

Anytime informed path re-planning and optimization for human-robot collaboration

Cesare Tonola¹, Marco Faroni², Nicola Pedrocchi², Manuel Beschi^{1,2}

Abstract—Robots working in proximity of humans often need to change their motion to avoid collisions and interference with the operators. This paper uses a path re-planning approach to change the robot path online when the human operator is in the robot way. The method exploits a set of pre-computed paths to compute a new feasible path in case of obstruction to enhance the trajectory’s readability. Moreover, the algorithm iteratively optimizes the current solution in an anytime fashion to deal with strict computing time requirements. Experimental results show the method’s effectiveness in a collaborative cell, compared with industry best practices.

Index Terms—Path planning; Anytime motion planning; Re-planning; Human-robot collaboration; Autonomous robots.

I. INTRODUCTION

Human-Robot Collaboration (HRC) is the research area that deals with the spatio-temporal overlap between humans and robots’ workspaces, and it has undergone a significant increase of interest in recent years [1]. The collaboration between humans and robots might be very advantageous for industries: on the one hand, robots are fast and precise, yet they are still unable to adapt their behavior to what is happening in the workstation; on the other hand, humans can react to environmental changes, but they are not repeatable and quickly tired. Their combination could allow exploiting the potential of both.

Most industrial solutions do not provide for a synergic collaboration between humans and robots. Usually, robots are enclosed into physical or virtual cages and, when the operator decides to enter, robots stop to guarantee operator safety [2]. Other approaches exist, such as the *speed and separation monitoring* (SSM) defined in the ISO-TS15066, which allows collaborative robots to operate without fencing but requiring to control the robot in such a way as to ensure that the robot can stop its movement to prevent collisions. The most widespread, yet conservative application of SSM provides that, depending on the operator position, the robot works at full speed, at reduced speed, or stops if the operator is inside a low, medium, or high danger area [3], [4]. In this case, the operator safety is guaranteed despite the absence of fencing, but this solution often reduces throughput. Real-time techniques have been implemented to reduce safety stops and optimize the speed reduction on a

pre-defined path, for example, via linear programming [5], PID control [6], or model predictive control [2]. However, if the operator continuously blocks the robot’s trajectory, the robot repeatedly reduces its speed or remains completely stationary, resulting in a substantial throughput reduction and weak support for the operator during the task execution. A natural and safe collaboration should require the robots to *rapidly change their path* to complete the task when humans interfere with their motion, avoiding any mobile obstacles with limited idle time.

Online path re-planning modifies an initial path during its execution to avoid collisions or to improve the trajectory. For example, it computes a new path when the current one is obstructed. Some works are based on RRT-like planners [7] and prune and modify the search tree when changes of the configuration space happen [8], [9], [10], [11]. Time dimension can be added to the search tree to plan a new path foreseeing possible future collisions with mobile obstacles [12], [13], [14]. Another typical approach consists of modifying the robot’s trajectory applying virtual repulsive forces to the robot to move it away from the obstacles [15]. These algorithms provide intuitive results; however, they are not efficient in managing complex cells, and they can get trapped in local optima.

From an HRC point of view, a re-planner should be fast enough to react to operator movements reducing as much as possible idle time, and it should find solutions easily predictable and readable by the operator. These are challenging requirements to be reached by typical approaches. Industrial environments, such as collaborative cells, make algorithms computationally challenging, resulting in the detriment of robot reactivity. Furthermore, only a local deformation of the current trajectory is often considered, resulting in a re-planned path that is far from optimal.

A. Contribution

This paper proposes a reactive motion planning strategy for HRC that combines a fast path re-planning and optimization algorithm [16] with speed modulation. This framework combines reactivity with the evaluation of multiple solutions to obtain an increasingly better result in an anytime fashion. The path re-planning algorithm searches online for new trajectories exploiting a set of paths computed offline. Notice that this is helpful in HRC as the obtained motion results to be more readable from the operator point of view. A speed monitoring module based on safety rules from ISO/TS 15066 [17] is then in charge of reducing the robot speed if the

This work is partially supported by ShareWork project (H2020, European Commission – G.A. 820807).

¹ Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia {c.tonola001,manuel.beschi}@unibs.it

² STIIMA-CNR - Institute of Intelligent Industrial Technologies and Systems, National Research Council of Italy {marco.faroni,nicola.pedrocchi}@stiima.cnr.it

human is close to the robot. The goal of the replanner is to drastically reduce the need to scale the speed of the robot.

