Online supervised global path planning for AMRs with human-obstacle avoidance

Marina Indri, Fiorella Sibona,
Pangcheng David Cen Cheng
Dipartimento di Elettronica e Telecomunicazioni
Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
{marina.indri, fiorella.sibona, pangcheng.cencheng}@polito.it

Corrado Possieri
Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti"
Consiglio Nazionale delle Ricerche
(IASI-CNR)
Via dei Taurini 19, 00185 Roma, Italy
corrado.possieri@iasi.cnr.it

***Abstract*—In smart factories, the performance of the production lines is improved thanks to the wide application of mobile robots. In workspaces where human operators and mobile robots coexist, safety is a fundamental factor to be considered. In this context, the motion planning of Autonomous Mobile Robots is a challenging task, since it must take into account the human factor. In this paper, an implementation of a three-level online path planning is proposed, in which a set of waypoints belonging to a safe path is computed by a supervisory planner. Depending on the nature of the detected obstacles during the robot motion, the re-computation of the safe path may be enabled, after the collision avoidance action provided by the local planner is initiated. Particular attention is devoted to the detection and avoidance of human operators. The supervisory planner is triggered as the detected human gets sufficiently close to the mobile robot, allowing it to follow a new safe virtual path while conservatively circumnavigating the operator. The proposed algorithm has been experimentally validated in a laboratory environment emulating industrial scenarios.**

***Index Terms*—Mobile robots, online path planning, human obstacle avoidance.**

I. Introduction and state of the art

The application of mobile robots in industry is growing in recent years, since they increase flexibility and enhance the performance of the production line. Those mobile platforms that follow a fixed path while performing repetitive tasks are known as Automated Guided Vehicles (AGVs), while those that move autonomously within the industrial environment are called Autonomous Mobile Robots (AMRs).

In the scenario envisaged for the smart factories, the mobile agents work in a space shared with human operators. In this context, the use of AMRs is preferred, since they are able to process the information coming from their surroundings, thanks to the on-board intelligent sensory system. However, the mobile robot motion planning is not an easy task. Path planning is a well-known problem for finding feasible collision-free paths that allow the robot to reach a desired destination. The complexity of the problem may vary depending on the knowledge about the environment, as well as the presence of dynamic obstacles, such as human operators in the industrial workspaces.

Therefore, many researchers have developed several algorithms for robot navigation. For example, there are navigation algorithms based on probabilistic methods, such as the Rapidly-exploring Random Tree (RRT) [1], RRT* [2], and the Probability Roadmap (PRM) [3], that randomly explore the free spaces on the map to find a feasible path that connects the initial and final poses. Despite several improvements of such methods and their wide use thanks to their simplicity, they do not guarantee the optimality of the solutions [4].

Other path planning algorithms are based on heuristic-search methods, such as Genetic Algorithms (GAs) [5], Dijkstra [6], A* [7] and D* [8]. The GAs are inspired by the Darwinian evolution, by selecting the fittest sub-optimal solutions for reproduction, so to produce offsprings of the next generation. On the other hand, A* is an improvement of the Dijkstra algorithm to find the shortest path, whilst D* is an algorithm based on the A*, but with a dynamic cost behaviour.

In particular, the conventional A* algorithm has been applied in many path planning problems because of its simplicity and high efficiency in finding the optimal solution. Even though it does not provide a safe and smooth path, it is still widely used due to its versatility, since it can be customized and/or combined with other path-finding algorithms so that specific requirements can be fulfilled. Therefore, many versions of the A* algorithm were proposed. In [9], a modified A* algorithm is presented, in which virtual obstacles are added in the environment in order to guarantee safety during the path planning process. The improved A* algorithm presented in [10] takes into account the safety and time cost in the objective function, so it can be applied also in complex terrain environments.

Another navigation method that is recently emerging is the Artificial Potential Field (APF), in which the robot is attracted towards the desired destination while it is repelled from obstacles [11]. A Membrane Evolutionary Artificial Potential Field (memEAPF) is proposed in [12]. The parameters for generating a feasible and safe path are obtained through a combination of three methods: APF, membrane computing and a genetic algorithm. In [13], the authors presented a navigation system for dynamic industrial cluttered environments that combines the information coming from a sensor network and an APF based navigation algorithm.

