Application of Realtime Robotics platform to execute unstructured industrial tasks involving industrial robots, cobots, and human operators

Article *in* Procedia Computer Science · March 2022

DOI: 10.1016/j.procs.2022.01.337

Ahmed Zaki
University of Pavia

Marco Carnevale
University of Pavia

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 200 (2022) 1359–1367

Procedia Computer Science

www.elsevier.com/locate/procedia

3rd International Conference on Industry 4.0 and Smart Manufacturing

Application of Realtime Robotics platform to execute unstructured industrial tasks involving industrial robots, cobots, and human operators

Ahmed Magdy Ahmed Zaki[a,*], Ahmed Mohsen Mohamed Fathy[a], Marco Carnevale[a], Hermes Giberti[a]

[a]*Dipartimento di Ingegneria Industriale e dell'Informazione, Università di Pavia, via Ferrata 5, 27100 Pavia, Italy*

Abstract

Small and medium-sized craft enterprises are still characterized by production process involving human labour to a large extent. The possibility of a safe interaction of the operator in the collaborative working area of the machine, and the development of an easy-to-program robotic cell, would pave the way to the introduction of automation in such an environment. This paper proposes an application of a commercial platform for collision avoidance, based on Dynamic Road-map algorithms, to execute unstructured industrial tasks involving industrial robots, cobots, and human operators at the same time. An example is presented in a pick-and-place and assembling application, in which the components are feed by a conveyor belt with random order and random flow rate. The adopted platform enhances the possibility to exploit in the same robotic cell both industrial robots and human labour, thus allowing a reduction of costs and more precise applications. Moreover, the possibility to calculate at run-time alternative trajectories to avoid obstacles (both human being and other robot in the cell), avoiding to make the robot temporarily inactive, allows to improve the system productivity.

Peer-review under responsibility of the scientific committee of the 3rd International Conference on Industry 4.0 and Smart Manufacturing

Keywords: Industrial robots, cobot, collision avoidance, collaborative industrial tasks, human robot interaction;

1. Introduction

The noticeable progresses occurred in robotics have enabled the possibility to deploy robotic manipulators to execute new tasks that were impossible only few years ago. Despite that progress, collision avoidance is still an open topic to be completely solved, particularly in the case of highly dynamic and unstructured industrial environments in which the work-floors are shared between collaborative robots, autonomous mobile robots, human operators and

* Ahmed Magdy Ahmed Zaki. Tel.: +39-038-298-5264
E-mail address: ahmedmagdyahme.zaki01@universitadipavia.it

Peer-review under responsibility of the scientific committee of the 3rd International Conference on Industry 4.0 and Smart Manufacturing
10.1016/j.procs.2022.01.337

possibly industrial robots.

Collision avoidance in industrial robots has been the main topic of many researches for more than twenty years [1]. Many techniques have been proposed to avoid, dynamically, the obstacles blocking the robot's path, while executing industrial tasks. In the case of industrial robots, that are usually found behind fences and safety light barriers, the obstacles to avoid are other industrial manipulators, sharing the same work space, or the work-pieces handled during the industrial operation. The paper in [2] reports an example of a recent collision avoidance algorithm, able to coordinate multiple SCARA robots sharing the working space, and based on recurrent neural networks and Quadratic Programming. The algorithm corrects the previously programmed trajectory when a minimum distance between robots is reached.

Since 2016, after the publication of the ISO/TS 10566 [3] Technical Specification, the subject of collision avoidance has been further developed by considering possible interactions between man and machine. The mentioned standard defines the technical characteristics that a collaborative robot must be equipped with (i.e. limitation on speed and power), and the characteristics to be accomplished by the shared work space between a human operator and the cobot. However, in [4] the authors show that following ISO/TS 10566 for limiting speed and force allows to limit the entity of the impact between robots and human operators, certainly containing the effects but not excluding injuries.

In unstructured environments, to guarantee the safety of human operators, collision avoidance techniques have to be highly responsive, possibly to act in time period shorter than human reaction time. The system has to perceive collision, re-plan the action and execute it in the shortest possible time. To increase the cognitive abilities of robotic systems, 3d vision systems can be used to detect obstacles [5]. In [6], the collision avoidance technique proposed is based on the detection of the human pose using a combination of sensors (e.g. wearable sensors like magnetic tracking sensors and IMU and a laser scanner) to detect different body regions.