The main contributions of this paper are:

- It proposes a novel planning framework for HRC applications combining online path re-planning and optimization [16] with speed modulation from ISO/TS 15066 [17];
- It evaluates the performance of the proposed approach with experimental tests, showing that the proposed solution outperforms a standard SSM strategy.

A video of the tests can also be found in the attached material.

This paper is organized as follows: the overall framework is presented in Section II; the reactive planning module is explained in Section III; experimental results are provided and discussed in Section IV; finally, conclusions are drawn in Section V.

II. HRC MOTION PLANNING FRAMEWORK

The proposed framework follows a common paradigm in HRC. As shown in Figure 1, it is made up of an offline *path planner* and a *reactive planner*, which is composed, in turn, by a *path re-planner* and a *speed modulation* module. The offline planner deals with the offline planning of the robot motion; the reactive planner provides a reactive behaviour to the operator movements. The framework relies on current information about the environment and the human state; this information is typically acquired through a vision system. In the next paragraphs, the two blocks will be explained in details.

A. Offline path planner

It computes the initial trajectory from a start position to the desired goal position. Based on information about the static obstacles, the algorithm computes paths in the robot configuration space using path planning algorithms. In our framework, the offline planner computes a set of feasible paths. Then, one of them is selected as the current path (typically the one which minimizes a cost function, *e.g.* the path length). The current path is parametrized by means of a path parametrization method (*e.g.*, TOPP [18]) and the result is the robot's nominal trajectory defined as a set of timed waypoints.

The proposed framework is independent from the path planning algorithm used to find the initial set of paths. Usually, sampling-based algorithms, such as RRT-Connect [19], are used to carry out this task for they can efficiently deal with high-dimensional search spaces of robot manipulators. It is worth noticing that the paths can be computed just before the movement, so that information on the human state at planning time can be exploited. This is why Figure 1 shows a dashed connection between the vision system and the offline planner. In this regard, human-aware path planners can be used to reduce the probability of collision with humans or to find paths that optimize human comfort [20], [21].

B. The reactive planner

It is at an intermediate layer between the offline planner and the robot controller. This layer provides the robot with a reactive behaviour to make it able to avoid the operator during the task execution to obtain a safe and efficient collaboration. As a matter of fact, the robot path may be obstructed by the human (which behaves like a moving obstacle) during the movement.

Most times, only a *speed modulation* block is implemented to slow down or stop the robot and prevent collisions. In this work, we deploy a tiered approach that also integrates a path re-planning module. In this way, the robot is able to change its path at run-time in case the current one becomes infeasible or is sub-optimal.

The online planner receives as inputs the set of paths computed by the offline planner, the current robot's trajectory, and the information about mobile obstacles from the vision system. As output, it provides the new robot state target to be given to the robot controller; this state is obtained sampling the new computed trajectory, based on the path modified by the re-planner and considering the velocity scaling calculated by the speed modulation technique.

Note that the strategy of switching between pre-computed paths offers advantages to HRC applications. Firstly, it enhances the search speed of the algorithm, improving the robot reactivity. Secondly, it allows to obtain more readable solutions for the operator since the robot switches between paths calculated offline and which can be known to the operator.

III. REACTIVE PLANNER

This section describes the algorithms used by the reactive planner. The first sub-section deals with the *path re-planning* method, the second one describes the *speed modulation* block.

A. Re-planning framework

The re-planning framework used is the one proposed in [16], which consists of three different threads:

- the *trajectory execution* thread;
- the *collision checking* thread;
- the *path re-planning* thread.

The *trajectory execution* thread receives the set of paths and the current robot trajectory computed by the offline planner. It samples the trajectory at a high rate and sends the new robot state target to the robot controller. In this case, the thread communicates with the speed modulation module to define the time instant along the trajectory at which it will be sampled. The *collision checking* thread deals with obtaining information from the vision system to verify if the current path, or one belonging to the set of pre-computed paths, collides. The *path re-planning* thread receives the current path, the set of the available paths, and the current robot configuration as inputs, tries to find a new path that optimizes the current one or which avoids mobile obstacles that are obstructing it and computes a new trajectory based on this

