Motion Planning for Mobile Robots using the Human Tracking Velocity Obstacles Method

Zoltán Gyenes[1] [a], Ilshat Mamaev[2] [b], Dongxu Yang[2], Emese Gincsainé Szádeczky-Kardoss[1] [c]
and Björn Hein[2] [d]

[1]Department of Control Engineering and Information Technology, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary
[2]Intelligent Process Automation and Robotics Lab, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Keywords: Mobile Robot, Motion Planning, Autonomous System, Human Tracking.

Abstract: Human-robot interaction is playing an increasingly important role in everyday life and we can expect an even bigger explosion in the use of robots in the future. One such use is where a mobile robot needs to follow the human. The main objective of this paper is to introduce a novel motion planning algorithm for mobile robots, which can be used to enable the robot to follow a human while maintaining a given distance. The motion planning algorithm has to take into account obstacles in the workspace of the robot at each sampling time and to generate a collision-free motion for the agent.

1 INTRODUCTION

Mobile robots are becoming more prominent in our daily lives. There are already commercially available street delivery robots (Valdez et al., 2021; Jung, 2020), wide range of applications for warehouse robots (Ch'ng et al., 2020), and service robots in healthcare industry (Reiser et al., 2009). Fueled by advances in artificial intelligence, the developments in autonomous mobile robotics with co-existence of humans and robots, have opened new challenges in motion planning and control. Collision avoidance and person following are the most basic tasks in mobile robotics, ensuring safety of the robotic application and comfortable human robot interaction.

In this paper, we propose an approach to mobile robot motion control for person-following with dynamic collision avoidance combining directive circle and velocity obstacles methods.

The rest of the paper is structured in the following way: Section 2 presents selected related work, considering the motion planning algorithms for mobile robots. In Section 3, the novel motion planning algorithm is introduced. After that, in Section 4, the simulation results are presented. Later on, in Section 5, the paper will be summarized.

[a] https://orcid.org/0000-0003-1803-7146
[b] https://orcid.org/0000-0003-2546-7304
[c] https://orcid.org/0000-0002-7789-6402
[d] https://orcid.org/0000-0001-9569-5201

2 RELATED WORK

In this Section, human tracking and motion planning algorithms for mobile robots will be presented.

2.1 Human Detection and Tracking

Object tracking is one of the classical tasks in computer vision. Due to the recent advances in AI-based object detection, tracking-by-detection has attracted interest in this field. The Simple Online and Realtime Tracking (SORT) was introduced by (Bewley et al., 2016) in 2016 and extended with deep association metric in 2017 (Wojke et al., 2017). The latter approach was tuned for pedestrian detection and outperformed other tracking algorithms (Xiang et al., 2015; Yang and Jia, 2016). Another approach is based on decomposition the person tracking process to several parts, using threshold segmentation and morphological operations for detection, Kalman filter algorithm for tracking and Hungarian algorithm for the data association algorithm (Qian et al., 2020). A tracking framework called EagerMOT fuses all available object observations originating from 2D and 3D detec-

Gyenes, Z., Mamaev, I., Yang, D., Szádeczky-Kardoss, E. and Hein, B.
Motion Planning for Mobile Robots using the Human Tracking Velocity Obstacles Method.
DOI: 10.5220/0011318200003271
In *Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2022)*, pages 484-491
ISBN: 978-989-758-585-2; ISSN: 2184-2809

tors via a two-stage association procedure. The first stage associates object detections from different sensor modalities, while in the second stage, a tracking formulation that allows to update track states is employed (Kim et al., 2021). Another sensor fusion approach (Ali et al., 2013) utilizes stereo camera to extract body and face features and uses LiDAR for extraction of legs features, combining these features to classify the target person. However, it only works when target faces towards the robot, which is a limitation for person following task. For multiple object tracking (MOT) a single-shot network for simultaneously detecting objects and extracting tracking features is applied to real-time calculation (Guo et al., 2020). Some models also learn to manage to track life circles in a data-driven approach (Chiu et al., 2021).

In an approach, a marker-based solution was introduced for the tracking task (Lenain et al., 2018a). At this method not only the magnetic but also the optical and acoustic and inertial tracking are described and also evaluated.

