2024 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR) | 979-8-3503-6193-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/AQTR61889.2024.10554096

Human Avoidance During Robot Operations

Delia Guran
Automation Department,
Technical University of Cluj-Napoca,
Cluj-Napoca, Romania
guran.do.delia@student.utcluj.ro

Tassos Natsakis
Automation Department,
Technical University of Cluj-Napoca,
Cluj-Napoca, Romania
tassos.natsakis@aut.utcluj.ro

***Abstract*—This paper centers on a human avoidance algorithm designed to enhance the coexistence of humans and robots within industrial settings. While both humans and robots share a workspace, it is crucial to ensure the robot operates seamlessly without encroaching upon the human's space. However, human space is not static and therefore robots need to adapt in real time to changes. The proposed approach involves employing a depth camera to actively identify and track humans in real time. It determines the distance between the human and the robot's end effector and computes a repulsive vector. At the same time, an attractive vector is calculated from the robot end effector towards the target pose. Based on the addition of the repulsive and attractive vectors, a new trajectory for the robot is calculated, keeping the same goal pose. This enables the robot to steer clear of potential collisions while achieving its goal. Human detection is facilitated using an Astra Pro depth camera from Orbbec. We demonstrate this algorithm on the UR5 robot from Universal Robots which it is tasked with moving at a velocity of 0.1 m/s towards predefined targets. The avoidance algorithm remains constantly active, ensuring continuous trajectory for the robot post-avoidance of the human presence.**

Keywords—Avoidance, collision, repulsive field, robot, robotic, ROS, collision.

I. Introduction

Robots are often associated with the notion of "danger". Indeed, when humans enter a robot's operating area, unexpected outcomes might arise. The most commonly employed solution consists of isolating the robot in a cage to carry out tasks undisturbed. However, technological evolution dictates that robots are expected to perform various tasks both in collaboration with humans and often in an environment in which people live, e.g., when performing household chores, waste disposal, cooking, entertainment and more. Collaboration between humans and robots is also a trend in industrial setups, with the introduction of co-bots (collaborative robots), where robots and humans combine their abilities to achieve more complex tasks. Implementing collision avoidance software ensures human safety within a robot while allowing the robot to maintain productivity. The primary advantage of human avoidance lies in the robot's ability to identify obstacles, navigate around them and continue its tasks uninterrupted. This approach prioritizes human safety, ensures task completion and sustains production levels without incidents.

Collision avoidance has been one of the most studied fields in robotics due to its relevance and many various planning and control systems for obstacle avoidance have been developed [1], [2], [3]. It requires a quick way of determining the distance between a robot and an obstacle.

An online algorithm in an augmented environment is being used in [4] to monitor the workspace and to prevent collisions by using 3D models of robots and images of people from two depth cameras are utilized. When the two cameras are calibrated, the background of the images is removed, leaving just the unknown objects.

Hong et al [5] outline a C-space strategy ensuring safe human-robot interaction. In the process of constructing a roadmap, this method employs a distance measure to select neighboring sampled nodes within C-space. A more efficient alternative to cell decomposition, which requires a substantial amount of time, is utilizing sampled surface points to represent the obstacle, making the roadmap a preferable approach.

Human recognition is a key step for a human avoidance algorithm. In [6], the skeleton data from Orbbec and Kinect sensors were compared. The purpose is to verify if the Orbbec sensor can be used instead of the Kinect for the gesture recognition. The results were similar, the difference is that the Kinect 2 skeleton can detect 6 extra joints.

Since collision avoidance is an important issue, there are many alternatives to solve it. The robot should work efficiently even if it must navigate around obstacles, so a method often employed is to estimate the pedestrian's behavior and avoid it only when the robot does not detect any human intention to prevent the impact [7]. By calculating a trajectory of the pedestrian, the robot can determine whether the person intends to avoid the encounter, and if he/she does, the robot will not change its behavior. In [8], the distance between the robot and the human was calculated with a sphere-based geometric model. This collision check method is based on predicted movements of both the robot and the human for several time-steps. When a possible collision appears, the cost of each search direction for predicted locations is calculated, and the lowest cost will be the best choice for the next step of the robot.