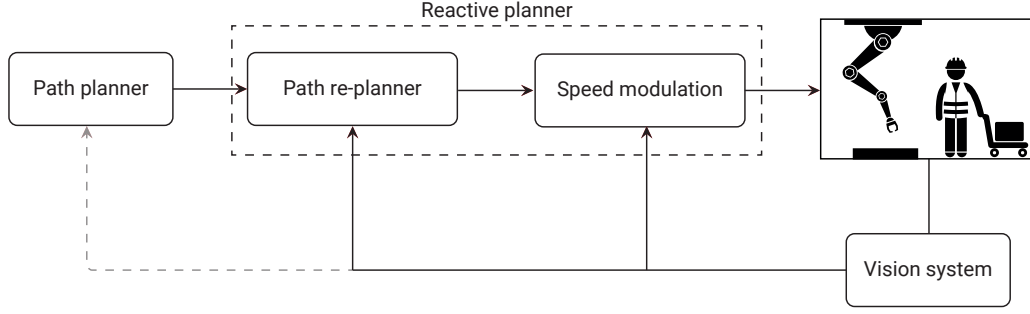


Fig. 1: Proposed planning scheme. The offline planner finds a set of paths from the initial to the desired final configuration and the best solution is executed while the path re-planning module keeps looking for improved solutions. The speed modulation module adapts the robot speed according to ISO/TS 15066.

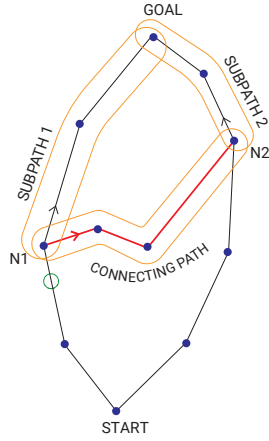


Fig. 2: Illustration of the *pathSwitch* algorithm. The green circle is the current robot configuration on the current path. The algorithm searches for a feasible path from a node of the current path, e.g. N_1 , to a node of the other available paths, e.g. N_2 .

new path. Finally, it sends this new trajectory to the *trajectory execution* thread.

The re-planning takes place through two algorithms that communicate with each other, *pathSwitch* and *informedOnlineReplanning*, which will be briefly explained in the following paragraphs. See [16] to get more information and pseudocode.

1) *pathSwitch* algorithm: *pathSwitch* algorithm aims to create a path from a node N_1 of the current path to one of the other available paths. As input, it receives the current path, the set of the available paths, and the node N_1 of the current path from which starting to search a new one. It tries to find a connecting path from N_1 to each node N_2 of all the available paths. The solution will be made up of the connecting path and of the subpath from N_2 to the goal, as represented in Figure 2. This search is conducted smartly to reduce the computational requirements of the algorithm and, therefore, the time required for execution. First of all, a connecting path between N_1 and the selected node N_2 is

searched if and only if (1) is verified:

$$\|N_1 - N_2\| < c_{\text{best}} - c_{\text{subpath2}} \quad (1)$$

where c_{best} and c_{subpath2} are the costs (i.e. the path length) of the best solution found up to now and of the subpath of the selected path from N_2 to the goal, respectively. Until a first solution is found, c_{best} is equal to c_{subpath1} , the cost of the subpath of the current path from N_1 to the goal (referring to Figure 2). According to informed sampling [22], if (1) is verified, a connecting path between N_1 and N_2 is searched inside the following ellipsoid:

$$X_{\text{ell}} = \{x \in X_{\text{free}} \mid \|x - N_1\| + \|x - N_2\| < c_{\text{best}}\} \quad (2)$$

where x is a generic configuration of the robot and X_{free} the set of all collision-free configurations.

2) *informedOnlineReplanning* algorithm: This algorithm manages the whole re-planning procedure by calling several times *pathSwitch* and giving it the required inputs. In particular, it enriches the set of available paths adding the portion of the current path that lies beyond the obstacle that is obstructing it; furthermore, it defines the set of free nodes from which *pathSwitch* will be called. This set comprises all the free nodes of the current path between the robot current configuration and the obstacle. The idea is to start calling *pathSwitch* from the farthest node from the current configuration to have enough time to find at least a first feasible solution before traveling through that node. Every time *pathSwitch* is called from a node, the cost of the best available solution found up to now is used in (1). The algorithm considers the nodes of the current solution located before the obstacle. When the replanner finds a new solution, the new nodes are added to the searching set to conduct more profound research towards better and better solutions. The algorithm ends when all the nodes have been analyzed or when the computing time exceeds the maximum allowed time. In the *path re-planning* thread, *informedOnlineReplanning* is called at each cycle, to continuously optimize the current path in an anytime fashion. The maximum allowed time may vary. When an obstacle blocks the current path, a small amount of time should be given to the algorithm, in

TABLE I: Speed scaling module parameters.