Collision avoidance techniques can be divided in different classes [7]: reactive techniques, based on artificial potential fields, have the limitation of considering only local information of obstacles at the near proximity of the robot's body. This may generate a local minimum problem. The second category consists in path planning approaches. Rapidly Exploring Random Tree (RRT) reaches good performance in dynamic environments but the calculation at every time step of every possible collision path asks for high computational effort that makes it infeasible for real-time applications. Probabilistic Road-map (PRM), through the preprocessing of the static environment, speeds up the planning process that makes it feasible for real-time planning but with the limitation of working only in static environment. To decrease the response time of the system in dynamic environments, Dynamic Road-map (DRM) could be a feasible solution. It creates, through offline processing of the environment and of static obstacles, a road-map of all possible paths between every possible poses in the robot working space, which is later used at run-time to avoid dynamic obstacles. For the online re-planning of the trajectory it is necessary to invalidate the blocked paths and to use an optimization procedure to determine the best feasible trajectory among the remaining valid ones.

There are few products on the market featuring collision avoidance solutions with industrial performance (e.g. short reaction time, reliability, ease of setup and flexibility to control different robots from different manufacturers). In this paper we exploit a Realtime Robotics platform [8] to design a robotic solution to be deployed in a highly dynamic and unstructured industrial environment, representative of most of cases found in small and medium craft enterprises. The robotic cell is made of both a collaborative robot (Mitsubishi Assista) and an industrial robot (RV-2FRB Mitsubishi), sharing the work space both among each other and together with a human operator. Realtime Robotics platform [8] is a dedicated commercial industrial computer able to manage the execution of industrial tasks and collision avoidance between several robots while handling the products, as well as the collision with human operator sharing the work space.

The aim of the work is to assess the actual capabilities of the system, which would pave the way to a number of applications in unstructured, craft enterprises [10]. The paper is organized as following: Section 2 describes the proposed system architecture developed, both in terms of simulation and experimental robotic cell. Section 3 describes the developed task. Two cases are considered: firstly, the collision avoidance between two robots sharing the same area. Secondly, the interaction with a human operator is added to the previous case, so that possible interactions between an industrial robot, a collaborative robot and a human operator are taken into account. Conclusions are drawn in Section 4.

2. Proposed system architecture

The robotic cell design deployed has the purpose to allow robotic manipulators to execute operations commonly required in craft enterprises, in which the presence of a human operator sharing the work space with the robots must be considered as an essential factor. Exemplary tasks required to robots may be pick and place, product handling or product finishing. The robotic cell shown in fig. 1 consists of two robotic manipulators, an RV-2FRB Mitsubishi industrial robot and a Mitsubishi Assista collaborative robot. The robots are placed near to each other, in a shared work space. A set of 4 Intel Realsense D435 3D stereo cameras allows to monitor continuously the working cell, detecting the position of each obstacle or human operator approaching any of the two robots. The adoption of the Realtime Robotics platform allows to control in real-time the trajectories of both the industrial and collaborative robots, enabling the possibility to exploit the former even in the presence of a human operator, and thus leading to reduced costs, higher accuracy and higher stiffness, which are all limiting factors in the use of a collaborative robots.
The development of a robotic cell exploiting this control platform would therefore enable the adoption of industrial robots not enclosed behind safety fences. To the authors' knowledge, the approval process of the mentioned platform for collaborative applications is still in progress, nevertheless the assessment of the system performance from a technical point of view is urgent, due to the great benefit that the system would bring to craft manufacturing enterprises.