2.2 Motion Planning Methods

The Directive circle (DC) method designs a collision-free, near-optimal path for a mobile robot to chase another mobile robot by avoiding moving obstacles. Finding the shortest path to the another mobile robot is the first step at the algorithm. Then, the collision-free directions of the robot are computed using the velocity vectors and the Directive Circle (Masehian and Katebi, 2007). The collision-free path closest to the optimal path is then found. These subtasks are repeated one after the other until the two objects meet.

A novel potential field based motion planning method was introduced where a switching strategy can be used between the attractive potential of the target and a novel helicoidal potential field (Fedele et al., 2017). The helicoidal potential field helps to bypass an obstacle that has a motion around the agent.

The motion planning algorithms can be used also for Off-road mobile robots (Lenain et al., 2018b). A novel motion planning method was introduced for four wheel steering mobile robot that allows the motion capacity with respect to a normal car-like mobile robot, while the friction can be reduced.

The main concept of the *Velocity Obstacles algorithm (VO)* (Fiorini and Shiller, 1998) is to select a velocity vector that would result in a collision-free motion for the agent if the position and velocity information of the obstacles is known at every sampling time.

B_i defines the obstacles ($i = 1...m$ where m defines the actual number of obstacles that occur in the workspace of the agent) and the agent is A.

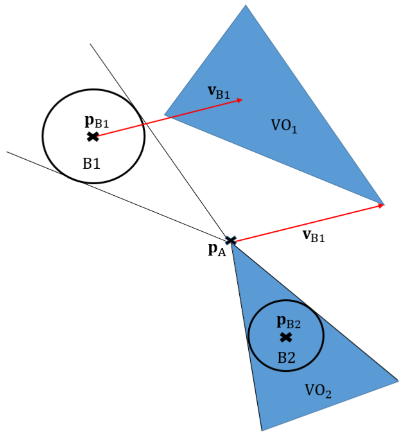

Figure 1: Velocity Obstacles method.

All velocity vectors of the agent can be calculated that would cause a collision between the obstacle (B_i) and robot (A) in a future time, this sets are the VO_i cones:

$$VO_i = \{\, \mathbf{v}_A \mid \exists\, t : \mathbf{p}_A + \mathbf{v}_A t \;\cap\; \mathbf{p}_{Bi} + \mathbf{v}_{Bi} t \neq 0 \} \quad (1)$$

where $\mathbf{v}_A$ and $\mathbf{v}_{Bi}$ are the velocity vectors and $\mathbf{p}_A$ and $\mathbf{p}_{Bi}$ are the positions of the agent and the obstacle. As an assumption, both the robot and the obstacles are disk-shaped and the velocities of the obstacles and the agent are unchanged until t.

The whole *VO* can be determined as the union of the different VO_i sets if there are more obstacles as:

$$VO = \cup_{i=1}^{m} VO_i \quad (2)$$

Figure 1 represents an example where a moving obstacle is in position $\mathbf{p}_{B1}$ and it has velocity $\mathbf{v}_{B1}$ at the current time. There is a static obstacle in the workspace with position $\mathbf{p}_{B2}$. The two *VO* areas are depicted with blue color.

After calculating the VO_i sets, the *Reachable Velocities (RV)* can be determined that consist every $\mathbf{v}_A$ velocity vector of the agent that is reachable considering the previously selected velocity vector. After the subtraction of the *VO* from the *RV* the *Reachable Avoidance Velocities (RAV)* can be received.

3 HUMAN TRACKING VELOCITY OBSTACLES METHOD

In this Section, the *Human Tracking Velocity Obstacles (HTVO)* motion planning algorithm is introduced where the *Directive Circle* and the *Velocity Obstacles* methods are combined and extended. Using this novel

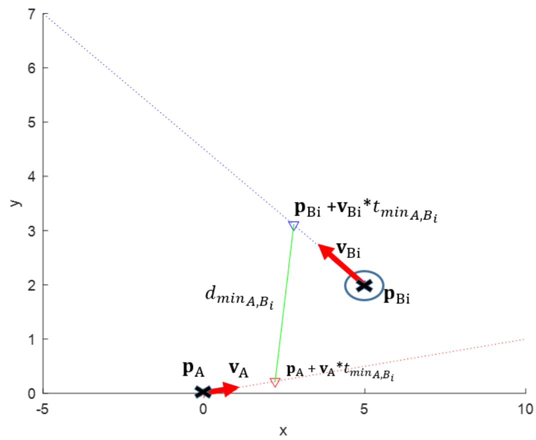

Figure 2: Precheck algorithm.