T_r (s)	a_s (m/s ²)	S_p (m)	C (m)	v_h (m/s)
0.2	0.1	0.6	0.3	measured

order to find quickly a feasible solution; in the next iterations, when the aim is to optimize the current solution, a bigger amount of time can be used by the re-planner. Note that the responsiveness of the replanner is critical to not interrupting the task execution waiting for the replanner output.

B. Reactive speed modulation

Although the main focus of the paper is the replanning approach, a speed modulation strategy is implemented to meet ISO/TS 15066. In this way, the robot is able to avoid collisions even while the replanning is computing the new path. Safety standard defines a relation between the robot velocity and the distance between the manipulator and the human. This relation is an inequality to ensure that the human-robot distance $S(t_0)$ is large enough for the robot to stop without colliding with the human. To this purpose, SSM defines the *protective distance* S_p and requires that

$$S_p = v_h \left(T_r + \frac{v_r}{a_s} \right) + v_r T_r + \frac{v_r^2}{2a_s} + C \leq S(t_0) \quad (3)$$

where v_r is the robot velocity towards the human, v_h is the human velocity towards the robot, a_s is the maximum Cartesian deceleration of the robot towards the human, T_r is the reaction time of the robot, and C is a parameter accounting for the uncertainty of the perception system. Under the assumption of constant Cartesian deceleration, it is possible to derive a maximum robot velocity (towards the human) v_{max} , given the current human-robot relative distance and the human speed:

$$v_{max} = \sqrt{v_h^2 + (a_s T_r)^2 - 2a_s(C - S(t_0))} - a_s T_r - v_h \quad (4)$$

In order for the robot to respect the safety specifications given in [17], its speed should be monitored and, at each time instant, limited to v_{max} .

To do so, we use a trajectory scaling algorithm that calculates the robot speed override in real-time [2]. The trajectory scaling algorithm receives as input the robot nominal velocity \dot{q}_{nom} and the maximum allowed human-robot relative velocity v_{max} and computes the robot speed scaling factor by solving an optimal control problem:

$$\underset{\dot{q}, s}{\text{minimize}} \|\dot{q} - s \dot{q}_{nom}\|^2 \quad (5)$$

subject to $v_r \leq v_{max}$. Details on the algorithm to solve (5) can be found in [23]. The robot speed scaling factor s is a scalar value that represents the slowdown of the robot due to the human proximity from 0 (robot stop) to 1 (nominal velocity).

IV. EXPERIMENTAL RESULTS

Experimental tests have been performed to evaluate the capabilities of the proposed strategy. The experiments have

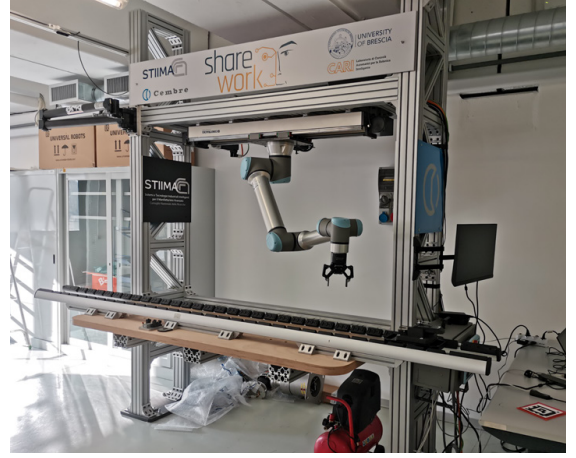


Fig. 3: Experimental setup used to test the proposed framework and to compare it with industrial common practice.