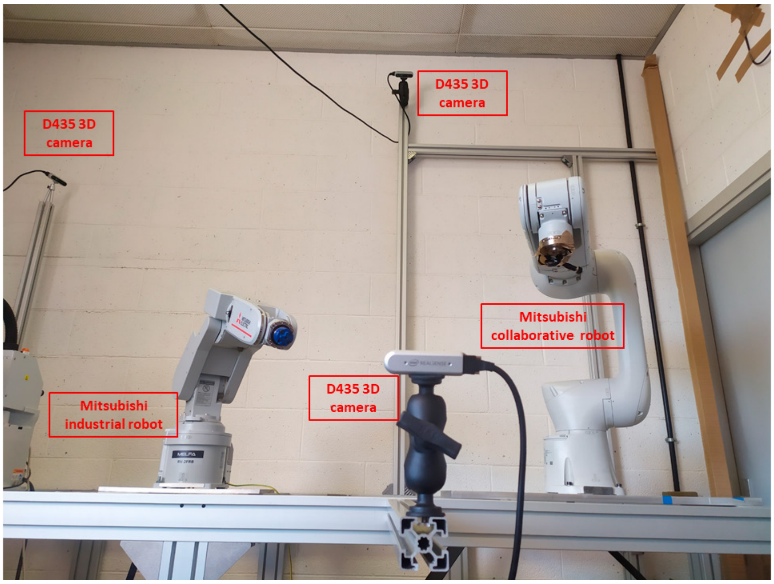


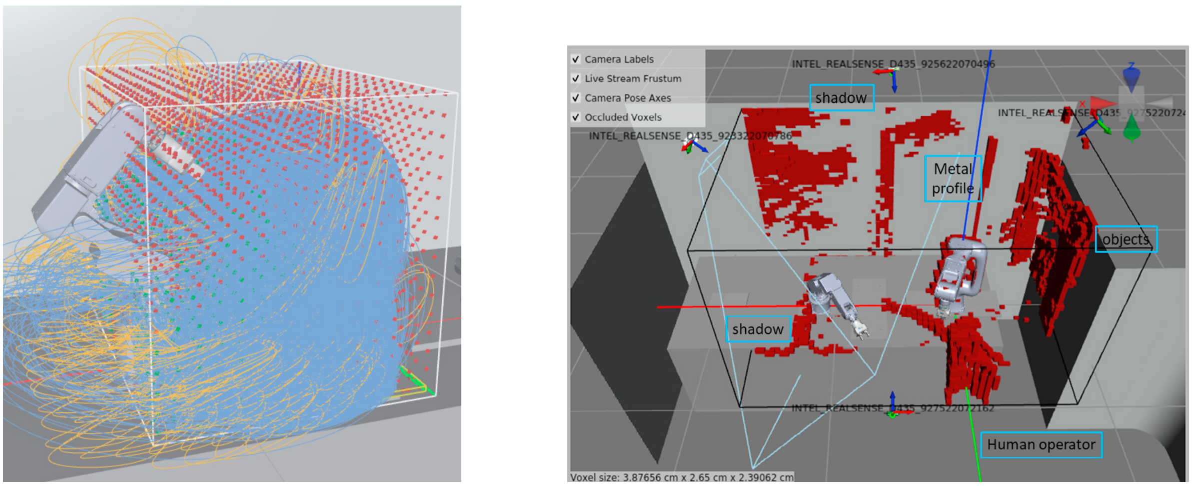
Fig. 1. Robotic cell setup

The Realtime Robotics platform allows the coexistence of more than one robot (up to four) and of a human operator in the same work space, planning trajectories in real-time to avoid any possible collision. This would permit to overcome typical techniques based on laser scanners in industrial environments [9], like Monitored stop or speed and distance monitoring. These techniques provide good solutions from human safety point of view. However, in situations where the human may enter frequently the working area, they have the drawback of the productivity reduction, due to repeated slow down of the task execution speed, or in the worse cases to several stop periods.
In academia there are many techniques that are being proposed to manage collision online, without stopping several times the robotic manipulators. However, a lower number of solutions is available at commercial level. Among those, Realtime Robotics controller is an industrial platform that is used to design robotic cells where collision avoidance has to be guaranteed.
The working principle of the real-time controller can be divided into the following main steps:

1. Creation of 3D model of the robotic cell using RapidPlan software: importing CAD models of robots, end-effectors, tables, walls and all static obstacles. This model is then used to design the collision behavior of each

robot. Firstly, the collision region, where a collision may happen, is defined. This is represented in the white box in 2a. This region is then discretized in a grid made by many robot poses. Secondly, the poses to be reached by each robot to execute its task are defined. The discretized region and the task related poses are then used to determine the working road-map, which is built by the calculation, for every pose, of all the valid paths to reach the rest of the poses. The total number of paths calculated, shown as blue and yellow trajectories in 2a, could reach hundreds of thousands. This process is therefore performed offline, to simulate the behavior of the robots and the feasibility of the task.