One point of view of human-robot coexistence is the use of a high-speed tracking vision-chip [9]. This is a smart camera that requires fast communication and powerful computers to analyze image data. The processing elements and the photodetectors are integrated on a single chip with only binary images and each pixel can store only one bit.

Both the human and the robot collaborate to avoid a future collision in [10]. Human-robot interaction (HRI) is achieved through voice messages from the robot and gestures from the human.

In industrial and dynamic environments, the distance between the manipulator and the obstacle can be calculated from the depth data. In [11] there are two methods, one is that the repulsive vector is associated only with the obstacle with the shortest distance. The other option is to use all the obstacle points for the direction of the repulsive vector because the repulsive vector is not sensitive to the noise of the depth sensor and is handled better.

In this paper, a new collision avoidance algorithm was developed to establish the human-robot collaboration with the help of the repulsive field that pushes the robot away from the dynamic obstacle (human). The robot returns to its original task even in the presence of a human. Therefore, the solution involves calculating an attractive force as well, inducing the

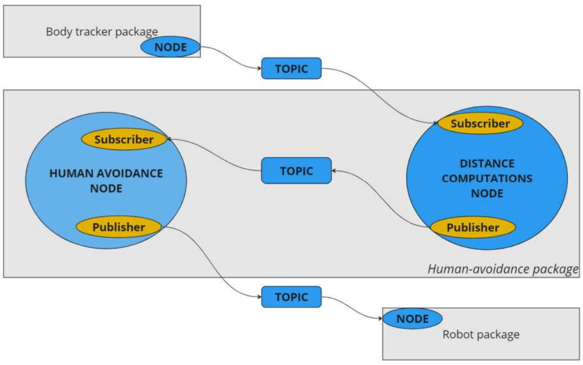

Fig. 1. ROS Structure of the algorithm

robot to follow an avoidance path while simultaneously approaching the target pose.

The structure of the paper is outlined as follows: The concept of Robot Operating System is summarized in Section II. Section III details the computation of the repulsive field, covering the human-robot distance (Section III-A) and calculation involved in generating the repulsive force (Section III-B). Section IV discusses human avoidance, including the trajectory computation (Section IV-A) and the utilization of inverse kinematics (Section IV-B). Section V provides a description of the laboratory setup and presents the experimental results obtained.

II. Robot Operating System (ROS)

ROS [12], the Robot Operating System, is an open-source framework for robotics development. It features a modular structure with nodes, representing task-specific units, and topics facilitating communication between nodes through message exchange. Organized into packages, ROS enables scalable and collaborative development by encapsulating related functionalities such as nodes, libraries, datasets, and configuration files. Fig. 1 represents the ROS structure of the proposed algorithm.

III. Repulsive Field Calculation

A. Camera-Robot Distance

The camera is frequently defined in computer vision applications based on a set of intrinsic and extrinsic parameters. The extrinsic parameters determine the camera's position and orientation with respect to a fixed frame of reference by determining its rotation and translation. In our setup, extrinsic calibration is performed once at the beginning of each test session, using the asymmetric circle pattern shown in Fig. 2.

The pattern is positioned and oriented on a stable and predefined location relative to the robot. The pose from the pattern to the robot is therefore known and as post-calibration, the transformation from the camera to the pattern is calculated. Subsequently, the task involves determining the robot's coordinates within the camera's frame using (1).

$$\mathrm{R}_{\mathrm{C}}^{\mathrm{P}} \cdot \mathrm{R}_{\mathrm{P}}^{\mathrm{B}} = \mathrm{R}_{\mathrm{C}}^{\mathrm{B}} \tag{1}$$

Where $\mathrm{R}_{\mathrm{C}}^{\mathrm{P}}$ is the transformation between the camera frame and the pattern frame, $\mathrm{R}_{\mathrm{P}}^{\mathrm{B}}$ is the transformation between the pattern frame and the robot's base link frame and $\mathrm{R}_{\mathrm{C}}^{\mathrm{B}}$ is the transformation between the camera frame and base link frame.