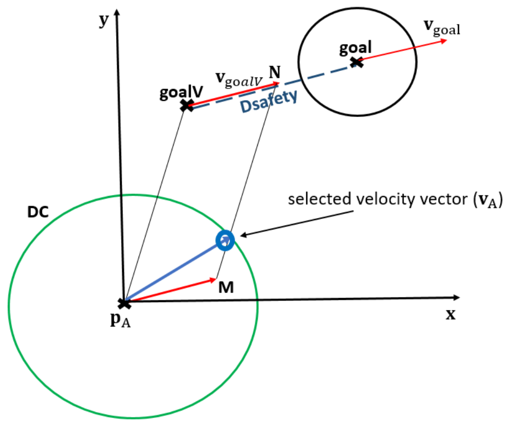

Figure 3: Virtual target and Directive Circle in HTVO.

method, the human tracking and obstacle avoidance aspects can be considered at the same time, generating a collision-free motion for the agent in the dynamic workspace.

3.1 Precheck Algorithm

At the beginning of the algorithm, it has to be selected, which obstacles play the less role in the motion of the agent so which VO_i could be eliminated from the workspace of the robot at the sampling time. This step could be called Precheck algorithm.

In Figure 2, the Precheck algorithm is shown.

Two different aspects can be considered while selecting the obstacle to be discarded:

- which obstacle will be reached in the longest time in the workspace of the robot
- at that time, how large will be the distance between the robot and the obstacle.

So for every obstacle, the minimum time and distance must be calculated when the agent and the obstacle are closest to each other during their motion.

$$t_{min_{A,Bi}} = \frac{-(\mathbf{p}_{\mathrm{A}} - \mathbf{p}_{\mathrm{Bi}})(\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{Bi}})}{||\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{Bi}}||}, \tag{3}$$

where $t_{min_{A,Bi}}$ presents the time interval when the robot and the obstacle will be nearest to each other. If the value of this parameter is a negative number, then it was in the past. $||.||$ represents the secondary norm.

Between the agent and the obstacle, the minimal distance can be calculated:

$$d_{min_{A,Bi}} = ||(\mathbf{p}_{\mathrm{A}} + \mathbf{v}_{\mathrm{A}} t_{min_{A,Bi}}) - (\mathbf{p}_{\mathrm{Bi}} + \mathbf{v}_{\mathrm{Bi}} t_{min_{A,Bi}})||, \tag{4}$$

So only those obstacles must be considered that fulfill the next equation:

$$\begin{aligned} 0 < t_{min_{A,Bi}} < 2 * T_{precheck} \quad \text{AND} \\ d_{min_{A,Bi}} < v_{\max} * T_{precheck} \end{aligned} \tag{5}$$

where $v_{\max}$ means the maximum velocity that the robot can reach and $T_{precheck}$ is a parameter of the algorithm that must be tuned. This parameter can be setted using apriori knowledge. Our experiments showed that if the value of the $T_{precheck}$ parameter is too small, it generates not a smooth path for the agent.

3.2 Velocity Selection for the Robot

For the human tracking and collision avoidance task, a novel motion planning algorithm, the *Human Tracking Velocity Obstacles* method can be used which is based on the combination of *Directive circle* and the *Velocity Obstacles*. Both the obstacles and the human are represented with circles. In that case, a virtual target must be introduced which is always behind the real target position (the human that the agent follows). The distance between the real and the virtual target position is always constant. In Figure 3, **goal** presents the real goal position, $\mathbf{v}_{\mathrm{goal}}$ is the velocity vector of the human and **goalV** is the virtual target position with $\mathbf{v}_{\mathrm{goalV}}$ velocity vector with an end point of N which is the same velocity vector as the human has ($\mathbf{v}_{\mathrm{goalV}} = \mathbf{v}_{\mathrm{goal}}$). The distance between the human and the virtual target position is the constant *Dsafety*. The robot is in position $\mathbf{p}_{\mathrm{A}}$. The DC circle is presented with green color which represents a circle with a radius of the maximum velocity v_{max} of the agent. M denotes the end point of the velocity $\mathbf{v}_{\mathrm{goalV}}$ if it starts from $\mathbf{p}_{\mathrm{A}}$. Where $M-N$ section intersects the DC circle, it will be the velocity vector of the robot which will generate a motion for the agent resulting in reaching the virtual target position. This velocity vector must be selected for the robot if it is in the *RAV* set, considering the *RV* and *VO* sets in the workspace.