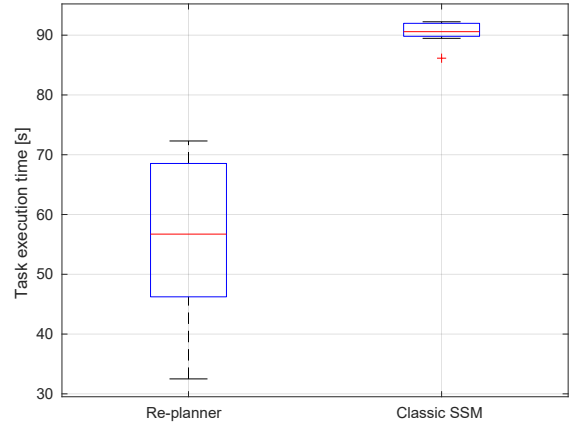


Fig. 4: Execution time of the desired movement. Left: proposed framework with path re-planning; right: three-fixed-area SSM.

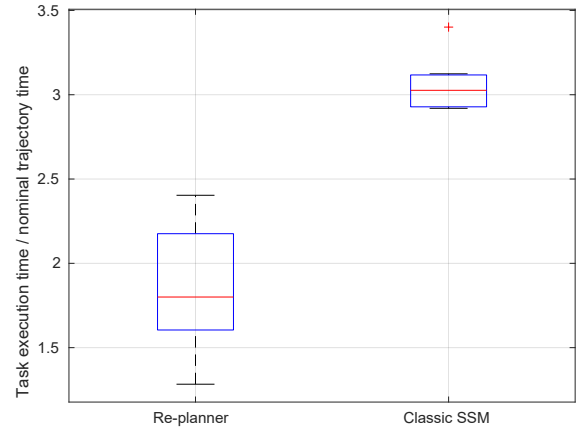


Fig. 5: Ratio of the actual execution time and the nominal execution time. Left: proposed framework with path re-planning; right: three-fixed-area SSM.

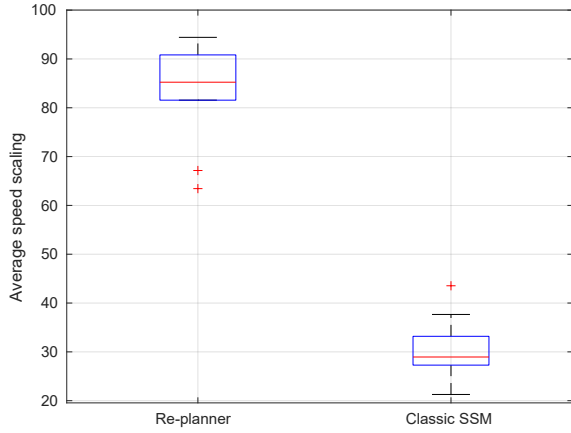


Fig. 6: Average speed scaling (the higher, the better). Left: proposed framework with path re-planning; right: three-fixed-area SSM.

been executed in the industrial cell of Figure 3, which is composed of a Universal Robots UR10e, a 6 degree-of-freedom collaborative robot, mounted upside down and working on a work-table in front of it.

The proposed method has been implemented in ROS (Robot Operating System) using *MoveIt!*. The offline planner uses RRT-Connect [19] to compute the set of paths and the less costly one (*i.e.*, the shortest path) is selected as the current path. The vision system uses an Intel Realsense D435 camera with a framerate of 30 Hz. The *speed modulation* block runs at 500 Hz while the frequency of the *path re-planner* block depends on the time given to *informe-dOnlineReplanning*: the maximum allowed time is equal to 100 ms when an obstacle blocks the current path and 150 ms when the aim is to optimize the current path. Table I summarizes the values of the terms of (4) used during the tests.

Ten tests were carried out to compare the proposed framework effectiveness with a standard three-fixed-areas SSM strategy (namely, the robot works at full speed, reduced speed, or stops if it is in a low, medium, or high danger zone) [24].

The tests were executed as follows. At the beginning of each test, a set of four paths is computed. The number of paths affects the performance of the replanner: the more the paths, the better the optimality of the solution could be but the greater the computational complexity. Furthermore, the more paths, the less readability of the approach as the human has to remember several possible paths. A trade off should be identified. Then, the robot moves from a selected start configuration to a selected goal configuration emulating a pick-and-place action, while an operator is doing some task on the workbench present in the cell. When the robot starts moving, the operator approaches the work table and stays for 60 s before leaving. Approaching the table, the operator obstructs the robot path, forcing it to reduce its speed and

possibly re-plan a new path. In particular, when the proposed framework is applied, it is expected that the robot reduces its speed and follows a new path that avoids the obstacle, so that the robot can rapidly reach the goal. Instead, during the tests of the three-fixed-areas SSM technique, the presence of the operator in the shared work-space causes a larger reduction of the robot speed or even a robot stopping.