In industrial environments where the human operators and mobile agents share the working space, safety is a fundamental factor to be considered. Nevertheless, most of the navigation algorithms do not take into account the human factor, since it is a dynamic obstacle whose behaviour is difficult to predict, or consider the human operators as generic dynamic obstacles, without any specific distinction [14].

Instead, there are some safety concepts presented in [15], that consider the usage of mobile robots in different scenarios. Furthermore, there are currently no standards for safe navigation in industry for AMRs, although some guidelines are recently being developed (R15.08 Drafting Subcommittee presentation from the Autonomous Mobile Robot Conference, September 2019 [16]). The current working scenario synergistically involves human operators and fixed-base or mobile robots, making the latter ones part of the production line. This makes new questions arise about the operators' willingness to work closely with unpredictable machines (as usually the AMRs are) [17].

This paper refers to an improvement of the Supervisory Global Planner (SGP) architecture introduced in [18] with the integration of an online update of the computed path, and the human detection capability presented in [19], which allows for a more conservative local behaviour around human obstacles. In the case of static scenarios, the position of the obstacles in the environment are well known, so the generated path can be considered as a safe path. However, when the robot deviates due to the presence of a human operator, the local planner reacts to that obstacle, and the SGP is triggered starting from the current robot pose.

The key point is given by the definition of predefined safe paths, to be followed by the mobile agents each time it is possible, in order to minimize the risk of unexpected obstacles along their motion (similarly to the fixed paths followed by traditional AGVs). This is enriched by the ability of simultaneously handling the presence of dynamic obstacles, guaranteeing a safer avoidance policy in case of humans and the return to the safe path as soon as possible. The resulting behaviour is especially suitable for industrial scenarios, in which a proper level of autonomy must be left to the mobile agents to really exploit them in the implementation of flexible production lines, while assuring at the same time safety conditions adequate for the presence of human operators.

The path planning algorithm run by the supervisory planner has a deterministic and repetitive behavior (i.e., given the initial and final poses, the path will always be the same). It is then considered as safe not only because it automatically avoids all the known static obstacles, but also because it provides a fixed path followed by the mobile robots, if no variation occurs in the environment along such a path. This way, the human operator who is working cooperatively with mobile agents is aware about the trajectory of the robot, and can possibly avoid intersecting the robot safe path. Nevertheless, a human worker could unintentionally cross the dedicated pre-defined route: the proposed algorithm allows to conservatively avoid obstacles identified as humans even though the latter ones are not expected.

The combination of the mentioned features ensures an overall safe behaviour, since the robot (i) follows a safe virtual path, (ii) has the ability of re-planning when a human obstacle is close and (iii) can adapt its behaviour to the environment's changes.

The paper is organized as follows: Section II presents the proposed procedure, at first providing some details about the online supervisory planner, then describing the algorithm implementation. The testing of the procedure is unfolded in Section III, where the results obtained from different test scenarios are reported. Finally, Section IV provides the conclusions and some open issues.

II. Online Supervisory algorithm for safe path traveling with human obstacle avoidance

In this section, a description of the developed online path planning algorithm is provided.

In general, the mobile robot path planning is performed at two levels: Global Planning and Local Planning. Given a starting point and the destination coordinates, the Global Planner (GP) computes offline the path that only considers the static obstacles in the real world environment, while the Local Planner (LP) updates the computed path when the sensor system detects the presence of an unexpected obstacle, allowing the robot to avoid any potential collision while moving.

The introduction of the SGP as the highest level within this planning hierarchy allows the robot to follow a set of deterministic waypoints, along a safe virtual path. The SGP algorithm is based on the collision-free motion planning presented in [20], that generates a path that tends to an algebraic curve. In particular, the SGP takes into account the kinematic model of the unicycle robot, and the safe curve is obtained letting the intersection of the obstacle space and a set of polynomial functions describing the possible trajectories be empty. Note that the equations describing this curve will be omitted here for brevity, since they were already provided and experimentally validated in [18]. However, in the latter, the SGP computed the waypoints offline: whenever the LP deviated the robot due to the presence of unknown obstacles, it was not ensured to resume the motion along the SGP computed path. In fact, the modified A* algorithm is intrinsically attracted by the imposed safe curve when the robot is sufficiently near to it, while only the pure A* mechanism is kept otherwise. Moreover, obstacles were detected but not identified, i.e, a human obstacle was treated in the same way as a generic object.