A grid based velocity map is introduced to the *RV* set so if the desired velocity vector is not in the *RAV* set, then a grid-based velocity selection can be used.

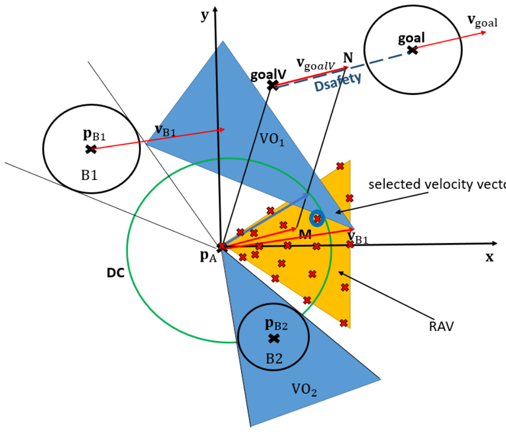

Figure 4: VOs for HTVO.

In Figure 3, the presented method can be seen.

There are several situations that can be considered in the velocity selection method (similarly to (Masehian and Katebi, 2007)):

- If there are no intersection point between the DC and the MN section then the human tracking task is not solvable because the robot cannot select a velocity vector that is large enough to reach the virtual target position

$$DC \cap MN = \emptyset \tag{6}$$

- If there is one intersection point between the DC and the MN section ($\mathbf{p}_1$), then it is the solution for the velocity selection task.

$$DC \cap MN = \mathbf{p}_1 \tag{7}$$

$$\mathbf{v}_\mathrm{A} = \mathbf{p}_1 \tag{8}$$

- If there are two intersection points between the DC and the MN section ($\mathbf{p}_1$ and $\mathbf{p}_2$), then that point must be selected which is closer to point N:

$$DC \cap MN = \{\mathbf{p}_1, \mathbf{p}_2\} \tag{9}$$

$$\mathbf{v}_\mathrm{A} = argmin_{\mathbf{p}_\mathrm{i}} ||\mathbf{N} - \mathbf{p}_\mathrm{i}|| \tag{10}$$

- If there are two intersection points between the DC and the MN section ($\mathbf{p}_1$ and $\mathbf{p}_2$), and N is situated between these two points, then N must be selected as the velocity vector of the agent. This situation can occur when the robot will reach the virtual target position in the next time step:

$$DC \cap MN = \{\mathbf{p}_1, \mathbf{p}_2\} \;\; AND \;\; \mathbf{N} \in \mathbf{p}_1\mathbf{p}_2 \tag{11}$$

$$\mathbf{v}_\mathrm{A} = \mathbf{N} \tag{12}$$

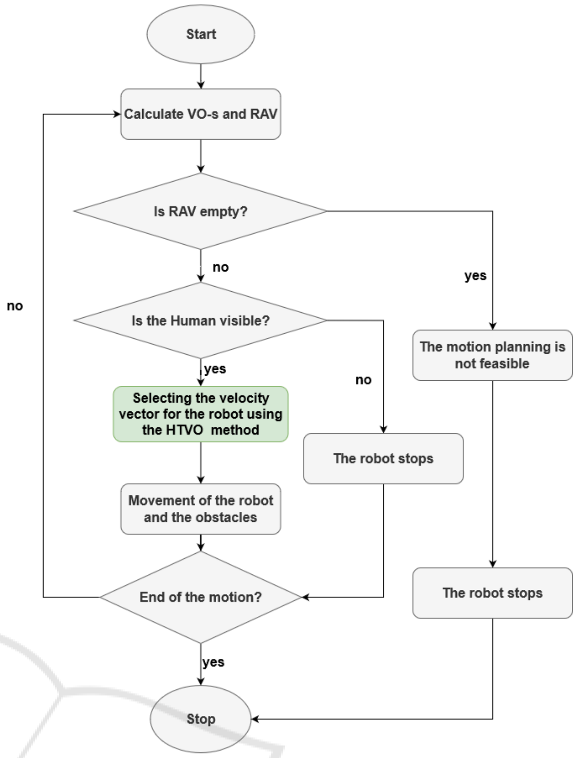

Figure 5: Steps of the HTVO algorithm.

