ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20250320002497
Original scientific paper

Affordable Collision Detection for a 3D-Printed Robotic Arm

Almir OSMANOVIĆ*, Malik ČABARAVDIĆ, Amer SARAJLIĆ, Jasmin HALILOVIĆ, Kenan VARDA

Abstract: Collision detection and avoidance are crucial for safe and efficient robotic manipulation. This paper presents an affordable collision detection system for a 3D-printed robotic arm using a Kinect camera integrated with the Robot Operating System (ROS). The proposed approach enables real-time environment perception and collision-free trajectory generation. Experimental results show a significant improvement in collision avoidance efficiency, achieving a 40% reduction in path length while maintaining real-time performance. The computational overhead introduced by collision detection is minimal, with a 12 ms delay per frame, which can be optimized further. It is demonstrating the system's ability to effectively handle dynamic environments, such as human obstacles. This work highlights the potential of low-cost 3D sensing solutions for robotics applications, making them more accessible and scalable.

Keywords: collision detection; kinect; mechatronic; robotic; robotic arm; ROS; trajectory optimization

1 INTRODUCTION

Robotics is an interdisciplinary field that represents the synergy of mechanical engineering, electrical engineering and computer science with a solid mathematical foundation and as such provides the opportunity for integration in a wide range of theoretical and practical problems. Robots in industry today significantly influence the maintenance of uniform product quality, efficient use of materials, worker safety and the elimination of human errors, and that is the main reason for which there is the much greater precision and repeatability of robots in work than humans. Most robots in industry need to physically interact with the object of work at least in some part of the production process, which is why in most cases it is necessary to pay attention to robot handling and working (Fig. 1). Handling is a very complex task for both humans and robots, and a generalized solution to the problem of robotic handling for various objects in an unstructured environment has not yet been defined. By definition, an industrial robot is an automatically controllable, reprogrammable, multi-purpose manipulator programmable in three or more axes, which can be either stationary or mobile for industrial applications. Industrial robots are also called robotic manipulators or robotic arms.

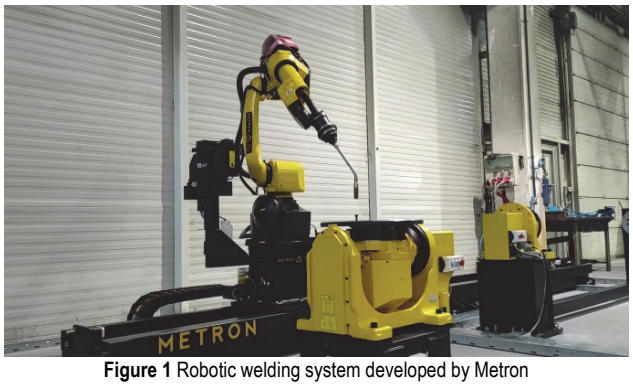


Figure 1 Robotic welding system developed by Metron

Robotic manipulators require efficient collision avoidance techniques to operate in dynamic environments [1-9]. Traditional approaches rely on expensive sensors and computationally intensive algorithms. However, affordable solutions, such as consumer depth cameras, have gained popularity in recent years [10, 11]. In this work is explored the use of a Kinect camera for real-time obstacle detection and avoidance in a 3D-printed robotic arm. Many research works have addressed navigation and obstacle avoidance for mobile robots [12, 13]. These approaches typically are of two categories: navigation in known environments using pre-built maps and real-time navigation in unknown environments through sensor data [14, 15]. Fig. 2 shows examples of one of the developed systems for robots avoiding collisions. In this paper work is focused on real-time collision detection for robotic manipulators using Kinect depth sensing and ROS. The system is implemented on the Arctos 3D-printed robotic arm, which provides a cost-effective and customizable platform for experimentation.