In order to obtain a safer behaviour, in terms of human obstacle detection and avoidance, the features of smart AMRs presented in [19] are added now, so the human operator can be identified and published as a virtual obstacle, whose inflation radius is larger than the one around generic obstacles. In terms of safe path conservation, an event-based trigger is added to the SGP. This planner will be referred in the next sections as the Online Supervisory Global Planner (OSGP). The hierarchical planning structure is reported in Figure 1.

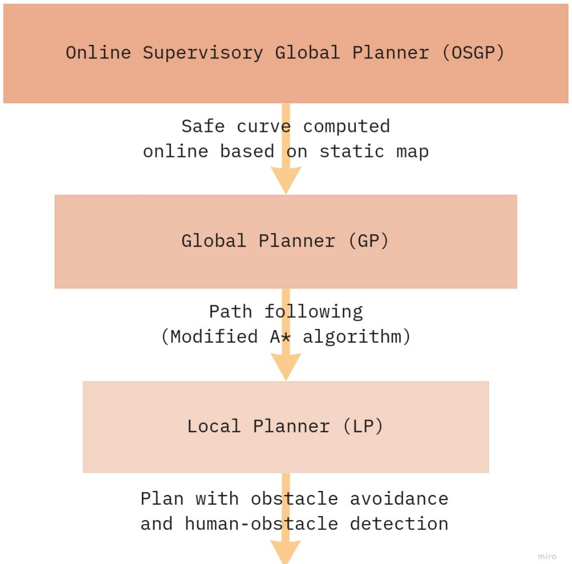

Fig. 1. Hierarchical structure of the online supervised global planning.

A. *OSGP algorithm implementation*

The code for computing the SGP path is implemented in MATLAB and provides as output a set of waypoints from a starting pose towards a final goal. The MATLAB function also registers as a ROS (Robot Operating System) [21] node, in order to convey the planning information onto the ROS Parameter Server. The communication with the robot ROS network is performed through the MATLAB ROS Toolbox, that allows the local machine to compute and send the waypoints, and read the status of the robot in order to trigger the re-planning behaviour.

The waypoints values are suitably namespaced to ease potential multi-robot implementations, and made available to the ROS system nodes managing the autonomous mobile robot navigation and vision-enhanced obstacle avoidance. It is worth noting that the whole MATLAB part is run in *headless mode*, i.e., the code is run from the command line with specific options that suppress the display server, the splash screen display, and the desktop version modules, which significantly reduces the CPU usage.

Following a system of event-based triggers, the SGP algorithm is re-computed only when strictly necessary, to ensure a deterministic and time-efficient behaviour. As previously mentioned, the path computed by the OSGP can be considered as safe, since it is based on a binary occupancy grid map in which static obstacles are conservatively enclosed by ellipses of minimum radius. The GP implements a modified A* algorithm where waypoints passed by the OSGP are favoured in terms of cost, for the heuristic global plan computation. As a human obstacle is detected and sufficiently close to the AMR, the LP deviates the robot and a trigger is sent to the OSGP for path re-computation. In order to let the robot trigger the OSGP planning while already overcoming the human obstacle, a small delay is introduced. The current pose of the mobile platform is taken into account and used as a starting pose for the collision-free motion planning algorithm.

The dynamic obstacles are avoided thanks to the LP and, in particular, human obstacles are identified using YOLO (You Only Look Once) [22]: the C++ YOLO code has been modified filtering the information about the bounding boxes of the identified objects in order to consider only pixels labelled as "person". These data are then written to a text file, which is fed to a ROS topic. The identified humans relative distances are subsequently computed, exploiting camera-laser data sensor fusion. The humans' positions on the map are then published as virtual obstacles enclosed in virtual cages: to influence the local planner costmap, humans are assigned a greater inflation radius value than other obstacles. This prevents the AMR from traveling too close to the operator when trying to overcome it. For further details about this behaviour, refer to [19]. The overall algorithm flow is showed in Figure 2.