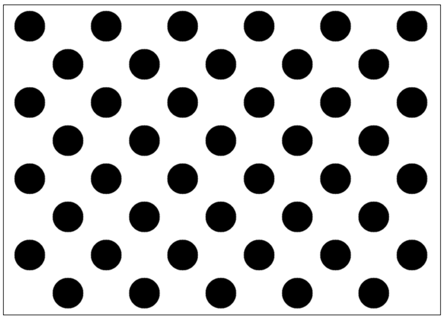

Fig. 2. Asymmetric circle pattern

B. Human-Robot Distance

Real time data on the distance between the robot and the human are essential for dynamic obstacle avoidance. The skeleton tracking package provided by the camera manufacturer, replicates the coordinates of individual human joints within the camera frame through subscription to the skeleton topic. Rather than transforming every human joint to the end-effector frame, a more straightforward method involves transforming only the robot's end-effector pose into the camera frame. This ensures uniformity of all values within the same coordinate system. Equation (2) presents the end-effector coordinates within the camera frame.

$$\mathrm{R}_{\mathrm{E}}^{\mathrm{B}} \cdot \mathrm{R}_{\mathrm{B}}^{\mathrm{C}} = \mathrm{R}_{\mathrm{E}}^{\mathrm{C}} \tag{2}$$

Where $\mathrm{R}_{\mathrm{E}}^{\mathrm{B}}$ is the transformation between the robot's end-effector frame and the robot's base link frame, $\mathrm{R}_{\mathrm{B}}^{\mathrm{C}}$ is the transformation between the base link frame and the camera frame and $\mathrm{R}_{\mathrm{E}}^{\mathrm{C}}$ is the transformation between the end effector's frame and the camera frame. In Fig. 3 we are presenting the overall transformations.

In terms of avoidance, one approach is the use of vector fields. The main goal of an obstacle vector field is to guide the robot's arm, enabling it to steer clear of humans. This method aims to generate a repelling force that is inverse proportional to the distance between the robot and the obstacle. When multiple obstacle points exist, numerous forces are combined to create a combined obstacle repulsive vector. The joints of the body serve as the obstacle points, necessitating the computation of the repulsive force exerted by each segment of the skeleton.

In (3), there is the summation between the camera-human joint vector ($\overrightarrow{\mathrm{CH}}$) and the human joint-robot vector ($\overrightarrow{\mathrm{HR}}$),

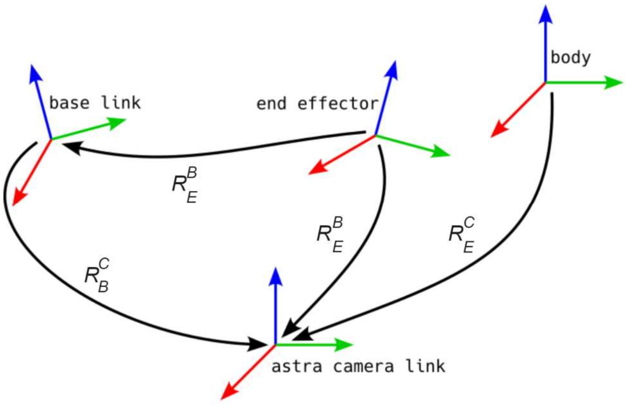

Fig. 3. Coordinate Frames kinematic chain

which, according to the triangle law of adding vectors, is equal with the camera-robot vector ($\overrightarrow{CR}$). For each relevant skeleton joint, the equation (4), which is extracted from (3), is computed to find the human joint-robot vector ($\overrightarrow{HR}$).

$$\overrightarrow{CH} + \overrightarrow{HR} = \overrightarrow{CR} \tag{3}$$

$$\overrightarrow{HR} = \overrightarrow{CR} - \overrightarrow{CH} \tag{4}$$

Once the direction of the repulsive force is determined, the subsequent step involves computing the distance between the human joint and the robot's end-effector. The calibration provides the camera-robot distance, while the skeleton tracker algorithm offers the camera-human joint distance. Therefore, to obtain the required distance (denoted with $||\overrightarrow{HR}||$) is calculated as in (5).

$$\|\overrightarrow{HR}\| = \sqrt{(H_x - R_x)^2 + (H_y - R_y)^2 + (H_z - R_z)^2} \tag{5}$$

Where H_x, H_y, and H_z are the human joint's coordinates, and R_x, R_y, and R_z are the end-effector's coordinates. All coordinates are in the camera frame.

