceedings of the IEEE international conference on neural networks*, volume 4, Citeseer, pp. 1942–1948.