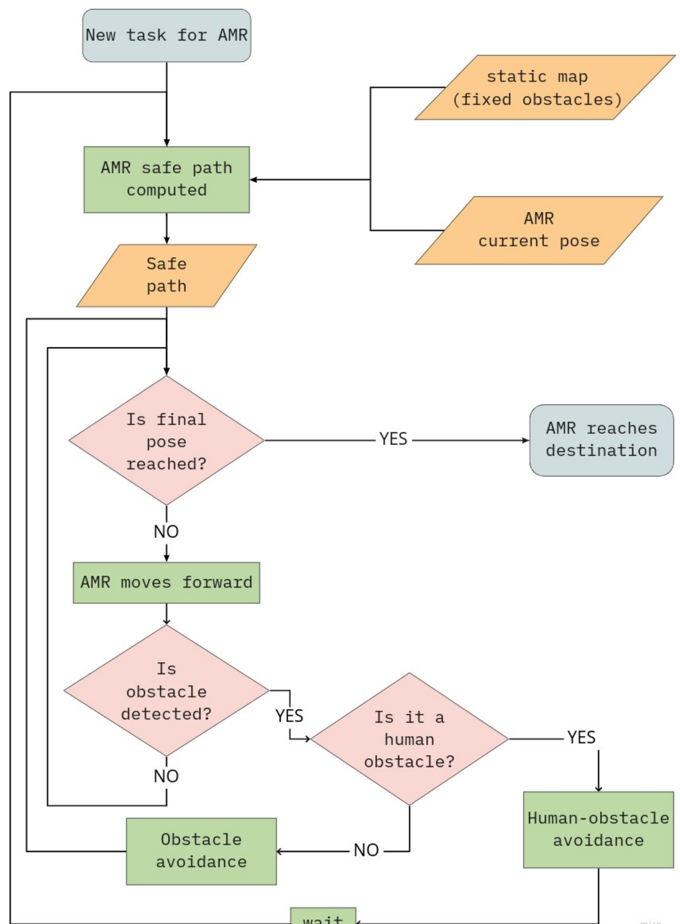

Fig. 2. Flowchart for the OSGP algorithm.

To better highlight the core difference from the SGP algorithm proposed in [18], the expected behaviour is represented in Figure 3. When using the offline SGP version, a significant deviation from the supervisory path could result in the activation of the pure A* mechanism, which computes the shortest path (grey dotted line) to reach the final goal. However, this is undesirable, since it is not compliant with the safety requirements, due to the fact that there may be other human operators outside the safe path. To make up for this unwanted behaviour, the SGP safe virtual path is re-planned online, taking as starting pose the current one (blue solid line).

Moreover, the proposed algorithm presents some implementation measures that can improve the technology criticality level [23] and can be considered as catalyst for an eased transferability to real industrial contexts.

- *Containerization as portability facilitator*. The OSGP algorithm with human obstacle detection and avoidance has been containerized using Docker [24]. The elements which can be considered robot-agnostic have been grouped based on their main task, e.g., AMR vision, AMR navigation, and SGP computations. Therefore, to ease up the migration to other hardware specifications and software environments, all the robot-specific elements and tuning parameters have been grouped to enable suitable edits. Using containers allows to speed up the transfer process from laboratory demonstrators to commercially available AMRs. Indeed, the Linux containers technology is considered a lightweight alternative to virtual machines [25] and allows the end-user to run the containerized application having the installation of Docker on the target machine as only constraint.
- *Cost-efficient improvements as technology transfer enablers*. The use of cost-efficient sensors enhanced by deep learning algorithms can be considered a key enabler for technology transfer in contexts that would, as a matter of principle, not consider it. By exploiting sensors which may not fall within the classification of high-end and/or high tech devices, the focus is moved from the economic value toward the innovation meaning of a resulting solution. Upgrading obsolete equipment can foster the adoption of new technologies avoiding to exacerbate low technology transfer rates that may affect Small and Medium Enterprises [26].

