DOI: https://doi.org/10.12700/btsz.2025.7.1.1

AI-Driven Control System for Safe and Adaptive Human-Robot Collaboration in Welding Applications

AI-TÁMOGATOTT VEZÉRLŐRENDSZER A BIZTONSÁGOS ÉS ADAPTÍV EMBER-ROBOT EGYÜTTMŰKÖDÉSHEZ HEGESZTÉSI ALKALMAZÁSOKBAN

AICHAOUI Nada El Yasmine[1] – KOVÁCS Tünde Anna[2]

Abstract

The study presents a system to support safe collaboration in a human-robot welding environment. The welding parameters change dynamically according to the evaluation of real-time sensed environmental data by an artificial intelligence algorithm. The system follows the movement of the human working in a collaborative environment, takes into account the UV danger zone and integrates the welding robot's operation. Simulations show that the system effectively minimises risks without compromising weld quality. The algorithm developed aims to reduce the risk of human-robot collaboration in the field of occupational health and safety by evaluating real-time environmental data with AI support, while meeting the quality requirements of welding. Currently, it can't find a similar innovation with the presented AI-supported system in the welding industry.

Absztrakt

Az ember-robot hegesztési környezetben történő biztonságos együttműködést támogató rendszert mutat be a tanulmány. A hegesztési paraméterek dinamikusan változnak a valós időben érzékelt környezeti adatok mesterséges intelligencia algoritmus értékelése és szerint. A rendszer követi a kollaboratív környezetben dolgozó ember mozgását figyelembe veszi az UV veszélyzónát és ehhez integrálja a hegesztő robot működését. Szimulációk igazolják, hogy a rendszer hatékonyan minimalizálja a kockázatokat a hegesztés minőség romlása nélkül. A kifejlesztett algoritmus a human-robot együttműködés munkavédelmi kockázatát kívánja csökkenteni a valós idejű környezeti adatok AI támogatással történő értékelése alapján a hegesztés minőségi követelményeinek kielégítése mellett. Jelenleg a hegesztési ipari gyakorlatban még nem található a bemutatott fejlesztéshez hasonló AI támogatott rendszer.

Keywords

Welding Robotics, Collaborative Environment, Real-time Sensor, Artificial Intelligence (AI), Risk Assessment, UV Zone

Kulcsszavak

Robotos hegesztés, kollaboratív környezet, valós idejű érzékelő, mesterséges intelligencia (AI), kockázatértékelés, UV zóna

[1] nada.aichaoui@phd.uni-obuda.hu | ORCID: 0009-0008-4361-675X | PhD candidate, Doctoral School on Safety and Security Sciences, Óbuda University | PhD hallgató, Biztonságtudományi Doktori Iskola, Óbudai Egyetem

[2] kovacs.tunde@bgk.uni-obuda.hul | ORCID: 0000-0002-5867-5882 | professor, Bánki Donát Faculty of Mechanical and Safety Engineering, Óbuda University | egyetemi tanár, Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar, Óbudai Egyetem

INTRODUCTION

The welding robot has control systems that reflect a significant evolution toward advanced automation and precision [1]. Traditional control systems relied heavily on predefined trajectories and fixed parameters, limiting their adaptability to variations in welding conditions and workpiece geometries [2], [3]. This approach, while functional in controlled environments, often fell short when faced with unpredictable factors or complex shapes, highlighting the limitations in addressing diverse industrial needs [3], [4]. Consequently, the current state of welding technology lacks extensive integration of advanced algorithms for optimizing parameters and identifying defects, which is evident from limited accessible information and practical implementation gaps [5], [6].

However, recent advancements have revolutionized this landscape [7]. Modern welding robot control systems incorporate sophisticated sensor technologies such as vision systems, laser scanners, and force/torque sensors, enabling real-time feedback and adaptive control [8], [9], [10]. These systems leverage advanced algorithms, including machine learning and artificial intelligence [11], [12], to optimize welding parameters and trajectories dynamically. Such adaptability ensures that welding processes are not only precise but also resilient to variations in environmental and operational conditions [13], [14]. Furthermore, cobots with advanced safety features have emerged as a prominent trend, allowing for human-robot collaboration in welding tasks [15], [16], [17]. These cobots are equipped with sensors and safety protocols that minimize risks and enhance productivity, making them increasingly viable for tasks requiring human oversight or intervention [12].

Overall, the current state-of-the-art of welding robot control systems emphasizes flexibility, efficiency, and quality, paving the way for increased productivity and competitiveness in industries reliant on welding technology [18]. The integration of adaptive algorithms, real-time feedback, and collaborative capabilities ensures that welding processes are not only efficient but also safe and reliable, setting a new standard for automation in this domain [19], [20].