2. Online perception of obstacles using RapidSense software: at run-time, by using the set of Intel Realsense D435 3D stereo cameras shown in fig. 1, the system is able to monitor continuously the working cell. When an obstacle or human operator approaches the robots, the system detects its position by the 3D cameras and invalidate the occupied paths of the previously calculated road-map. As an example, fig. 2b shows the obstacles detected by the system from the perspective of the left side robot (RV-2FRB Mitsubishi industrial), some of which are not present in the 3D model created offline. The human operator with an extended arm is clearly visible in the bottom-right corner, as well as the upright profile in proximity of the right robot and other objects laying on the right wall and not related to the presented application. Also the right side robot (Mitsubishi Assista collaborative) is considered as an obstacle for the left one. Some shadows are detected in the left top corner, which nevertheless do not affect the final application, being in the immediate vicinity of the wall. Other shadows are detected below the left side robot, corresponding to the horizontal profiles supporting the robot. Also these shadows do not limit the working space in a significant way, being very close to the actual obstacle.
3. Online trajectory re-planning: from the remaining valid paths, the best one is determined through a minimization procedure, and the movement command is then sent to the robot to be executed. The system is equipped with a proprietary processor, tailor-made to accelerate the search process to find the optimal valid path.



(a) (b)

Fig. 2. Simulation a) Roadmap of feasible trajectories b) Obstacles in the Roadmap

3. Task description

The robotic cell introduced is used to execute industrial tasks that require collision avoidance. The tasks considered are the following:

3.1. Industrial task with two robots

The task consists in an unstructured pick-and-place application of two kinds of products, coming at random time and with random sequence on a conveyor belt. The objects to be picked, having different dimensions, also presents different features and geometries. They are detected and recognized by a 3D vision system based on a further Intel Realsense D435 camera of the same type exploited in the main bench.

Since the products are placed randomly on the conveyor belt with random order and flow rate, it is not possible to program a priori the order of the sequence each manipulator is involved in. Should two products arrive to the picking area very close to each other like in fig. 3, an interference is likely to occur between the two robots. In a traditional multi-robot task, this circumstance would lead to the imposition of an idle time to one of the robots, which should wait for the other to fully accomplish its operation. This idle time may prevent to pick the product in the due time [11], thus decreasing the picking rate and the overall efficiency of the multi-robot system. On the other hand, in the proposed application, the run-time definition of alternative trajectories enables both the robots to work simultaneously, for the benefit of system productivity.

The identification of the object type, as well as the definition of the picking poses to be fed to the robot controller in real time, are carried out by means of a dedicated Python algorithm, exploiting the open source libraries OpenCV and Librealsense. The algorithm elaborates the images acquired by the Intel D435 3D stereo depth camera of the picking area of the conveyor belt in the following steps:

1. **Detection of the coming product:** the 3D vision system allows to acquire, at the speed of 30 frame per second, separate color and depth images of the scene within its field of view. The considered objects have a different color with respect to the conveyor belt. By elaborating the colored image through cv.findContours function from OpenCV library, all the spots with different colors are detected. After finding the objects, it is possible to acquire useful related features like their area. Based on this value the objects are assigned to different robots.
2. **Evaluation of picking positions:** for the detected objects, a second feature calculated is the coordinates of the object center X_c and Y_c, which are used to obtain the depth value of the object center, Z_c, from the depth image. These pose coordinates are calculated in pixels in the camera reference frame. They are then geometrically transformed in the robot reference frame using a homogeneous transformation matrix. This matrix is obtained through a calibration procedure in which a well defined point or a set of points are being detected in the camera reference frame. The robot end-effector is then moved manually to reach the point or the points. These point poses are then used to determine the right homogeneous transformation matrix between the two set of values with respect to the two reference frames. Finally these objects' center pose is communicated to the robots as picking pose.

In fig. 3 an example of the objects that the system is able to identify. As an example, in the demonstration cell small products (green rectangle) are assigned to the industrial robot RV-2FRB and the biggest (blue rectangle) to the Collaborative Assista robot. The green dots represent the centers of the objects, which are used to calculate the picking pose of each robot.