- If the human stands in a position and it has no velocity vector, then the virtual goal is also a static point. In that case, the robot has to reach the virtual goal position as fast as it is possible. The velocity vector can be calculated as:

$$\mathbf{v}_\mathrm{A} = \frac{\mathbf{goalV} - \mathbf{p}_A}{||\mathbf{goalV} - \mathbf{p}_A||} \cdot min(v_{max}, ||\mathbf{goalV} - \mathbf{p}_A||), \tag{13}$$

- If the desired velocity vector that would cause an appropriate virtual target reaching is not in the RAV set, then the nearest velocity vector must be selected from the introduced grid.

$$DC \cap MN \notin RAV \tag{14}$$

$$\mathbf{v}_\mathrm{A} = argmin_{\mathbf{v}_{\mathrm{Grid(i)}}} ||(DC \cap MN) - \mathbf{v}_{\mathrm{Grid(i)}}||, \tag{15}$$

where every grid point must be investigated.

In Figure 4, a situation is represented where the velocity vector which would result the best human tracking solution is not reachable, so the nearest grid point will be selected for the agent from the RAV set. The grid points are the red x-s and the RAV set is presented with the yellow area.

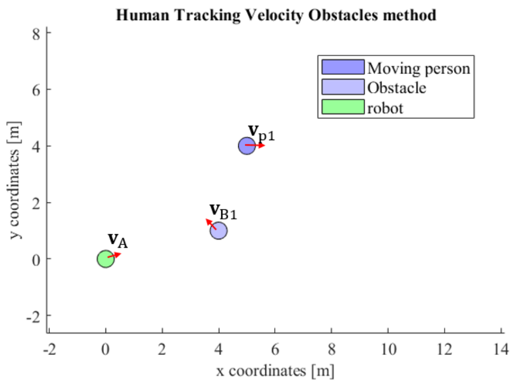

Figure 6: Simulation result in MATLAB, there is one obstacle in the workspace.

3.3 Visibility Check

In every sampling time, it must be investigated whether the human that the robot has to follow is visible or not. If it is visible than the velocity selection method can be used that was introduced in Section 3.2. The human is visible if minimum one point is observable from itself using a sensor (e.g.: LiDAR). If the human is not visible, the agent stops until the robot receives information from the human again.

In Section 4, there will be an example, where the human is not visible.

In Figure 5, the steps of the *HTVO* algorithm can be seen. After calculating the *VO-s* and *RAV* sets, it must be checked whether there are available velocity vectors in the *RAV* or not. If no, then the motion planning for the mobile robot is not executable and the agent stops. If the *RAV* set is not empty, then the visibility check is the next step. If the Human is not visible, then the agent must stop, on the other hand, if it is visible, then the velocity vector for the robot can be calculated using the *HTVO* method. Using this velocity vector for one time step, the whole method should execute again until the end of the motion is not reached.

4 SIMULATION RESULTS

In this section, the simulation results of the Human Tracking Velocity Obstacles are presented. In every simulation, the motion planning of an omnidirectional robot is presented, that follows the human.

4.1 MATLAB Simulation

The motion planning algorithm was implemented first in MATLAB environment. In every simulation result, the obstacles have constant velocity vectors. In Figure 6, there is one obstacle in the workspace of the robot which crosses the line between the robot and the human. The obstacle is presented with blue circle with velocity vector $\mathbf{v}_{B1}$ and the human (moving person) is a lila circle with velocity vector $\mathbf{v}_{p1}$. The robot is shown as a green circle with velocity vector $\mathbf{v}_{A}$. The robot has always the opportunity to select the velocity vector which will result in the human tracking method. In the middle of the motion, the human is not visible because of the obstacle. In that case the robot slows down and continues the motion only if the human is visible again. The average running time is 0.0016 s, so the introduced method can be also used in real time scenarios. The simulations were tested using the following environment:

- Processor: Intel(R) Core(TM) i5-3320M CPU @ 2.60 GHz
- Operation system: Win10, 64 bites
- Memory (RAM): 8,00 GB
- MATLAB 2021a

The motion of the robot can be seen in the following video (Gyenes, 2022a).

4.2 ROS-Gazebo Simulation

ROS (Robot Operating System) is an open source software development kit for robotics applications. It provides developers in a variety of industries with a standard software platform to aid in research and prototyping, deployment and production.