III. Experimental results

In order to demonstrate the validity of the OSGP algorithm in an environment similar to an industrial context, the emulated scenario involved a mobile robot moving in a closed space shared with human operators, like warehouse corridors with racks or assembly workstations with conveyors. The working

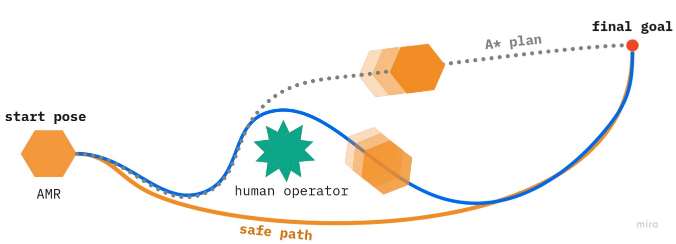

Fig. 3. SGP behaviour versus OSGP behaviour.

space used for testing is shown in Figure 4, reporting the MATLAB occupancy grid map and OSGP path plot, and the corresponding GP path plan visualization on ROS *rviz*.

To validate the proposed algorithm, we employed a Pioneer 3DX mobile robot equipped with a SICK LMS200 laser range finder with 10-meter range and scanning angle of 180°. The video stream from an entry level IP camera served as vision source, fed to the YOLO real-time object detection system. As a processing unit we used a Raspberry Pi 3 Model B mounting an ARM Cortex-A53 (x4 core) CPU (1.2 GHz) and 1-GB RAM; the core processes were performed on a desktop PC with a Intel Core i7-7700 CPU and a dedicated GTX1060/6GB GPU.

Notice that some open issues identified in the practical implementation of the offline SGP algorithm, have been solved. Among these, a smoother behaviour have been achieved by substituting the `TrajectoryPlannerROS` local planner with the `TebLocalPlannerROS`, based on Time Elastic Bands evaluation [27]. Also, the MATLAB code has been run with its GUI switched off, i.e., in *headless mode*, to get rid of the time necessary for the interactive program parts to load, since no interaction is needed during the algorithm execution. In particular, the supervisory planner function is run according to specific trigger flags which let MATLAB and the ROS system interact and automate the re-computation process. The overall high level setting representation is shown in Figure 5.

The execution of the online SGP algorithm in different test scenarios is showcased in the video footage available at [28].

A. Test Scenario 1: OSGP re-planning behaviour

In the first scenario (see Figure 6), we tested the re-planning behaviour of the OSGP when a human obstacle is identified. In this case, the person represents a human operator performing some operations in front of a workstation. As seen at 00:41, the AMR starts its motion along the safe path computed by the SGP and tracked by the GP plan (blue solid line) and when a person is detected, the LP (orange solid line) starts its standard obstacle avoidance mechanism.

As the human gets closer, the OSGP is eventually triggered (00:58) and a new reference path is provided to the GP. In this way, the mobile robot is always brought back towards the safe path even if its motion is significantly deviated. Moreover, it can be seen that the robot overcomes a person maintaining a conservative distance imposed by the virtual obstacle publication.

B. Test Scenario 2: human detection without re-planning

At 01:15 of the video, it is possible to appreciate the case in which a operator performing some quick checks along a rack is detected but, being far enough from the current AMR pose, the OSGP path re-computation is not triggered. Indeed, in this case the human presence does not disturb the mobile robot activities and a re-planning would introduce unnecessary overhead. Note that the operator pose is published as a virtual obstacle in the map (red dots). A screenshot of the performed test is provided in Figure 7.

C. Test Scenario 3: generic obstacle avoidance

The last test scenario, as shown at 01:37 of the video, considers the case in which a generic obstacle (not present in the static map) is encountered: the robot successfully avoids it without triggering a new safe path computation. This is desired to avoid further computations when it is not needed: the absence of a human operator in the robot neighbourhood does not impose a reactive resume of the safe virtual path. The third scenario OSGP behaviour can be seen in Figure 8.

IV. CONCLUSIONS AND FUTURE WORKS

In this paper we presented a supervisory planner algorithm that generates an online-updated safe path. In absence of changes in the environment, the motion of the AMR follows the safe virtual path that is provided by the OSGP. This feature can potentially increase the workers' confidence in sharing the workspace with mobile robots, since their overall motion can be predicted. Besides, using YOLO, the AMR can distinguish if an obstacle is a generic object or a human operator, and gather its distance through camera-laser data fusion. By publishing the human obstacle as a virtual one within the navigation map, its increased inflation radius allows for a conservative avoidance.