C. Repulsive Field

The repulsive force is the vector influencing the robot's end effector to maintain distance from the human. The direction of that vector should point towards the robot, and it should be inversely proportional to the distance from the end-effector to the human. Essentially, as the proximity between the robot and the human decreases, the repulsive force increases and vice-versa. The repulsive force for each skeletal joint is calculated using (6).

$$F_i(H, R) = \frac{\overrightarrow{HR}}{\|\overrightarrow{HR}\| \cdot \|\overrightarrow{HR}\| + \varepsilon} \tag{6}$$

Where $F_i(H, R)$ is the repulsive force of joint 'i', for each skeletal joint, and it depends on the human joint (H) and robot (R) coordinates. The $\overrightarrow{HR}$ is the human joint-robot vector and $||\overrightarrow{HR}||$ is the distance between the human joint and the robot, calculated with formula (5).

The first inverse of the distance in formula (6) is to normalize the length of the repulsive vector and the second one is for the inverse proportionality. In the case where the distance is somehow 0 ($||\overrightarrow{HR}|| = 0$), the repulsive force could not be correctly calculated, that is why the constant error ε is needed. The ε is a small constant with an imposed value that ensures the denominator is never 0.

Equations (4), (5), and (6) are applied for every relevant skeleton joint. This paper prioritizes the upper body, leading to the repetition of the earlier computations eleven times, yielding a total of eleven repulsive forces determined. Certain areas of the human body hold greater significance in terms of collision avoidance. To differentiate the importance of the human parts, weights are added for each repulsive force of the upper body.

The weights assigned to each skeletal part are represented by natural numbers raging from one to four. A weight of one signifies lower importance, such as the palm, while a weight of four indicates the highest importance, such as the head. All the repulsive forces are added to create the repulsive vector for the entire body. The formula (7) is the computation for the repulsive force.

$$F(H, R) = \textstyle\sum_{i=1}^{n} (w_i \cdot F_i(H, R)) \tag{7}$$

Where F(H, R) is the repulsive force, the sum of the repulsive forces of all the joints, w_i is the weight for every joint 'i', $F_i(H, R)$ is the repulsive force for every joint 'i' and n is the total number of skeleton joints.

The main coordinate system for all the computations is the camera frame. Once the repulsive force starting point is determined, its vector will be at the camera origin instead of the end effector's position. Consequently, to adjust this, the vector must be shifted by adding the position of the end effector in the camera frame, to the position of the repulsive force. The result is the vector pointing from the human to the robot.

To enhance the avoidance algorithm, an extension of the repulsive force is required for further computations. This extension must originate from the end effector's current position and should also be inversely proportional with the human-robot distance. The extended repulsive force, $F_E(H, R)$ is equal with the repulsive force for all the body, F(H, R) (8).

$$F_E(H, R)= F(H, R). \tag{8}$$

The coordinates for the new repulsive force vector and the distance between the human and the robot's end-effector are sent to the avoidance node by a ROS publisher.

IV. Human Avoidance

A. Trajectory Computation

In the previous section, the repulsive force was calculated in the camera frame. Subsequently, computations no longer involve the camera frame, making it preferable to solely operate within the robot's frame. The repulsive force coordinates are transformed into the end effector's coordinate system using preexisting transformations. Formula (9) underlies this transformation.

$$R_C^B \cdot R_B^E = R_C^E \tag{9}$$

Where R_C^B is the transformation between the camera frame and the robot's base link frame, known from extrinsic calibration. R_B^E is the transformation between the base link frame and the robot's end effector's frame and R_C^E is the transformation between the camera frame and the end effector's frame.