Robot simulation is a necessary tool in testing the results of the motion planning algorithms. Well-designed simulators can quickly test algorithms, design models, perform tests, and train AI systems using real-world scenarios. Gazebo provides the ability to accurately and efficiently simulate robotic populations in complex indoor and outdoor environments. To integrate ROS with stand-alone Gazebo, a set of ROS packages offer wrappers for Stand-alone Gazebo. They provide the essential interfaces to simulate robots in Gazebo with ROS messages, services and dynamic reconfiguration.

Since in simulation phase, we do not receive data from real sensor while the controller needs the position and velocity information of target and obstacle as input, two nodes are created in ROS to publish needed information of goal and obstacle respectively. Afterwards the controller node subscribes the essential data of virtual objects and then calculates the output according to the programmed algorithm and finally publishes velocity command. The introduced solution of the ROS graph can be seen in Figure 7.

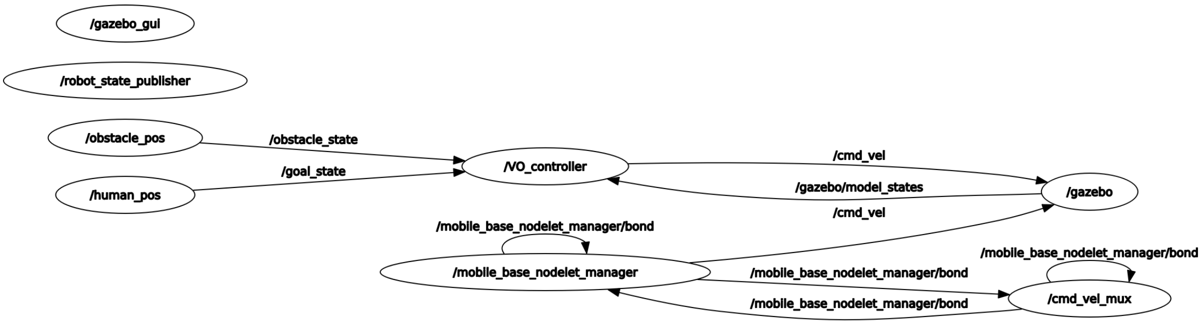

Figure 7: ROS graph of the motion planning method.

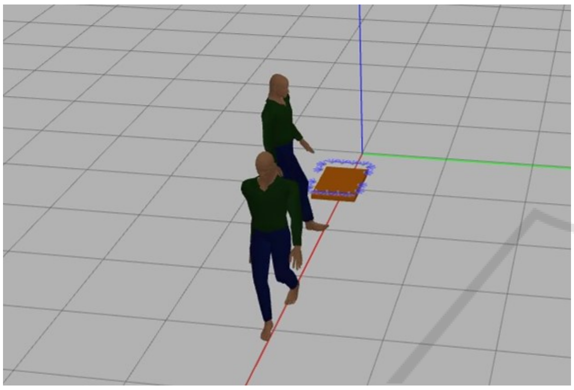

Figure 8: The robot follows the person while an another person passes the path.

Two scenarios were tested in ROS-Gazebo environment.

1. The mobile robot follows a person who walks along a straight-line trajectory, while another person (as an obstacle) passes by, which makes the goal invisible to the mobile robot. In this case, the robot would stop moving according to the introduced algorithm until it could receive the information of its goal again after the obstacle moved away. The presented example can be seen in Figure 8. Figure 9 left side represents the path of the robot and the human that the agent follows. It can be seen that the robot stops when the other person crosses the line and after that follows the target person. The right side of the picture shows the distance between the human and the robot, and the desired distance and the minimum keeping distance. It can be seen that during the motion, the distance will be even smaller until the desired distance is reached. There is a little offset between the desired and the actual distance in the end of the motion, it is because the actuators and the delays of the system but it does not influence the solution of the motion planning method. The motion of the robot can be seen in the following video as well (Gyenes, 2022b).

2. The mobile robot follows a goal person who walks along a straight-line trajectory, while a static obstacle stays between the target and robot. The goal person is still visible to robot although this obstacle exists. In this case the robot should manage to get around the obstacle generating an evasive maneuver and continues to track its target.

   This case can be seen in Figure 10.

   The path of the the motion can be seen in Figure 11, where an evasive maneuver is generated considering the static obstacle and after that, the target person is followed. On the right side of this figure, the distance can be seen between the agent and the person, which reaches the desired distance during the motion of the agent. In both examples, the human tracking task can be reached while the collision avoidance task is solved. The result of the motion can be seen in the following video (Gyenes, 2022c).