A possible further enhancement could be given by the creation of a specific dataset to train the YOLO neural network, with the aim of making it recognise peculiar pieces of equipment (not included in common datasets, e.g., the COCO dataset). These may be machines whose surroundings should be avoided due to particular conditions, e.g., high temperatures or radiations, implying that a close approach during motion could compromise the mobile platforms and/or the machinery itself. Clearly, the extension of the object recognition capabilities would depend on the plant configuration and on the client specific needs.

The tracking of a reference safe virtual path is always ensured, since the re-planning mechanism of the OSGP is triggered when a human is close enough to the robot in motion. In addition, thanks to the ROS namespace feature it is possible to extend the algorithm to the multi-agent case, while containerization using Docker fosters the technological transfer towards industrial processes by facilitating portability.

Even though some of the implementation choices can ease the criticality level of the proposed algorithm, being the range of AMRs available on the market very wide, some custom tuning for integration would be necessary. For sure some test scenarios on the field would be more indicative of the usability in a real industrial context.

As of now, the OSGP re-planning mechanism is triggered after a small delay. As a future improvement to make the system more efficient, re-planning while overcoming the human obstacle could be guaranteed by choosing a different type of trigger, to make sure that the robot has indeed left the original safe virtual path. For example, the trigger signal could be set if the distance from the initially imposed path is higher than a threshold, so to reduce the number of trigger events. In fact, re-triggering after an imposed waiting time does not ensure that the robot has moved from its current pose, maybe delayed by some external disturbance or internal computations.

It should be noted that the online computation of the SGP could be fully implemented in ROS, e.g., with Python, using

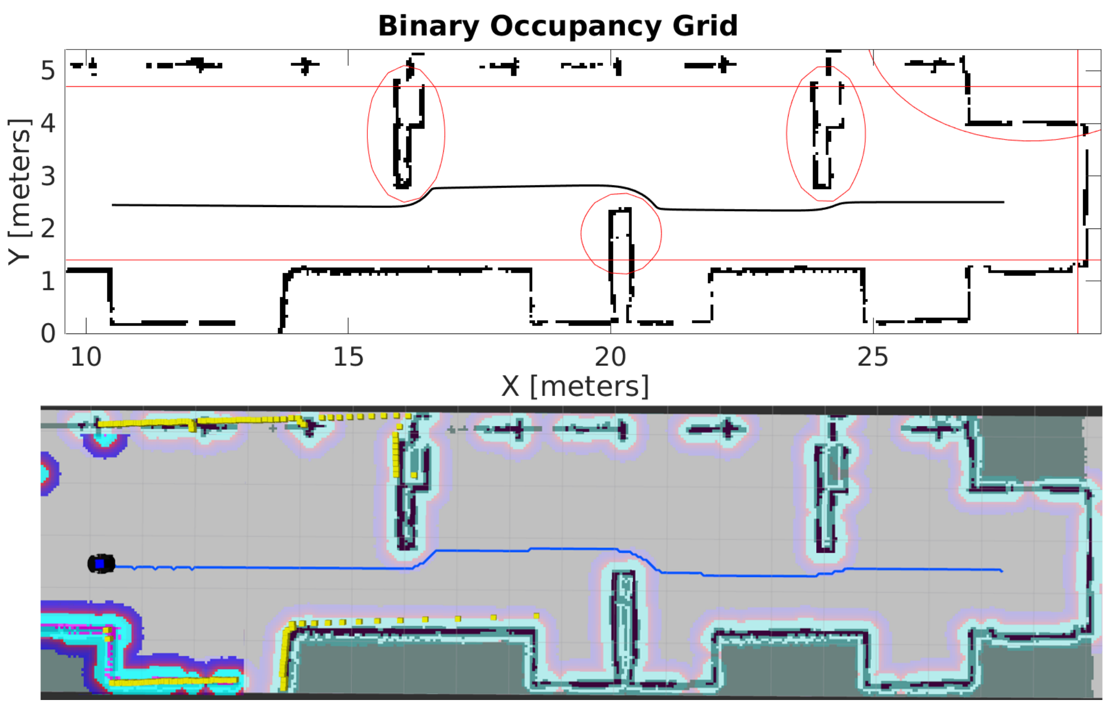

Fig. 4. Top: MATLAB OSGP path plot. Bottom: *rviz* visualization of the static map used for SLAM navigation.