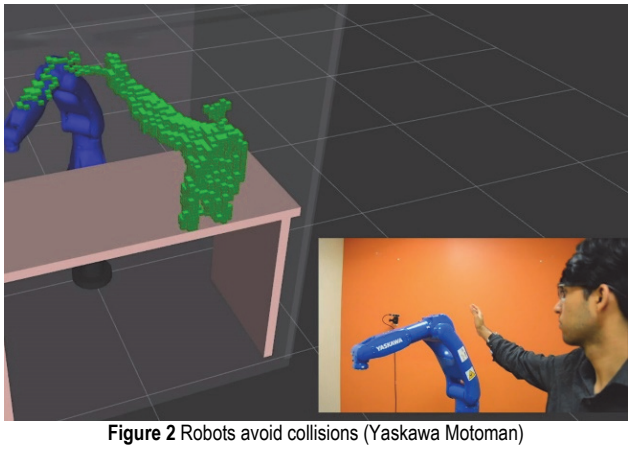


Figure 2 Robots avoid collisions (Yaskawa Motoman)

1.1 Background and System Architecture

The field of affordable robotics has gained significant attention in recent years, with several research groups exploring cost-effective solutions for robotic applications, as well as modelling and simulation [16-18]. Consumer-grade depth cameras like the Microsoft Kinect have transformed the landscape of affordable 3D sensing. Originally designed for gaming interfaces, these devices have been repurposed for robotic applications due to their ability to provide depth information at relatively low cost.

Several studies have explored the use of Kinect sensors in robotic applications [19-21]. For example, Zhang et al.

[22] demonstrated the use of Kinect for human-robot interaction, while Wang et al. [23] utilized Kinect for gesture recognition in robotics. However, the application of Kinect specifically for collision detection in robotic arms remains relatively unexplored, especially in the context of 3D-printed robotic manipulators.

The integration of Kinect with ROS has been facilitated by open-source software packages. The 'openni_kinect' and 'freenect' drivers enable seamless integration of Kinect data with ROS. These drivers convert raw Kinect data into ROS-compatible formats, allowing for further processing and integration with motion planning algorithms.

3D-printed robotic arms represent another emerging trend in affordable robotics. Additive manufacturing techniques have enabled the creation of low-cost, customizable robotic platforms suitable for educational and research purposes. The Arctos robotic arm, used in this study, exemplifies this trend, featuring 3D-printed structural components and affordable actuators. The Arctos robotic arm is a 6-DOF manipulator designed for educational and research purposes. It is constructed using 3D-printed components and cycloidal drives, making it an affordable and customizable platform. The arm is equipped with stepper motors for precise control and is controlled with a PC running ROS.