3.2. Collaborative industrial task

As a further step, the presence of an operator is added to the previous task. The two robots, picking different components as described in the previous task, work in the same area of an operator in charge of assembling the components. The assembly area is a shared area between the human operator, the industrial robot and the collaborative robot. To accelerate the cycle time of the task, the RealTime robotics platform is used to manage the online trajectory planning of the two robots, so as to guarantee the absence of collision between the two robots, and also guaranteeing the safety of the human operator. In this way, the robots follow collision-free trajectories instead of stopping and waiting for the obstacle to be removed from their trajectory.

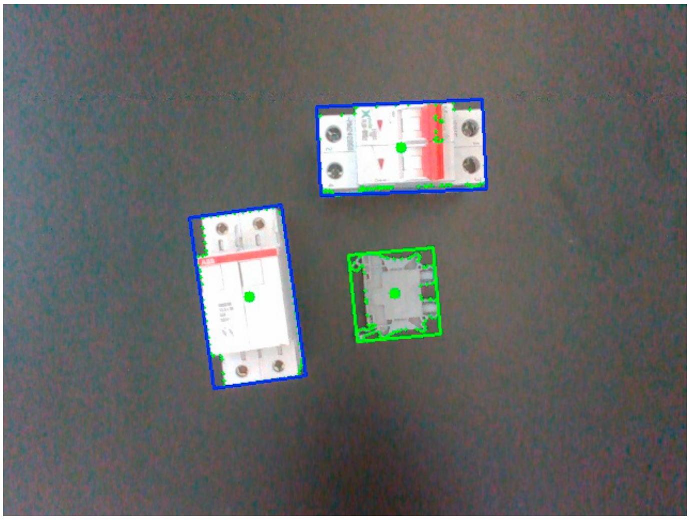


Fig. 3. Objects to be picked

4. Experimental tests and Results

This section describes the results of the tests carried out to compare the performance of the proposed system architecture with other solutions that are commonly used for collision avoidance management. As a Key Performance Indicators (KPI), we consider the total cycle time required to perform the required task, which allows to assess the production flow rate of the overall system.

In the tested case the end effector speeds are set to $8000mm/s$. Each of the two robots has to pick a different object placed at the following coordinates referenced its own base: $pick = [600, 500, 200]$ [mm]. The other positions related to the execution of the task (i.e. place and home) are also referenced with respect to each robot base, and are: $home = [400, 0, 900]$ [mm] , $place = [600, -500, 200]$ [mm]. The working cycle consists in the movement of the robot to the picking position and then the movement to the placing position passing through the home position.

4.1. Industrial task with two robots

4.1.1. Sequential movement

In an industrial application the two robots would be enclosed behind safety fences and no interactions with human operators would be intended. To execute the unstructured pick and place task, where objects could arrive close to each other in the picking area over the conveyor belt, the traditional technique used in industry to avoid collisions between the robots is the sequential movement, in which the robots execute sequentially the picking and placing processes of the assigned objects. In this way, one of the robots remains stopped until the other finishes its task and clear the working area.

The total cycle time to pick and place two objects is then the sum of the single cycle times of the robots and can be calculated in the following way:

$$T_t = T_{r1} + T_{r2} \tag{1}$$

$$T_{ri} = T_{move} + T_{pick} + T_{place} \tag{2}$$

Where T_t is the overall cycle of the task and the T_{ri} is the time needed for robot i to pick and place the assigned object. T_{move} is the time required to move during the pick and place cycle. T_{pick} and T_{place} are the times needed for the gripper to open and close to grip and release the object. With reference to the positions defined in the test case, the total distance to be covered to execute the task is $d = 1897.3mm$. Since the movement speed is $v = 8000mm/s$ the $T_{move} = d/v = 237ms$ (neglecting acceleration and deceleration phases). From the data-sheet of the used Onrobot