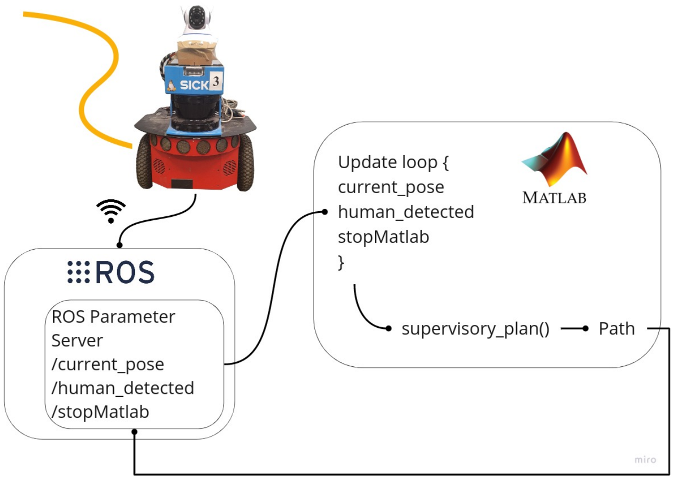

Fig. 5. High level algorithm implementation schema.

a proper differential equation solver. However, MATLAB was chosen since its solvers provide good enough results and can be easily integrated with other systems.

Future work could deal with the extension of the proposed framework to the multi-robot case. This problem is particularly challenging since, in this case, the OSGP also has to account for possible collisions among the robots, thus making the path planning more complex.

References

[1] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[2] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[3] L. Kavraki and J.-C. Latombe, "Randomized preprocessing of configuration for fast path planning," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE, 1994, pp. 2138–2145.

[4] Z. Tahir, A. H. Qureshi, Y. Ayaz, and R. Nawaz, "Potentially guided bidirectionalized rrt* for fast optimal path planning in cluttered environments," *Robotics and Autonomous Systems*, vol. 108, pp. 13–27, 2018.

[5] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," *Procedia Computer Science*, vol. 127, no. C, pp. 180–189, 2018.

[6] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[7] J. Yao, C. Lin, X. Xie, A. J. Wang, and C.-C. Hung, "Path planning for virtual human motion using improved A* star algorithm," in *2010 Seventh international conference on information technology: new generations*. IEEE, 2010, pp. 1154–1158.

[8] M. Korkmaz and A. Durdu, "Comparison of optimal path planning algorithms," in *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)*. IEEE, 2018, pp. 255–258.

[9] H. Mujtaba and G. Singh, "Safe Navigation of Mobile Robot Using A* Algorithm," *International Journal of Applied Engineering Research*, vol. 12, no. 16, pp. 5532–5539, 2017.

[10] H.-M. Zhang, M.-L. Li, and L. Yang, "Safe path planning of mobile robot based on improved A* algorithm in complex terrains," *Algorithms*, vol. 11, no. 4, p. 44, 2018.

[11] B. Patle, A. Pandey, D. Parhi, A. Jagadeesh *et al.*, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, 2019.

[12] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Applied Soft Computing*, vol. 77, pp. 236–251, 2019.

[13] H. Li and A. V. Savkin, "An algorithm for safe navigation of mobile robots by a sensor network in dynamic cluttered industrial environments," *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 65–82, 2018.

[14] M. Lutz, C. Verbeek, and C. Schlegel, "Towards a robot fleet for intralogistic tasks: Combining free robot navigation with multi-robot coordination at bottlenecks," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2016, pp. 1–4.

[15] M. Papa, D. Kaselautzke, K. Stuja, and W. Wölfel, "Different safety certifiable concepts for mobile robots in industrial environments." *Annals of DAAAM & Proceedings*, vol. 29, 2018.

[16] Preview of R15.08 Industrial Mobile Robot Safety, by Michael Gerstenberger (Chair of the R15.08 Drafting Subcommittee). Accessed: May 2020. [Online]. Available: https://www.robotics.org/userAssets/riaUploads/file/AMR19_Michael_Gerstenberger.pdf

[17] A. Markis, M. Papa, D. Kaselautzke, M. Rathmair, V. Sattinger, and M. Brandstttter, "Safety of mobile robot systems in industrial applications," 05 2019.