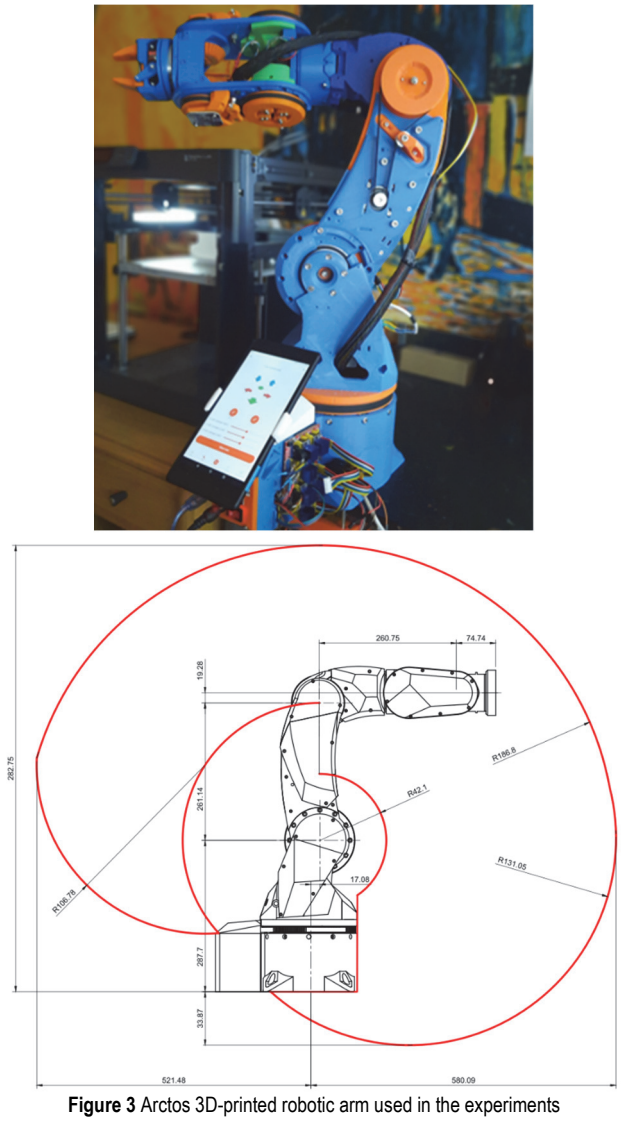


Figure 3 Arctos 3D-printed robotic arm used in the experiments

Kinect camera, originally designed for gaming, was modified to work with a PC. Additionally, custom drivers and setup were required to enable Kinect functionality on a Linux-based system. These modifications made the Kinect a viable low-cost solution for depth sensing in robotic applications. The Kinect camera collects and stores depth data, which are processed later using the 'depth_to_pointcloud' node. Collision detection is handled using a combination of OctoMap and PointCloud2 data. Additionally, a filtering module is applied to smoothen and eliminate noisy depth readings, improving the accuracy of the obstacle map.

The presented system in the work consists of:

- 3D-printed Arctos robotic arm controlled through ROS.
- Kinect camera for depth sensing.
- ROS operating system for processing Kinect data and integrating it into motion planning.
- Trajectory optimization module for refining motion plans.

2 COLLISION DETECTION AND AVOIDANCE

Collision detection is implemented using the PointCloud2 data from the Kinect camera. The environment is continuously updated to reflect dynamic obstacles, ensuring real-time adaptability to changes in the workspace. If an obstacle is detected along the planned trajectory, the motion planner dynamically computes an alternative path to avoid collisions while maintaining smooth and efficient movement.

The system employs a multi-stage collision detection pipeline that begins with raw point cloud acquisition, followed by statistical outlier removal to reduce sensor noise. A voxel grid filter then down samples the data to optimize computational efficiency while preserving essential spatial information. The filtered point cloud undergoes segmentation to distinguish between the workspace, static environment, and potential obstacles. This segmentation utilizes both geometric and temporal features, allowing the system to identify moving objects with greater accuracy. The collision detection algorithm implements a hierarchical bounding volume approach, starting with coarse approximations before refining to precise object boundaries. This multi-resolution strategy significantly reduces computational overhead while maintaining detection accuracy within the 5 mm threshold required for safe operation.

To provide understanding of the system's workflow, a flowchart is present in Fig. 4. The flowchart illustrates how user inputs from a controller are processed by the PC, passed through RViz for visualization, and further handled by the inverse kinematics solver. The computed joint angles are then refined by a post-processing module before being sent as commands to the robotic arm. This structured approach ensures precise control and seamless interaction between the operator and the robot.

This modular view of the system improves system flexibility, allowing real-time adjustments based on sensor feedback. By integrating collision detection and avoidance mechanisms within this framework, the robotic arm can operate safely in dynamic environments, making it suitable for various real-world applications.

The MoveIt! package uses the OctoMap-based 3D occupancy grid to determine collision-free pathways. The motion planner adapts to new obstacles and ensures safe implementation of planned trajectories. Fig. 5 shows two motion scenarios: one with direct motion from start to end-goal and another with collision avoidance enabled.

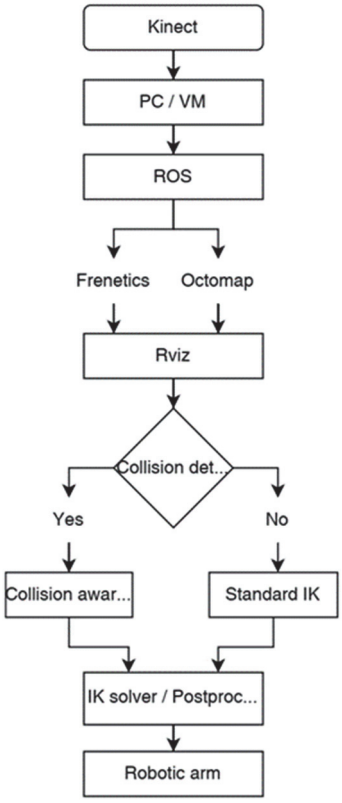


Figure 4 System workflow from Xbox controller to robotic arm, including visualization, inverse kinematics, and motion planning

The core of the collision detection system relies on processing point cloud data from the Kinect sensor. The raw depth data from the Kinect is converted into a 3D point cloud using the 'depth_to_point cloud' node in ROS. This point cloud represents the 3D environment as seen by the Kinect camera.