The user sends the target point, and the robot operates with a single goal at a time. Transformation (10) is only performed if the user sends the coordinates in a different frame than the required one, which is the base frame.

$$R_T^X \cdot R_X^B = R_T^B \tag{10}$$

The indices from (10) follow the same logic as before.

Next, we calculate the avoidance vector. The target vector is the one pointing to the end pose of the robot's end-effector. If it is in the base frame of the robot and for finding the avoidance vector, all vectors need to be in the end-effector

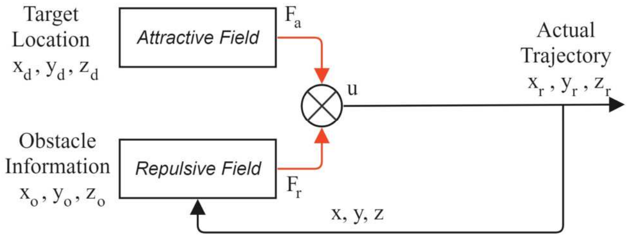

Fig. 4. Architecture of the trajectory computation

frame, so the target pose should be transformed into the end effector frame using (10).

The repulsive force directs the robot away from the human. The vector pointing toward the goal is the attractive force, because it pulls the robot towards the desired pose. Having the attractive force and the extended repulsive force vectors, the avoidance vector can be readily derived through their summation.

The architecture of the trajectory of the robot is depicted in Fig. 4, where x_d, y_d, z_d are the desired position for the robot's end effector, x_o, y_o, z_o are the coordinates for repulsive force position, x_r, y_r, z_r are goal point of the trajectory, F_r is the repulsive force, F_a is the attractive force, u is the sum of F_r and F_a and x, y, z are new robot's coordinates.

The trajectory of the robot should align with the avoidance vector end point in the absence of a nearby human. The values of the repulsive force might be large, potentially causing the avoidance vector also to be unattainably large for the robot. To mitigate this, a scale of 0.25 for the repulsive force is made. The trajectory vector should be divided into smaller steps so that the repulsive force and the avoidance vector update faster when a human is in proximity. In such scenarios, a period of 0.01 is multiplied by the final avoidance vector. This period cannot exceed the value one.

B. *Inverse Kinematics*

Inverse kinematics (IK) is a function to determine joint positions enabling the robot to attain a specific orientation and position. TRACLabs engineered an inverse kinematics solver that has two parts [13]. First involves specifying the base and end link, the URDF parameter, a timeout in seconds, the error and the solver type. The type of the solver chosen in this paper is Speed, the timeout is 0.008 seconds, and the error is 1e-5. The secondary part is a method responsible for computing the joint values and returning a Boolean output. The parameters for that method are the initial guess, which is set to the robot's current position, the target pose, and the target joints. The last parameter is given only to be updated and to be used next. Once the target joints are identified, the corresponding values associated with the correct joint name are transmitted to the robot, prompting it to move and attain the desired pose.

In the context of inverse kinematics, the coordinates indicating the desired position of the end effector, must be in the base frame. Consequently, the target and the avoidance pose need to be into the base coordinate system before being sent to the IK function. The target received is already in the robot base frame, only the avoidance pose requires transformation. After the transformation of the avoidance is complete, the target incorporates the avoidance value and is subsequently transmitted to the IK function.

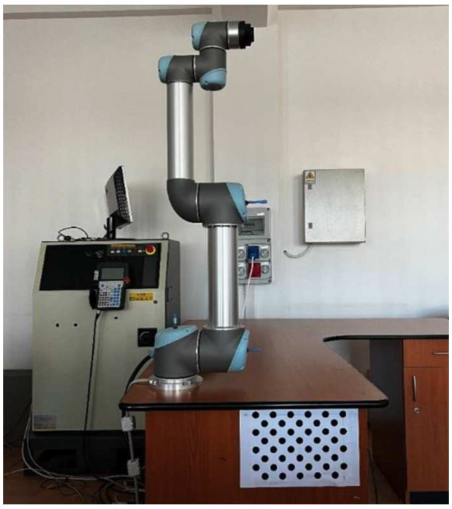

Fig. 5. Pattern's location related to the robot

After the robot reaches that goal, if an obstacle emerges within the robot's workspace, the robot will avoid it and return to the predetermined target location.