This paper will outline the simulation environment used, describe the modeling of the welding process, and provide an analysis of safety testing. The results from the simulations serve as an initial validation of the control system's performance, offering insights into its real-world applicability and identifying areas for further optimization. By providing a foundation for testing safety and collaborative workflows, simulation ensures the robustness and reliability of the control system before it is introduced into industrial welding applications.

Integrative Control with Operator Interaction and Safety

The current state of welding technology still lacks significant incorporation of advanced algorithms for optimizing parameters and detecting defects, as evidenced by the limited amount of readily available information [21], [22]. However, recent advancements have shown promising progress in harnessing neural networks to automate these processes, even though such developments are still in the early stages [23], [24], [25]. These advancements aim to improve the precision and efficiency of welding operations while reducing human intervention and error [26].

One of the primary goals in this field is to gain a deeper understanding on how welding robots operate, especially in scenarios where an object, often a person, accidentally enters the robot's hazardous zone during its work as shown in Figure 1. This understanding is essential to ensure both safety and productivity in industrial settings. In typical situations, the main risks emerge when humans approach the robot's danger zones while it is actively functioning. This underscores the need for a thorough comprehension of the robot's response mechanisms and the application of effective safety protocols to avoid accidents and injuries.

In industrial welding environments, the danger zone is often defined by the spread of ultraviolet (UV) radiation emitted during the welding process [27]. This radiation poses serious health risks to humans, particularly with prolonged exposure, and can lead to severe harm if safety measures are not properly enforced [28], [29]. A critical challenge occurs when a person inadvertently enters the robot's hazardous UV zone during operation. While stopping the robot immediately can prevent harm, it may also interrupt vital welding tasks, causing production delays or material waste.

To address this issue, the following study proposes the development of an intelligent control system designed to prioritize human safety while enabling adaptive decision-making. This system will assess risks in real-time and make decisions that balance the need for safety with the demands of the welding process. By integrating advanced technologies and algorithms, the solution aims to create safer, more efficient industrial environments that can respond dynamically to potential hazards without compromising productivity.

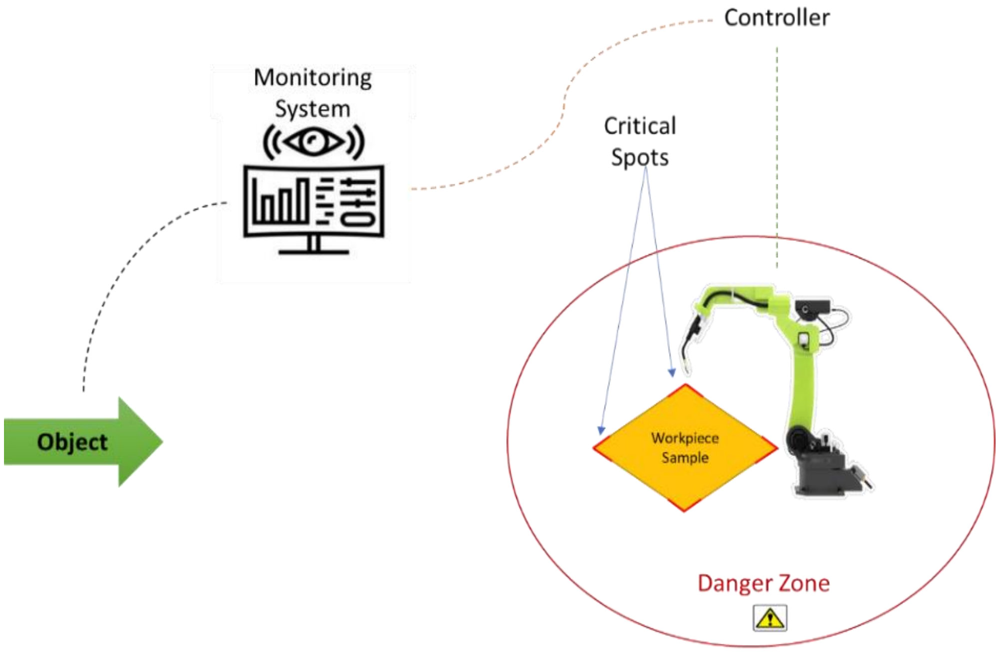

Figure 1. The general implementation of an object approaching to welding robot danger zone.