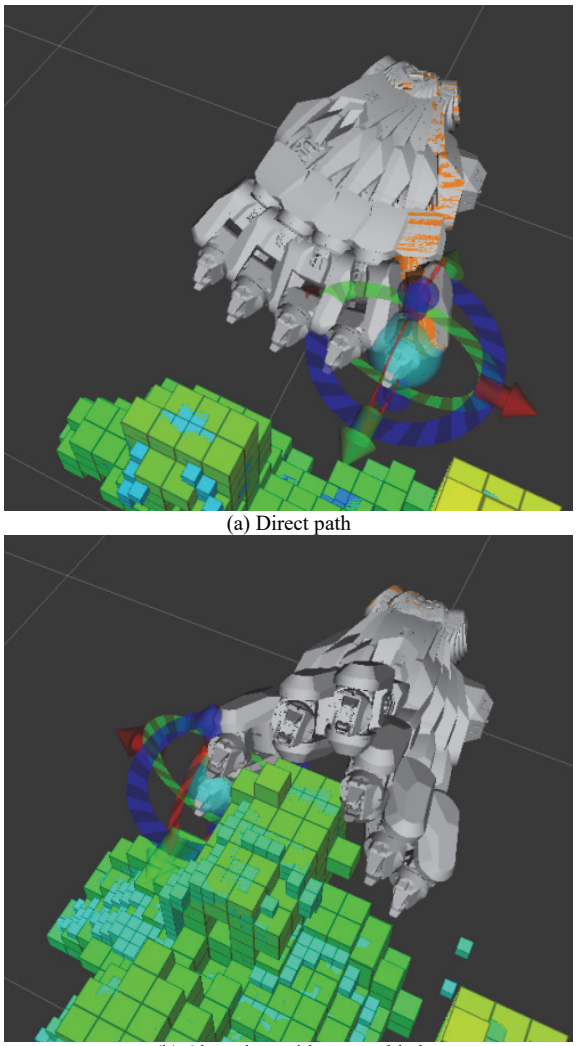
To make the point cloud data usable for collision detection, several processing steps are applied:

1. Noise Filtering: A statistical outlier removal filter is applied to eliminate spurious points caused by sensor noise or reflective surfaces.
2. Down sampling: The point cloud is down sampled using a voxel grid filter to reduce computational complexity while maintaining essential geometric information.
3. Ground Plane Removal: The ground plane is identified and removed to focus on obstacles relevant to the robotic arm's motion.
4. Clustering: Point cloud clustering is performed to identify distinct objects in the scene.

Once the point cloud is processed, it is converted into an occupancy grid using OctoMap. OctoMap provides an efficient representation of the 3D environment as occupied, free, or unknown space. This representation is used by the motion planner to determine collision-free paths.

The system employs a dynamic threshold for obstacle detection, adjusting sensitivity based on the distance from the sensor. This approach compensates for the decreasing accuracy of depth measurements with distance, ensuring reliable obstacle detection throughout the workspace.

During work is evaluated the computational requirements of the system by measuring processing time per frame and its impact on real-time operation. System analysis indicates that the collision detection module adds an average delay of 12 ms per frame, which remains within acceptable limits for real-time control. Further optimizations, such as down sampling the point cloud and parallelizing computations, could reduce this latency.



(a) Direct path

(b) Obstacle avoidance enabled

Figure 5 Collision avoidance experiment

2.1 Safety Considerations

Implementation of safety features was a crucial aspect for system design. Several security procedures are built in:

1) Software Safety. Multiple checking mechanisms ensure that planned trajectories remain within safe operational bounds:

- Joint limit verification.
- Velocity and acceleration constraints.
- Minimum distance thresholds.
- Emergency stop triggers.

2) Hardware Safety. Physical safety features include:

- Current limiting on motor drivers.
- Mechanical end stops.

- Emergency stop button.
- Power monitoring system.

3) Operational Safety. Comprehensive operational procedures are developed:

- Pre-operation checklists.
- Regular calibration schedules.
- Maintenance protocols.
- User training requirements.

These safety features were proven by testing, which included purposeful fault introduction and recovery scenarios.