2FG7 gripper [12], the picking and placing time can be estimated with reference to the defined gripper opening distance and opening speed. It results equal to $365ms$.
From equation 1 and 2 the total cycle time of the single robot T_{ri} and total cycle time T_t are equal to:

$$T_{ri} = T_{move} + T_{pick} + T_{place} = 237 + 365 + 365 = 967ms \tag{3}$$

$$T_t = 2 * T_{ri} = 2 * 967 = 1934ms \tag{4}$$

4.1.2. *Proposed solution exploiting Real Time Robotics platform*

The robotic cell equipped with Real Time Robotics platform allowed to execute the unstructured pick and place tasks in a sensibly shorter time. The platform continuously re-plans in real time the trajectories of the robots to not collide with the other one, so that idle times are almost cancelled. The cycle time taken by the robot i to execute the pick and place task is described as follows:

$$T_{ri} = T_{move} + T_{pick} + T_{place} + T_{REACTION/RECONFIGURATION} \tag{5}$$

Where $T_{REACTION/RECONFIGURATION}$ is the time needed to change the robot configuration to avoid an obstacle, and to start following the new collision free path. This time is in order to few milliseconds, whereas the time needed to find the best collision free path from all the valid paths does not influence the total cycle time, this process being in parallel with the movement. As a consequence of the new collision-free path, the time T_{move} to execute the pick and place movements is slightly changed, depending on the new path length. The total cycle time is therefore mainly given by the longest of the cycle times of the two robot ($T_t \approx T_{ri}$).
Based on the cell volume defined in the tested application, the maximum overall cycle times detected during tests was $1100ms$, which is sensibly reduced compared to the standard reference industrial application of Equation 4.

4.2. *Collaborative configuration*

In the case of collaborative robotic cell, a traditional technique industrially adopted is Speed and Separation Monitoring. The performance of the developed cell exploiting Real Time Robotics platform are therefore discussed in comparison to this technique.

4.2.1. *Speed and separation monitoring*

To apply speed and separation monitoring, industrial sensors like the Safety laser scanner nanoScan3 [13] are used, allowing to detect the presence of obstacles or a human operator as soon as they enter a predefined area around the robotic cell. Based on the distance between the obstacle and/or operator and the robot, the sensor behaves differently between decreasing the robot's speed or stopping it until the obstacle exits the dangerous area. To avoid collisions between the two robots, sequential movement explained in 4.1.1 is used. In this case, the cycle time is calculated in the following way:

$$T_t = T_{r1} + T_{r2} + T_{stop} \tag{6}$$

Where T_{stop} is the stopping time when an obstacle or an operator enters the stopping area near the working cell. The overall cycle time is highly dependent on the time of the presence of the operator or the obstacle in the dangerous area and this increases significantly the total cycle time decreasing the productivity. Due to theses motivations, this technique could have acceptable performance in applications where the presence of the operator is not frequent but is still possible to happen.

4.2.2. *Proposed solution exploiting Real Time Robotics in collaborative mode*

To execute the task where the robots avoid the operator, the platform as explained in 4.1.2 re-plans continuously the trajectory of the robots to avoid obstacles. The human operator is treated in the same way. In this configuration, the robotic cell design proposed is able to operate without the stopping time related to the presence of the operator and without the delay due to the application of sequential movement to avoid collision between the robots, as explained in 4.1.1.

Experimental tested carried out up to now, showed that the system is able to avoid the human operator, so that the total cycle time can be still approximated to the cycle time detected in the absence of a human operator (i.e. $T_t \approx 1100ms$). The robotic cell design proposed has therefore good performance executing a collaborative assembly tasks where the presence of the operator is permanent or highly frequent in the shared working area with the robotic manipulators.

4.3. Discussion

Realtime Robotics is a good industrial solution for managing collision easily in industrial environments. It offers good flexibility, both in term of hardware, allowing to control the most common robotic manipulators, and in terms of software, allowing to program the robotic task in many languages commonly used in industrial environments (e.g. Python, C, C++, PLC code and finally robot manufacturers' proprietary codes). In this work the tasks are programmed in Python, that is easily integrable with open source libraries like OpenCV for computer vision. The system performed well in enabling each robot to respond to changing environmental conditions.
Some of the encountered drawbacks are the necessity to use the software for the offline definition of the road-map exclusively in conjunction with the hardware provided Realtime Robotics. The same hardware and software must be used too as an interface between the robots and the task planner software (e.g. Python, C, or PLC). Besides, the system performs complicated offline calculations, so that the process of changing the project layout may be time consuming. Finally, the implemented technique is based on the detection of obstacles using 3D stereo cameras, which makes it highly dependent on environment illumination.