V. EXPERIMENTS

A. *Simulation*

Before conducting tests involving human avoidance algorithm on the actual robot, several simulation tests were performed. These simulations entailed connecting the camera to the laptop and establishing a randomized position and orientation for the transformation between the camera and the simulated robot. A Gazebo simulator was employed to replicate the real robot. Upon successful simulation scenarios, the algorithm underwent testing on the real robot.

B. *Testing Setup*

To establish the camera's position relative to the robot, conducting extrinsic calibration is essential before algorithm testing. The calibration pattern is affixed to the table where the robot is positioned, as in Fig. 5.

C. *Testing*

Two main scenarios were tested: In the first one, the robot is in motion, and it is trying to reach a user-specified position and the human steps in. In the second scenario, the robot is

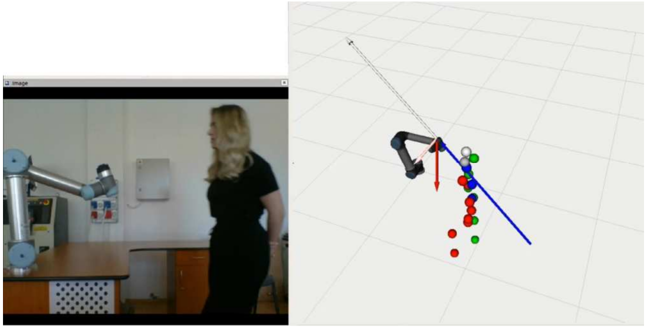

Fig. 6. RViz representation of the avoidance algorithm. The human is depicted using spheres colored in white, blue, red and green, representing the skeleton joints. The blue vector is the repulsive force, the white vector is the extended repulsive force, the red vector is the attractive force, and the pink vector is the avoidance vector. The robot avoids the human while moving toward the target pose.

already positioned as per the user's input and a human approach the end effector. Both tests were conducted at a robot velocity set to 30% of the maximum velocity.

To enhance the algorithm's visualization, RViz is used. Fig. 6 showcases these situations within a working environment. On the left side of the figure, live images from the workspace are displayed, while on the right side, the online graphical interface of the human avoidance algorithm is presented. This interface includes the robot, the skeleton recognition and tracking, and the computations for repulsive force, attractive force and the avoidance vector. The UR5 robot is accurately represented to mirror its real appearance. The repulsive force is illustrated at its initial length. However, in computations, its length is four times smaller, influencing the direction and magnitude of the avoidance vector.

The robot is receiving a new target position from the user and encountering the human while the robot is moving toward that position. Initially, the robot follows the avoidance vector, which is the same as the trajectory vector. However, as the human approaches, the repulsive force increases, and the avoidance vector will be different from the target one. The robot avoids the human and goes near its goal position. Only when the human is distant enough from the workspace, the robot steers towards its imposed position because the trajectory vector will coincide with the attractive force vector.

While the robot is in motion towards the designated target point, it continuously assesses its surroundings. In the event of a human lowering weighting point, such as the hand is approaching, the avoidance is triggered. Consequently, the avoidance vector diverges from the original target vector, enabling the robot to steer clear of potential collisions with the approaching human hand.

In Fig. 7 is presented the robot's end-effector avoiding the human oscillating near the fixed target position. Fig. 8 and 9 depict the robot performing a task, with movement from one point to another. In Fig. 8, the robot operates without human interference, allowing for an undisturbed trajectory. Conversely, Fig. 9 shows the presence of a human within the workspace, whose influence introduces an avoidance force that disturbs the robot's trajectory. The vertical axis of the figures illustrates the x-axis position in the robot's base frame, and the horizontal axis represents time in seconds.