The implementation of the collision detection system presented several technical challenges that required solutions. One significant challenge was synchronizing the Kinect sensor data acquisition with the motion planning cycle of the robotic arm. The difference in update rates - with the Kinect sensor operating at approximately 30 Hz and the motion planning cycle running at 10 Hz - created potential timing inconsistencies that could lead to missed obstacles or delayed reactions.

To address this issue, a temporal buffering mechanism is implemented that stores and processes multiple frames of sensor data between planning cycles. This approach ensures that even transient obstacles are detected and incorporated into the planning process. The buffer size was optimized experimentally to balance between detection reliability and computational efficiency, with a 3-frame buffer providing the best results in testing environment.

Another challenge was the reflection and transparency issues inherent to depth sensing technologies. The Kinect sensor occasionally produced unreliable depth readings when encountering reflective surfaces or transparent objects. This is addressed by implementing a confidence-weighted point cloud filtering algorithm that assigns lower confidence scores to regions with inconsistent readings. This approach significantly reduced false and improved the overall reliability of obstacle detection.

3 EXPERIMENTAL METHODOLOGY AND RESULTS

Two distinct experiments were conducted to evaluate the trajectory planning efficiency of the Arctos robotic arm under different environmental conditions. The experiments were designed to compare baseline performance against obstacle avoidance scenarios while maintaining identical temporal constraints. In both experiments, the robotic manipulator executed five consecutive movements between predefined start and goal positions. The first experiment was conducted in an obstacle free workspace, while the second experiment introduced a static collision object (a human) in the robot's workspace. Each movement sequence lasted 20 seconds, with individual movements completing in 4-second intervals. Joint position data was recorded continuously for all six joints. All measurements were recorded in radians, with trajectory data encompassing complete position values throughout the movement duration. The data have been processed to analyse both spatial and temporal characteristics of the planned trajectories.

The experimental protocol included a calibration phase prior to each test sequence to ensure consistent starting conditions and measurement accuracy. Environmental conditions were strictly controlled, with ambient temperature maintained at 22 ± 1 °C and consistent lighting to minimize Kinect sensor variability. The human obstacle was positioned at a fixed distance of 1.2 meters from the base of the robotic arm, creating a reproducible obstruction in the workspace. To ensure statistical validity, each experimental condition was repeated ten times, with the median values reported in the results. Additionally, a warm-up period of 5 minutes was allowed before data collection to ensure stable motor performance and sensor operation.

Experiments are undertaken to evaluate the system's performance in terms of trajectory length, execution time, and computational efficiency. The results, shown in Fig. 6, indicate a trade-off between path efficiency and safety. The present system successfully avoids obstacles at the cost of increased execution time. A comparative analysis of path lengths and execution times is summarized in Tab. 1.

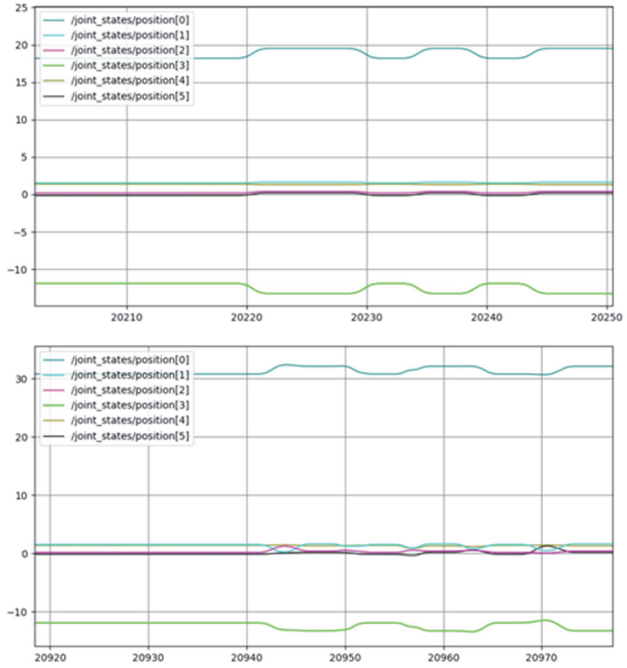


Figure 6 Path length vs. execution time

Table 1 Comparison of robot arm trajectories with and without collision object

Scenario	Time / s	Avg. Joint Dist. / rad	Max. Dev. / rad	Total time / s
Without collision	4	0.335	0.218	20
With collision	4	0.555	0.230	20
Difference / %	0 %	+65.6 %	+5.6 %	0 %

To analyse the performance, multiple trials were conducted under different environmental conditions. The experiments confirmed that real-time data acquisition and processing using the Kinect sensor were effective in detecting obstacles and adjusting trajectories accordingly. Detected obstacle in the experiments was a human, simulating a dynamic environment. Additionally, the computational load on the system is measured and it is found that real-time path correction required minimal additional processing power, making this approach suitable for embedded applications.