5. Conclusion

The paper discusses the deployment of a robotic cell equipped with a commercial system for collision avoidance (i.e. Realtime robotics platform). The system is based on the Dynamic Road-map (DRM) algorithm, and allows run-time trajectory planning for collision avoidance, both among robots and human operators. These kinds of systems are still not widely applied in industrial applications, so that deepening their behaviour would enhance the evolution of agile robotics, with huge benefits especially in small and medium-sized craft enterprises.
The paper proposes the design of a robotic cell to execute industrial tasks. The considered one is a pick and place application featuring different sizes of the objects and feeding on a conveyor belt with random order and timing. The deployed system allowed to manage the real-time trajectory planning of the two robots, exploiting the robot simultaneously even under the circumstance in which the objects to be picked are fed very close to each other on the conveyor belt. This allowed to reduce or even avoid idle times, and thus increase productivity. The system also allowed to use an industrial robot in a shared work space with a human operator, guaranteeing no contact between them. Some limitations of this study have to be pointed out: the tests were carried out with fixed objects in the picking position, whereas in the final application they will be moving on the conveyor belt. This might affect the performance of the vision system in identifying the objects and driving the two robots with related picking poses. Moreover, the performance of the collision avoidance system must be tested under this circumstance.

References

[1] E. Freund, M. Schluse and J. Rossmann, "Dynamic collision avoidance for redundant multi-robot systems," Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180), 2001, pp. 1201-1206 vol.3, doi: 10.1109/IROS.2001.977146.

[2] Yuhong He, Xiaoxiao Li, Zhihao Xu, Xuefeng Zhou, Shuai Li, Collaboration of multiple SCARA robots with guaranteed safety using recurrent neural networks, Neurocomputing, Volume 456, 2021, Pages 1-10, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2021.05.049.

[3] Robots and robotic devices — Collaborative robots

[4] M. J. Rosenstrauch and J. Krüger, "Safe human-robot-collaboration-introduction and experiment using ISO/TS 15066," 2017 3rd International Conference on Control, Automation and Robotics (ICCAR), 2017, pp. 740-744, doi: 10.1109/ICCAR.2017.7942795.

[5] Abdullah Mohammed, Bernard Schmidt & Lihui Wang (2017) Active collision avoidance for human–robot collaboration driven by vision sensors, International Journal of Computer Integrated Manufacturing, 30:9, 970-980, DOI: 10.1080/0951192X.2016.1268269

[6] Mohammad Safeea, Pedro Neto, Richard Bearee, On-line collision avoidance for collaborative robot manipulators by adjusting off-line generated paths: An industrial use case, Robotics and Autonomous Systems, Volume 119, 2019, Pages 278-288, ISSN 0921-8890, https://doi.org/10.1016/j.robot.2019.07.013.

[7] T. Kunz, U. Reiser, M. Stilman and A. Verl, ”Real-time path planning for a robot arm in changing environments,” 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 5906-5911, doi: 10.1109/IROS.2010.5653275.
[8] Realtime Robotics official website: https://rtr.ai/
[9] https://www.universal-robots.com/plus/urplus-components/accessories/sick-safety-sbot-stop-package/
[10] Castelli, K.; Zaki, A.M.A.; Dmytriyev, Y.; Carnevale, M.; Giberti, H. A Feasibility Study of a Robotic Approach for the Gluing Process in the Footwear Industry. Robotics 2021, 10, 6. https://doi.org/10.3390/robotics10010006
[11] Zaki A.M.A., Carnevale M., Castelli K., Giberti H. (2021) A Simplified Approach to Size Multi-robot System for Pick and Place Applications. In: Niola V., Gasparetto A. (eds) Advances in Italian Mechanism Science. IFToMM ITALY 2020. Mechanisms and Machine Science, vol 91. Springer, Cham. https://doi.org/10.1007/978-3-030-55807-9_38
[12] https://onrobot.com/sites/default/files/documents/Datasheet_2FG7_v1.1_EN.pdf
[13] https://www.sick.com/it/en/p/p653980