5 CONCLUSIONS

In this work, a novel motion planning algorithm was introduced for mobile robots where the human tracking method can be executed with considering the collision avoidance method at the same time. With this method, the agent can reach and keep a given distance to the target person and result in a collision-free motion in dynamic environment. The novel motion planning method was tested and validated in two different environments in different test-cases. As a future work, the introduced method would be tested on an omnidirectional mobile robot (with four omnidirectional wheels). The introduced algorithm can be also compared with an another human tracking algorithm based on Model Predictive Control (MPC). The introduced method could be used in the future in different real case situations (e.g. hospitals, airports) where it could help the people everyday life.

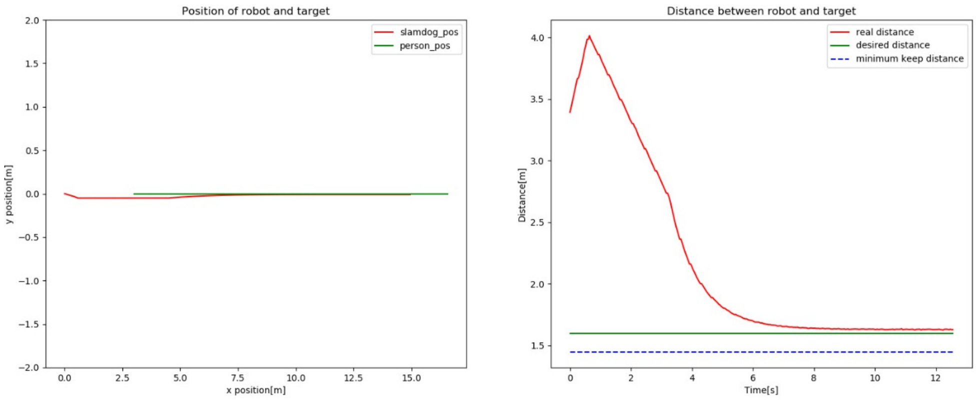

Figure 9: The path and the distance of the robot and the target.

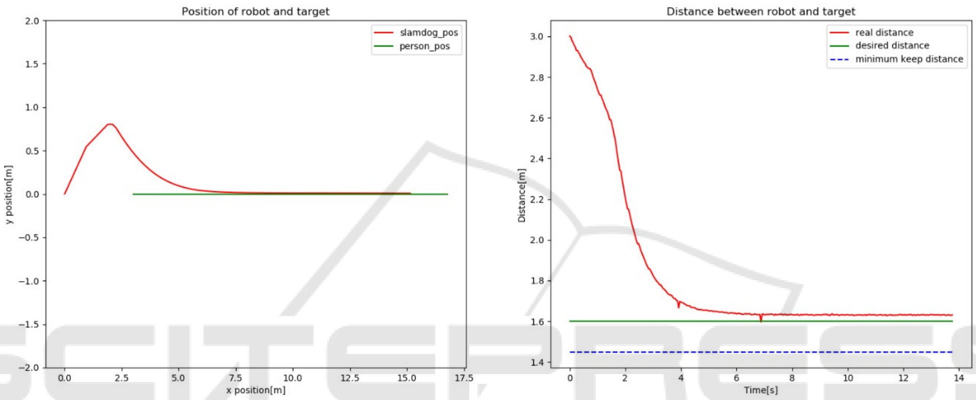

Figure 11: The path and the distance of the robot and the target in the case if there is a static obstacle between the robot and the human.

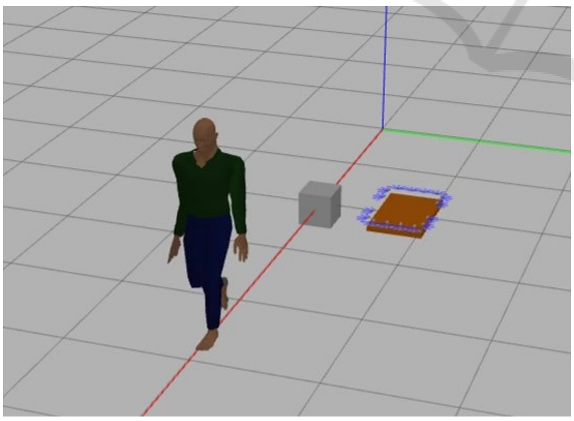

Figure 10: The robot follows the person while a static obstacle is in the workspace.