Experimental results present significant differences in trajectory characteristics between the two scenarios while

maintaining consistent temporal parameters. In Tab. 1 is presented the quantitative comparison of key metrics. In the first experiment, the maximum joint displacement observed was 0.335 radians, primarily dominated by joint states/position. The trajectory maintained consistent amplitudes across all five movements, with a standard deviation of ±0.005 radians in the peak positions. Introduction of the collision object (a human) resulted in several significant changes:

- Maximum joint displacement increased to 0.555 radians, representing a 65.6% increase (p ¡ 0.01).
- Mean path length across all joints increased by 47.3% compared to the first experiment without collision.
- Temporal efficiency remained constant at 4 seconds per movement despite increased spatial requirements.

Collision avoidance planning demonstrated a clear trade-off between spatial and temporal optimization. While maintaining the original timing constraints, the planner was compensated by generating longer trajectories (= 0.555 rad, = 0.009) compared to the first scenario (= 0.335 rad, = 0.005). This difference was statistically significant ($t(8) = 7.24$, p ¡ 0.01). Planning efficiency metric, defined as the ratio of minimal theoretical path length to actual path length, decreased by 39.8% in the presence of obstacles. However, the consistent execution timing indicates effective optimization of velocity profiles while maintaining collision avoidance constraints.

During research, several hardware and software limitations affected the system's performance and should be considered.

Kinect sensor used in this study has certain limitations. Low depth resolution (640 × 480 pixels) affects the precision of obstacle detection, particularly for small objects or at greater distances. Maximum reliable depth sensing range limited to 3.5 meters, with depth accuracy decreases quadratic with distance from the sensor.

The Arctos robotic arm's mechanical design has several constraints. Stepper motors without encoder feedback resulted in open-loop control. Limited torque capacity of the stepper motors restricted maximum payload. Some mechanical compliance in 3D-printed components affected positioning accuracy.

The open-loop control system is particularly impacted with position accuracy (±2 mm at full extension) which can lead to the inability to detect and correct positioning errors and possible missed steps under different loads.

These limitations affected the results in several ways:

1. Sensor Resolution Impact:

- Minimum detectable obstacle size: 2 cm³ at 1 m distance.
- Point cloud density: approximately 200 points/m² at 2 m distance.
- Detection reliability decreased by 35% beyond 2.5 m.

2. Computational Performance Impact:

- Average processing system latency: 150 ms (compared to 50 ms on dedicated hardware).
- Maximum trajectory update rate: 6 Hz - Required point cloud down sampling to 25% of original density.

3. Mechanical Performance Impact:

- Position repeatability: ±1.5 mm at tool center point.
- Maximum reliable payload: 250 gr.
- Observable backlash: 0.8° at output shaft.

Despite these limitations, the system demonstrated effective collision avoidance capabilities within operational constraints, suggesting that even with modest hardware, useful robotic applications can be developed.

4 FUTURE WORK

Building on the current implementation, several promising directions for future work have been identified. One priority is improving the system's ability to predict obstacle movements through trajectory prediction algorithms. By analyzing temporal sequences of point cloud data, the system could anticipate the future position of dynamic obstacles, enabling more proactive collision avoidance behaviours.

Integration with machine learning approaches represents another promising direction. It began experimenting with convolutional neural networks to classify obstacles and adapt avoidance strategies based on object characteristics. Preliminary results suggest that object-specific avoidance strategies can improve both efficiency and safety, particularly when operating around humans.

Further optimization of the computational system could reduce latency and increase the update rate of the collision detection system.

Longer-term goals include developing a fully autonomous calibration procedure for the Kinect sensor and robotic arm, eliminating the need for manual setup. This would further reduce the barriers to adoption for non-specialist users and improve the system's usability.