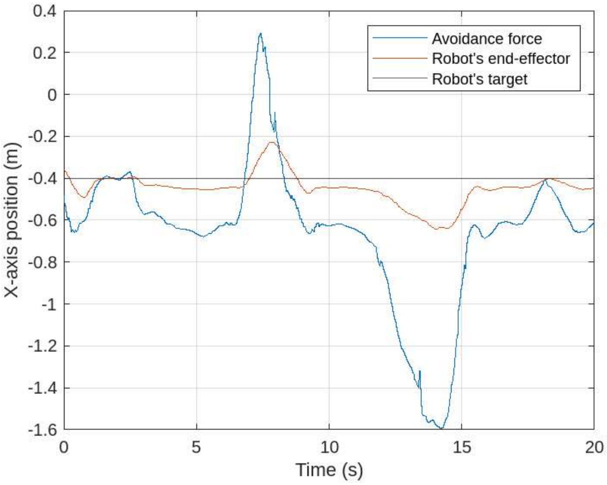

Fig. 7. The response of the robot end-effector to the application of avoidance force. The target point of the robot along the x-axis is prescribed as -0.4, with the robot exhibiting oscillatory motion in proximity to this value contingent upon the intensity of the force.

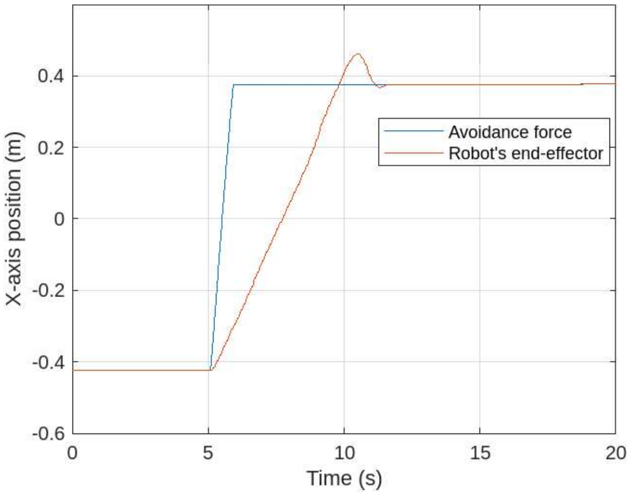

Fig. 8. The response of the robot's end-effector during task execution, under conditions with no human interference. The target point of the robot along the x-axis changes at the 5-second mark from -0.4 to 0.4.

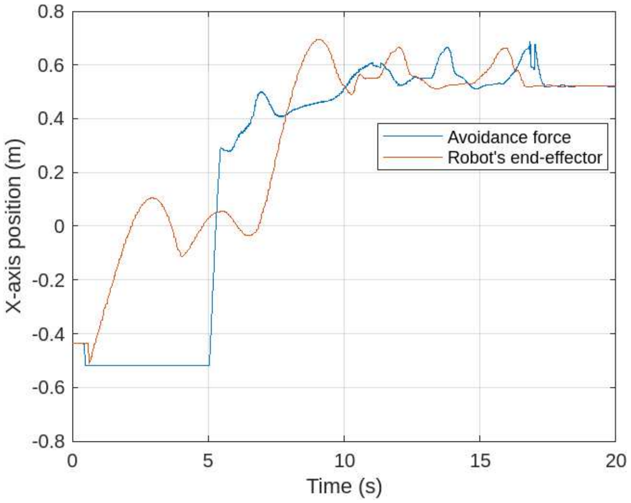

Fig. 9. The response of the robot's end-effector during task execution, under conditions subject to human interference. The target point of the robot along the x-axis changes at the 5-second mark from -0.4 to 0.4.

D. *Validation*

The algorithm underwent multiple tests involving varied positions for the human's head, hands, body, as well as different positions for the robot's end effector. In each instance, the robot successfully avoided the human and navigated back to its imposed position.