We measure the trajectory execution time from start to goal and the average reduction of the robot speed along the path during ten repetitions of the test.

Figure 7 shows how the re-planner works. The green box represents the operator box occupancy, while the available paths are represented, for simplicity, as coloured lines which connect the waypoints but the robot travels some sort of arc between them. The yellow path of Figure 7a is the result of the optimization of the initial travelled path, the green one. Then, in Figure 7b, the human obstructs the robot's trajectory, so the re-planner finds, firstly, a sub-optimal free path and, then, optimizes it, obtaining the yellow one. Figures 4, 5, 6 show the results of the tests.

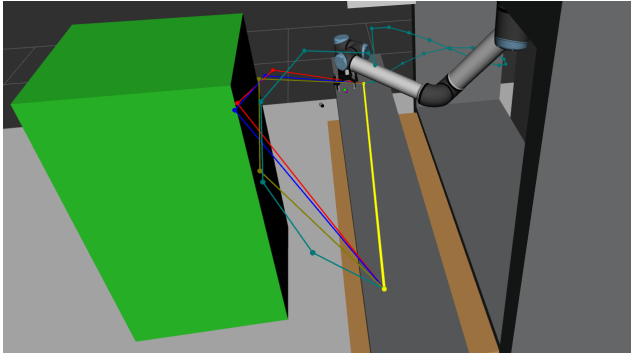
In Figure 4, the boxplots of the execution times needed to complete the task, when the re-planner was active and when the classic SSM was active, are compared. The average time for task execution in the first case is equal to 55.6 s, while in the second case it is equal to 90.5 s. Similarly, the ratio between the execution time and the nominal time to complete the trajectory (when there are no mobile obstacles inside the cell) is computed and shown in Figure 5 (the closer this ratio is to 1, the less the robot slowdown due to the presence of the human compared to the case without the human). This time difference can be well understood by looking at Figure 6. The speed scaling factor is represented as a value between 0 and 100, where 0 means that the robot is stationary while 100 that the robot is moving at the nominal speed, and the average value is calculated for each test. As shown in Figure 6, when the re-planner is running, the average speed scaling is larger than that obtained during the classic SSM technique tests. Thanks to path re-planning, the robot always moves along a free path that keeps it at a greater distance from the operator, and, consequently, it can move at higher speeds.

A video of the experiments is also available in the material attached to this paper.

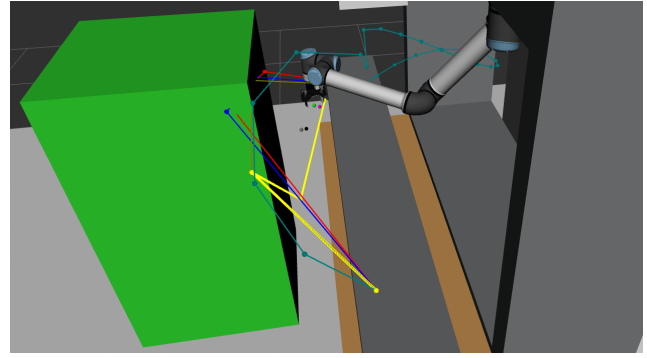
V. CONCLUSIONS

We have proposed a re-planning framework for Human-robot collaboration applications to cope with robots and humans simultaneous presence inside a robotic cell. Industrial typical applications use a simple three-fixed-areas speed monitoring and separation technique, which guarantees operator safety but provides an inadequate and inefficient collaboration. The proposed framework exploits an advanced SSM that combines a speed-scaling technique with a re-planning algorithm to make the robot more reactive to human movements.

Tests on an industrial cell have been executed to compare the proposed framework with a three-fixed-areas SSM approach, typically adopted in industries. Results show that



(a) The robot follows the yellow path, which is the result of the optimization of the initial followed path, the green one.



(b) The operator obstructs the robot current path, so the re-planner find a new free path, coloured in yellow.

Fig. 7: Example of how the re-planner works in the industrial cell used during the tests. The green box represents the operator occupancy box. The available paths are represented as coloured lines which connect waypoints, for simplicity, but the robot travels some sort of arc between the waypoints.

the proposed framework has outperformed industry most common practices.