[18] M. Indri, C. Possieri, F. Sibona, P. D. C. Cheng, and V. D. Hoang, "Supervised global path planning for mobile robots with obstacle avoidance," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2019, pp. 601–608.

[19] M. Indri, F. Sibona, and P. D. C. Cheng, "Sensor data fusion for smart AMRs in human-shared industrial workspaces," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1. IEEE, 2019, pp. 738–743.

[20] C. Possieri and M. Sassano, "Motion Planning, Formation Control and Obstacle Avoidance for Multi-Agent Systems," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2018, pp. 879–884.

[21] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[22] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[23] Z. Juan, L. Wei, and P. Xiamei, "Research on technology transfer readiness level and its application in university technology innovation management," in *2010 International Conference on E-Business and E-Government*. IEEE, 2010, pp. 1904–1907.

[24] "Docker official website," accessed: June 2020. [Online]. Available: https://www.docker.com

[25] M. Chae, H. Lee, and K. Lee, "A performance comparison of linux containers and virtual machines using Docker and KVM," *Cluster Computing*, vol. 22, no. 1, pp. 1765–1775, 2019.

[26] L. Gumbi and H. Twinomurinzi, "SMME Readiness for Smart Manufacturing (4IR) Adoption: A Systematic Review," in *Conference on e-Business, e-Services and e-Society*. Springer, 2020, pp. 41–54.

[27] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.

[28] Online supervised global planning algorithm for AMRs with human obstacle avoidance. [Online]. Available: https://youtu.be/pph_DN_CTtw

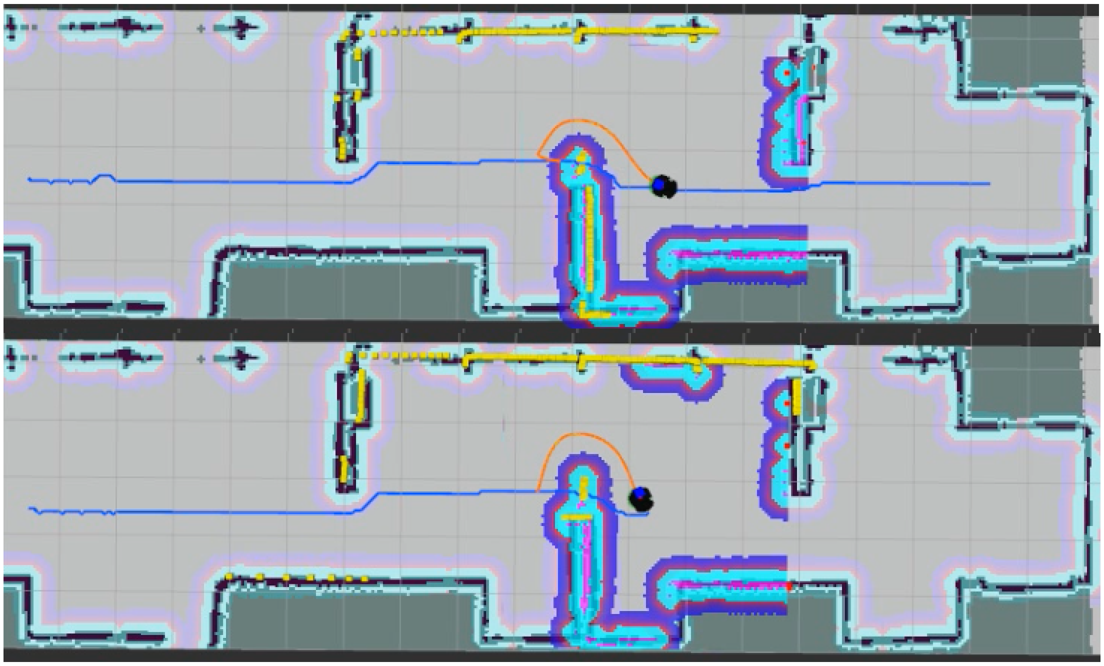

Fig. 6. Test scenario 1: *rviz* view of the OSGP re-planning behaviour after human detection.

Fig. 7. Test scenario 2: OSGP behaviour when a human obstacle is sufficiently far from the AMR.

Fig. 8. Test scenario 3: OSGP algorithm reaction to a generic obstacle detection.