ACKNOWLEDGEMENTS

The research was supported by the DAAD Short-term Research Scholarship. Additionally, the research reported in this paper is part of project no. BME-NVA-02, implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021 funding scheme.

REFERENCES

Ali, B., Iqbal, K. F., Ayaz, Y., and Muhammad, N. (2013). Human detection and following by a mobile robot using 3D features. In *2013 IEEE International Conference on Mechatronics and Automation*, pages 1714–1719. ISSN: 2152-744X.

Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468.

Chiu, H.-K., Li, J., Ambruş, R., and Bohg, J. (2021). Probabilistic 3D Multi-Modal, Multi-Object Tracking for Autonomous Driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14227–14233. ISSN: 2577-087X.

Ch'ng, C.-H., Liew, S.-Y., Wong, C.-S., and Ooi, B.-Y. (2020). An Efficient Multi-AMR Control Framework for Parcel Sorting Centers. In *2020 IEEE Sensors Applications Symposium (SAS)*, pages 1–6.

Fedele, G., D'Alfonso, L., Chiaravalloti, F., and D'Aquila, G. (2017). Path planning and obstacles avoidance using switching potential functions. In *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics - Volume 2: ICINCO,*, pages 343–350. INSTICC, SciTePress.

Fiorini, P. and Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *The international journal of robotics research*, 17(7):760–772.

Guo, E., Chen, Z., Fan, Z., and Yang, X. (2020). Real-time Detection and Tracking Network with Feature Sharing. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 519–522. ISSN: 2642-9357.

Gyenes, Z. (2022a). Icinco conference paper, simulation in matlab. https://youtu.be/SKNefY399Co. Accessed on 2022/04/21.

Gyenes, Z. (2022b). Icinco conference paper, simulation in ros-gazebo (1). https://youtu.be/FZFnf7xBhAk. Accessed on 2022/04/21.

Gyenes, Z. (2022c). Icinco conference paper, simulation in ros-gazebo (2). https://youtu.be/1xABzIMZ3uc.

Jung, J. (2020). Woowa Brothers to Debut New Dilly Drive at End of Year. https://www.koreatechtoday.com/woowa-brothers-to-debut-new-dilly-drive-at-end-of-year/. Section: Food Service.

Kim, A., Ošep, A., and Leal-Taixé, L. (2021). EagerMOT: 3D Multi-Object Tracking via Sensor Fusion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11315–11321. ISSN: 2577-087X.

Lenain, R., Nizard, A., Deremetz, M., Thuilot, B., Papot, V., and Cariou, C. (2018a). Path tracking of a bisteerable mobile robot: An adaptive off-road multicontrol law strategy. In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics - Volume 2: ICINCO,*, pages 163–170. INSTICC, SciTePress.

Lenain, R., Nizard, A., Deremetz, M., Thuilot, B., Papot, V., and Cariou, C. (2018b). Path tracking of a bisteerable mobile robot: An adaptive off-road multicontrol law strategy. In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics - Volume 2: ICINCO,*, pages 163–170. INSTICC, SciTePress.

Masehian, E. and Katebi, Y. (2007). Robot motion planning in dynamic environments with moving obstacles and target. *International Journal of Mechanical Systems Science and Engineering*, 1(1):20–25.

Qian, Y., Shi, H., Tian, M., Yang, R., and Duan, Y. (2020). Multiple Object Tracking for Similar, Monotonic Targets. In *2020 10th Institute of Electrical and Electronics Engineers International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 360–363. ISSN: 2379-7711.

Reiser, U., Connette, C., Fischer, J., Kubacki, J., Bubeck, A., Weisshardt, F., Jacobs, T., Parlitz, C., Hägele, M., and Verl, A. (2009). Care-O-bot® 3 - creating a product vision for service robot applications by integrating design and technology. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1992–1998. ISSN: 2153-0866.

Valdez, M., Cook, M., and Potter, S. (2021). Humans and robots coping with crisis – Starship, Covid-19 and urban robotics in an unpredictable world. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2596–2601. ISSN: 2577-1655.

Wojke, N., Bewley, A., and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649.

Xiang, Y., Alahi, A., and Savarese, S. (2015). Learning to Track: Online Multi-object Tracking by Decision Making. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4705–4713, Santiago, Chile. IEEE.

Yang, M. and Jia, Y. (2016). Temporal dynamic appearance modeling for online multi-person tracking. *Computer Vision and Image Understanding*, 153:16–28.