VI. Conclusions

This paper introduces a real-time algorithm designed for collision avoidance involving the manipulator robot UR5 equipped with an AstraCamera Pro sensor. Its purpose is to improve the safety of the working environment for operators by assessing the distance between the robot's end effector and every relevant joint of the human's skeleton. The algorithm is developed to obtain a protected human-robot coexistence and consists of a repulsive field for each significant human joint that will provide the repulsive force. The repulsive field only leads the robot to move in the opposite direction from the human. The attractive force is the vector that goes toward the user specified goal point for the robot's end effector. Both the repulsive force and the attractive force must be considered for the robot to avoid the human and to follow a trajectory that gets to its goal point. In this way, a new force is created to satisfy the human avoidance and the continuity of the robot's task.

This approach offers notable advantages, fostering human-robot coexistence in industrial workspace and manufacturing, while concurrently boosting productivity. This system safeguards both human and robot from harm during operations, facilitating uninterrupted production. Moreover, the use of an industrial robot can reduce labor costs over time, as a robot typically proves more cost-effective than a person performing multiple repetitive operations.

Acknowledgment

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI-UEFISCDI, project number PN-IV-P8-8.1-PRE-HE-ORG-2023-0012, within PNCDI IV.

References

[1] J.-H. Chen and K.-T. Song, "Collision-Free Motion Planning for Human-Robot Collaborative Safety Under Cartesian Constraint," in 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD: IEEE, May 2018, pp. 4348–4354. doi: 10.1109/ICRA.2018.8460185.

[2] W. Zhang, H. Cheng, L. Hao, X. Li, M. Liu, and X. Gao, "An obstacle avoidance algorithm for robot manipulators based on decision-making force," Robot. Comput.-Integr. Manuf., vol. 71, p. 102114, Oct. 2021, doi: 10.1016/j.rcim.2020.102114.

[3] J. Heinzmann and A. Zelinsky, "The safe control of human-friendly robots," in Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289), Kyongju, South Korea: IEEE, 1999, pp. 1020–1025. doi: 10.1109/IROS.1999.812814.

[4] B. Schmidt and L. Wang, "Depth camera based collision avoidance via active robot control," J. Manuf. Syst., vol. 33, no. 4, pp. 711–718, Oct. 2014, doi: 10.1016/j.jmsy.2014.04.004.

[5] Hong Liu, Xuezhi Deng, and Hongbin Zha, "A planning method for safe interaction between human arms and robot manipulators," in 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alta., Canada: IEEE, 2005, pp. 2724–2730. doi: 10.1109/IROS.2005.1545241.

[6] A. D. C. A. Coroiu and A. Coroiu, "Interchangeability of Kinect and Orbbec Sensors for Gesture Recognition," in 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca: IEEE, Sep. 2018, pp. 309–315. doi: 10.1109/ICCP.2018.8516586.

[7] Y. Tamura, T. Fukuzawa, and H. Asama, "Smooth collision avoidance in human-robot coexisting environment," in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei: IEEE, Oct. 2010, pp. 3887–3892. doi: 10.1109/IROS.2010.5649673.

[8] L. Balan and G. Bone, "Real-time 3D Collision Avoidance Method for Safe Human and Robot Coexistence," in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China: IEEE, Oct. 2006, pp. 276–282. doi: 10.1109/IROS.2006.282068.

[9] D. Ebert, T. Komuro, A. Namiki, and M. Ishikawa, "Safe human-robot-coexistence: emergency-stop using a high-speed vision-chip," in 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alta., Canada: IEEE, 2005, pp. 2923–2928. doi: 10.1109/IROS.2005.1545242.

[10] M. Ghandour, H. Liu, N. Stoll, and K. Thurow, "A hybrid collision avoidance system for indoor mobile robots based on human-robot interaction," p. 7.

[11] F. Flacco, T. Kroger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN: IEEE, May 2012, pp. 338–345. doi: 10.1109/ICRA.2012.6225245.

[12] Stanford Artificial Intelligence Laboratory et al., "Robotic Operating System [Internet]." [Online]. Available: https://www.ros.org

[13] B. Patrick and A. Barrett, "TRAC-IK: An Open-Source Library for Improved Solving of Generic Inverse Kinematics," in 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, Korea (South): IEEE, 2015, pp. 928–935. doi: 10.1109/HUMANOIDS.2015.7363472