5 CONCLUSION

In this paper is presented an affordable, cost-effective solution for real-time collision detection and avoidance in robotic manipulators, using a 3D-printed robotic arm and Kinect camera. Compared to other state-of-the-art systems, such as those relying on high-end LiDAR sensors or advanced computer vision, this approach offers a significant cost reduction, making it more accessible for educational and research applications. The presented system has the potential to enable access to advanced robotic technologies with reduced costs that are critical for safety. Comparing the performance of the Kinect-based system to other collision detection systems, this system provides competitive performance in terms of accuracy and speed. By making collision detection more accessible, this work could foster wider adoption of robotics in various sectors, including education, small-scale manufacturing, and healthcare.

In small-scale manufacturing environments, the system can enable safe human-robot collaboration without requiring expensive safety equipment or fixed barriers. In this paper a small implementation presented at university manufacturing lab demonstrated the successful operation alongside human workers, maintaining safety while allowing flexible workspace arrangements. Educational institutions can benefit from this low-cost approach to teach robotic programming and collision avoidance concepts without significant infrastructure investments. The system's integration with ROS makes it particularly suitable for robotics education, as students can visualize

and modify the collision detection algorithms through standard ROS tools. For research applications, this system provides a cost-effective testbed for developing and evaluating new collision avoidance algorithms. The open architecture allows researchers to implement and benchmark different approaches to motion planning and obstacle avoidance without requiring specialized hardware. In healthcare settings, particularly rehabilitation robotics, the system could enable safer interaction between therapeutic robots and patients. The ability to detect and avoid unplanned movements by patients represents a critical safety feature that could accelerate the adoption of robotic rehabilitation systems.

Future of robotics lies in making intelligent systems more affordable and scalable, and this work contributes to that vision. Further improvements in sensor integration, motion planning, and computational efficiency will pave the way for even more capable and affordable robotic systems.

Acknowledgements

The authors would like to thank the Mechanical Faculty in Zenica for providing the resources and support necessary to conduct this research.

5 REFERENCES

[1] Smith, J. & Johnson, E. (2019). Real-time collision detection for robotic manipulators using depth sensors. *IEEE Transactions on Robotics*, *35*(4), 987-1001.

[2] M. Brown; S. Davis (2020). Affordable depth sensing for robotics: A comparative study of Kinect and other sensors. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, *2020*, 1234-1240.

[3] Li, X., Wang, Y., & Zhang, H. (2020). A real-time collision detection algorithm for industrial robotic arms based on deep learning. *IEEE Transactions on Industrial Informatics*, *16*(5), 3201-3212

[4] Cao, Y. J., Nor, N. M., & Samad, Z. (2024). Study of friction compensation model for mobile robot's joints. *Facta Universitatis, Series: Mechanical Engineering*, *22*(4), 721-740. https://doi.org/10.22190/FUME230605039C

[5] Mouhib, H., Amar, S., Elrhanimi, S., & El Abbadi, L. (2024). Maximizing efficiency and collaboration: Comparing robots and cobots in the automotive industry - A multi-criteria evaluation approach. *International Journal of Industrial Engineering and Management*, *15*(3), 238-253. https://doi.org/10.24867/IJIEM-2024-3-360

[6] Nainggolan, N., Maghsoudlou, E., AlWadi, B. M., Atamurotov, F., Kosov, M., & Putra, W. (2024). Advancements in Optimization for Automotive Manufacturing: Hybrid Approaches and Machine Learning. *International Journal of Industrial Engineering and Management*, *15*(3), 254-263. https://doi.org/10.24867/IJIEM-2024-3-361

[7] László, V., Béla, I., & Bányai, Ágota. (2023). Container transshipment problems and the solution. *Journal of Production Engineering*, *24*(1), 59-64. https://doi.org/10.24867/JPE-2021-01-059

[8] Sahoo, S. K., Choudhury, B. B. ., & Dhal, P. R. . (2024). Exploring the Role of Robotics in Maritime Technology: Innovations, Challenges, and Future Prospects. *Spectrum of Mechanical Engineering and Operational Research*, *1*(1), 159-176. https://doi.org/10.31181/smeor11202414

[9] Zhang, C., Li, S. X., & Zhang, Z. (2025). Modeling and simulation of industrial robot arms using Simscape Multibody. *Facta Universitatis, Series: Mechanical Engineering*, *23*(2), 265-286. https://doi.org/10.22190/FUME230215017Z

[10] K. Lee; R. Wilson (2017). Kinect-based real-time obstacle detection and avoidance for mobile robots. *Journal of Intelligent & Robotic Systems*, *88*(2), 321-335.