Results show that the proposed framework makes the robot able to complete its task despite the presence of the human and requiring much less time than the three-fixed-areas SSM strategy. In general, the differences between the performance of the two methods depend on the scenario, namely the human and robot tasks, and a broader analysis should therefore be conducted to obtain more general results. Nonetheless, the results draw the conclusion that online re-planning combined with speed modulation can significantly enhance efficiency of HRC industrial applications.

REFERENCES

- [1] M. Koppenborg, P. Nickel, B. Naber, A. Lungfiel, and M. Huelke, "Effects of movement speed and predictability in human-robot collaboration," *Human Factors and Ergonomics In Manufacturing*, vol. 27, no. 4, pp. 197–209, 2017.
- [2] M. Faroni, M. Beschi, and N. Pedrocchi, "An MPC framework for online motion planning in human-robot collaborative tasks," in *Proceedings of the IEEE Int. Conf. on Emerging Tech. and Factory Automation*, Zaragoza (Spain), 2019.
- [3] E. Magrini, F. Ferraguti, A. Ronga, F. Pini, A. Luca, and F. Leali, "Human-robot coexistence and interaction in open industrial cells," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101846, 2020.
- [4] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human-robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.
- [5] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 882–893, 2016.
- [6] M. Faroni, R. Pagani, and G. Legnani, "Real-time trajectory scaling for robot manipulators," in *Proceedings of the International Conference on Ubiquitous Robots*, Kyoto (Japan), 2020.
- [7] S. LaValle, "Rapidly-exploring random trees: a new tool for path planning," *The annual research report*, 1998.
- [8] D. Ferguson, N. Kalra, and A. Stentz, "Replanning with RRTs," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 1243–1248.
- [9] D. Connell and H. La, "Dynamic path planning and replanning for mobile robots using RRT," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2017, pp. 1429–1434.
- [10] M. Zucker, J. Kuffner, and M. Branicky, "Multipartite RRTs for rapid replanning in dynamic environments," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1603–1609, 2007.
- [11] M. Otte and E. Frazzoli, "RRTx: Real-time motion planning/replanning for environments with unpredictable obstacles," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 797–822, 2016.
- [12] Z. Zhang, B. Qiao, W. Zhao, and X. Chen, "A Predictive Path Planning Algorithm for Mobile Robot in Dynamic Environments Based on Rapidly Exploring Random Tree," *Arabian Journal for Science and Engineering*, 2021.
- [13] J. van den Berg, D. Ferguson, and J. Kuffner, "Anytime path planning and replanning in dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 2366–2371.
- [14] Y. Chen, Z. He, and S. Li, "Horizon-based lazy optimal RRT for fast, efficient replanning in dynamic environment," *Autonomous Robots*, vol. 43, no. 8, pp. 2271–2292, 2019.
- [15] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *International Journal of Robotic Research*, vol. 21, pp. 1031–1052, 2002.
- [16] C. Tonola, M. Faroni, N. Pedrocchi, and M. Beschi, "Anytime informed path re-planning and optimization for robots in changing environments," *arXiv e-prints*, p. arXiv:2103.13245, Mar. 2021.
- [17] "ISO/TS 15066:2016 Robots and robotic devices – Collaborative robots," International Organization for Standardization, Geneva, CH, Standard, 2016.
- [18] Q.-C. Pham, "A general, fast, and robust implementation of the time-optimal path parameterization algorithm," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1533–1540, 2014.
- [19] J. J. Kuffner and S. M. LaValle, "RRT-Connect: An efficient approach to single-query path planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, San Francisco (USA), 2000, pp. 995–1001.
- [20] E. A. Sisbot and R. Alami, "A human-aware manipulation planner," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1045–1057, 2012.
- [21] S. Tarbouriech and W. Suleiman, "Bi-objective motion planning approach for safe motions: Application to a collaborative robot," *Journal of Intelligent and Robotic Systems*, vol. 99, pp. 45–63, 2020.
- [22] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Informed sampling for asymptotically optimal path planning," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 966–984, 2018.
- [23] M. Faroni, M. Beschi, C. Guarino Lo Bianco, and A. Visioli, "Predictive joint trajectory scaling for manipulators with kinodynamic constraints," *Control Engineering Practice*, vol. 95, p. 104264, 2020.
- [24] B. Matthias and T. Reisinger, "Example application of iso/ts 15066 to a collaborative assembly scenario," in *Int. Symp. on Robotics*. VDE, 2016, pp. 1–5.