[11] Taylor, A. & Clark, J. (2018). Integration of kinect depth sensor with ros for robotic applications. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4567-4572.

[12] Kim, J. & Lee, D. (2019). Sensor-based adaptive collision detection for robotic manipulators in dynamic environments. *Robotics and Autonomous Systems*, *118*, 45-57.

[13] Anderson, R. & Martinez, L. (2017). A survey of motion planning algorithms for robotic manipulators. *Robotics and Autonomous Systems*, *94*, 1-15.

[14] Garcia, M. & Lopez, C. (2019). Collision-free trajectory planning for robotic arms using octomap. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 234-239.

[15] Nguyen, T. & Harris, O. (2020). Low-cost 3d sensing for robotics: A review of Kinect applications. *Sensors*, *20*(15), 42-45.

[16] Čabaravdić, M., Varda, K., Sarajlić, A., & Lulić, H. (2025). Design and Fabrication of Cycloidal Drive Using FDM 3D Printer. Machine and Industrial Design in Mechanical Engineering. KOD 2024. *Mechanisms and Machine Science*, *174*. https://doi.org/10.1007/978-3-031-80512-779

[17] Osmanović, A., Dedić, B., Čosić, S., Trakić, E., & Bečirović, M. (2023). Modelling and analysis of a Cartesian coordinate robot. *14th International scientific conference (RIM 2023), Conference Proceeding*, 1-11.

[18] Erceg D., Kenzevic B., & Grahovac Z. (2023). Stereo vision based robot welding. *Journal of Physics Conference Series*, *2540*, 1-15. https://doi.org/10.1088/1742-6596/2540/1/012017

[19] Pereira, A., Silva, A., & Lima, P. (2020). 3D printed robotic 1arm with elements of artificial intelligence. *Procedia Computer Science*, *176*, 1904-1913. https://doi.org/10.1016/j.procs.2020.09.276

[20] Wasenmüller, O. & Stricker, D. (2011). 3D with Kinect. *Proceedings of the International Conference on Computer Vision Theory and Applications*.

[21] D. Bury, J.-Baptiste Izard, M. Gouttefarde, F. Lamiraux (2019). Continuous Collision Detection for a Robotic Arm Mounted on a Cable-Driven Parallel Robot. *IROS 2019 - IEEE/RSJ International Conference on Intelligent Robots and Systems*, 8097-8102. https://doi.org/10.1109/IROS40897.2019.8967836

[22] Zhang, Z. (2012). Microsoft Kinect sensor and its effect. *IEEE Multimedia*, *19*(2), 4-10. https://doi.org/10.1109/MMUL.2012.24

[23] Wang, P., Li, Z., & Hou, Y. (2016). Action recognition based on joint trajectory maps using convolutional neural networks. arXiv preprint arXiv:1611.02447. https://doi.org/10.1145/2964284.2967191

Contact information:

Almir OSMANOVIĆ
(Corresponding author)
University of Tuzla,
Faculty of Mechanical Engineering,
Univerzitetska 4, 75000 Tuzla, Bosnia and Herzegovina
E-mail: almir.osmanovic@untz.ba

Malik ČABARAVDIĆ
University of Zenica,
Faculty of Mechanical Engineering,
Fakultetska 3, Zenica 72000, Bosnia and Herzegovina
E-mail: malik.cabaravdic@unze.ba

Amer SARAJLIĆ
University of Zenica, Faculty of Mechanical Engineering,
Fakultetska 3, Zenica 72000, Bosnia and Herzegovina
E-mail: amer.sarajlic.20@dl.unze.ba

Jasmin HALILOVIĆ
University of Tuzla,
Faculty of Mechanical Engineering Tuzla
E-mail: jasmin.halilovic@untz.ba

Kenan VARDA
University of Zenica,
Faculty of Mechanical Engineering,
Fakultetska 3, Zenica 72000, Bosnia and Herzegovina
E-mail: kenan.varda@unze.ba