The implementation of the safety and quality requirement should pass as the flowchart in Figure 2 , where this system starts when a human comes closer to the danger zone, the monitoring system shall detect this action and provide an immediate response relative to the welding robot control system

The proposed system detects human presence in the UV zone of the welding robot, assesses the risk associated with the presence, and makes an intelligent decision based on the risk level and the stage of the welding process. The general scenario follows these steps:

- **Human Detection:** The system detects when a human accidentally enters the dangerous UV zone.
- **Risk Assessment:** The AI system evaluates the risk to human health:
    - If the risk level is unacceptable, the welding robot is immediately stopped, and the human is instructed to exit the zone, and for the next step and because the welding status in this case is unknown, it is necessary to relaunch or resume the process manually by human-operator who cooperate to evaluate the operated sample.
    - If the risk level is acceptable, the system further evaluates the welding process.
- **Welding Process Evaluation:**
    - If the robot is working at a normal point in the welding process (where stopping the process does not affect the quality), the robot is stopped, and the human is instructed to exit, in this case, since the workpiece is in known status, the next step, the robot will relaunch or resume the process automatically.
    - If the robot is working at a critical point (where stopping the process could damage the sample), the robot continues welding. Simultaneously, warnings are issued to the human.
- **Timeout Mechanism:**

If the human exits the UV zone before a preset timeout, the welding continues, and the robot moves to the next operation after finishing the current one.

If the human remains in the zone beyond the timeout, the AI reassesses the risk level. If the risk level becomes unacceptable, the robot is stopped. If it remains acceptable, the robot continues the welding process.

The following presented workflow in 2. Figure ensures that safety protocols are followed while minimizing downtime or wastage during critical welding operations. The behavior of this workflow was modeled using Python. The scenarios outlined in the research were implemented as follows:

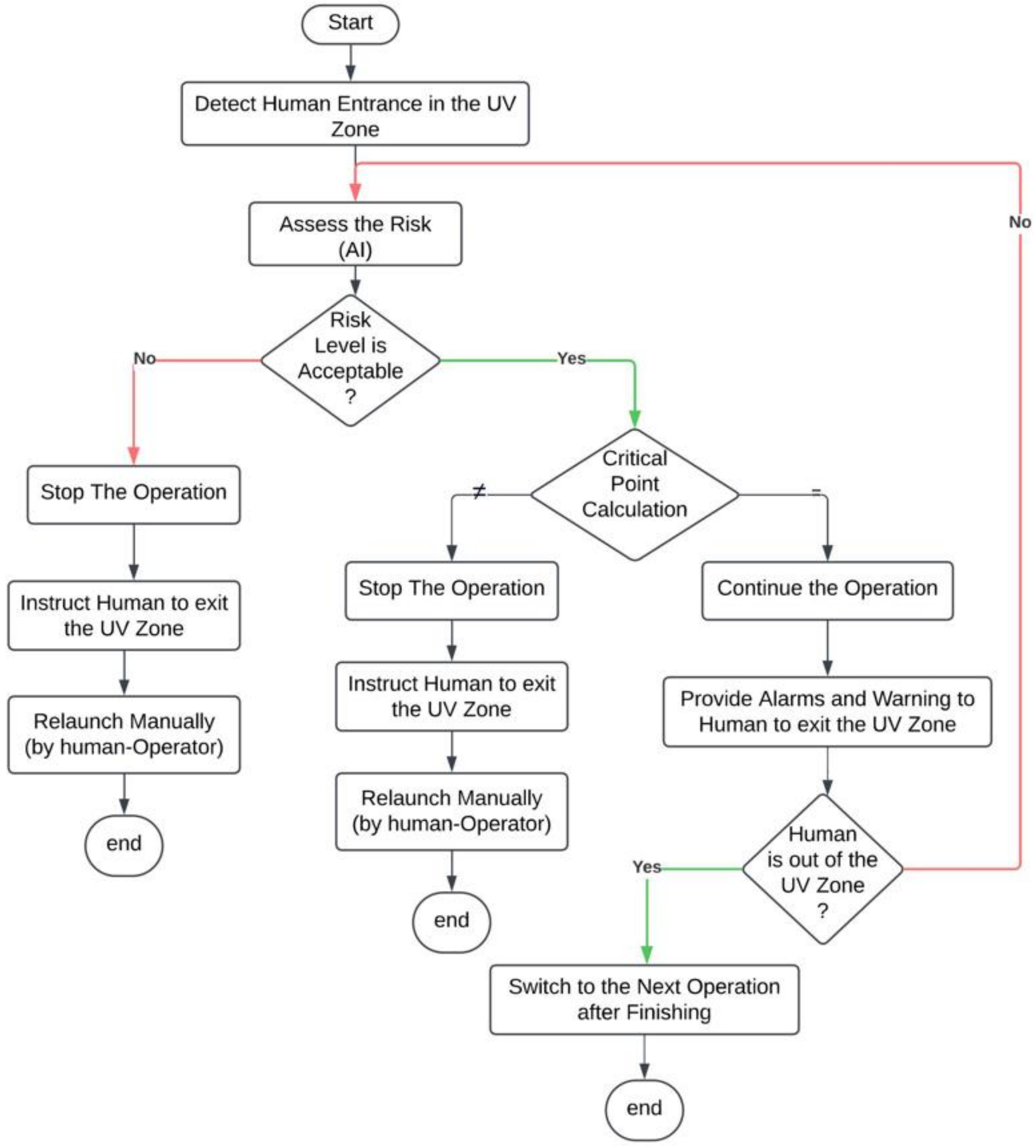

Figure 2. Workflow for Safety Control in Human-Robot Collaborative Welding Environments

The behavior of this workflow was modeled using Python. The four scenarios outlined in the research were implemented as follows:

1. **Scenario 1:** The system detects a human, assesses an unacceptable risk level, and immediately stops the welding robot. The human is instructed to exit the UV zone, and the system waits for manual relaunch. The script is as follows in Figure 3 and the result of the simulation appears in Figure 4:

```python
#Python Code Scenaio 1:
def detect_human_entry():
    # Simulate detection of human entry in the UV zone : Return True if human is detected, False otherwise.
    return True  # Human is detected
def assess_risk():
    # Simulate AI assessing the risk level : Return True if risk level is acceptable, False otherwise.
    return False  # Risk level is not acceptable
def determine_welding_point():
    # Simulate determining if the welding point is normal or critical : Return True if welding at a normal point, False if at a critical point.
    return True  # This function will not be called in this scenario
def does_human_exit_before_timeout():
    # Simulate checking if the human exits the UV zone before timeout : Return True if human exits before timeout, False otherwise.
    return True  # This function will not be called in this scenario
def continue_welding():
    # Simulate continuing the welding operation.
    print("Welding continues.")  # This function will not be called in this scenario
def send_stop_signal():
    # Simulate sending a stop signal to the robot.
    print("Stop signal sent to the robot.")
def instruct_human_to_exit():
    # Simulate instructing the human to exit the UV zone.
    print("Human instructed to exit UV zone.")
def wait_for_manual_relaunch():
    # Simulate waiting for manual relaunch of the robot.
    print("Waiting for manual relaunch.")
def wait_for_automatic_relaunch():
    # Simulate waiting for automatic relaunch of the robot.
    print("Waiting for automatic relaunch.")
def switch_to_next_operation():
    # Simulate switching to the next operation after finishing.
    print("switching to the next opperation after finishing")
def reassess_risk():
    # Simulate AI reassessing the risk level.
    print("Reassess the risk.")
def main():
    # Start of the workflow
    print("Workflow started.")
    if detect_human_entry():
        if not assess_risk():  # Risk level is not acceptable
            send_stop_signal()
            instruct_human_to_exit()
            wait_for_manual_relaunch()
        else:  # Risk level is acceptable
            if determine_welding_point():  # Welding at a normal point
                send_stop_signal()
                instruct_human_to_exit()
                wait_for_manual_relaunch()
            else:  # Welding at a critical point
                continue_welding()
                if does_human_exit_before_timeout():  # Human exited before timeout
                    continue_welding()
                else:  # Human did not exit before timeout, reassess risk level
                    if not assess_risk():  # Risk level is not acceptable after reassessment
                        send_stop_signal()
                        instruct_human_to_exit()
                        wait_for_manual_relaunch()
                    else:  # Risk level is acceptable after reassessment
                        continue_welding()
if __name__ == "__main__":
    main()
```

Figure 3. Python Script of Scenario 1

```
Workflow started.
Stop signal sent to the robot.
Human instructed to exit UV zone.
Waiting for manual relaunch.
```

Figure 4. Scenario 1 Response Result

2. **Scenario 2:** The system detects a human, assesses an acceptable risk level, and detects that the robot is working at a normal point (Figure 5). The robot is stopped, and the human is instructed to exit. The system then waits for automatic relaunch (Figure 6).

```python
#Python Code Scenaio 2:
def detect_human_entry():
    # Simulate detection of human entry in the UV zone: Return True if human is detected, False otherwise.
    return True  # Human is detected
def assess_risk():
    # Simulate AI assessing the risk level : Return True if risk level is acceptable, False otherwise.
    return True  # Risk level is acceptable
def determine_welding_point():
    # Simulate determining if the welding point is normal or critical : Return True if welding at a normal point, False if at a critical point.
    return True  # Welding at a normal point
def does_human_exit_before_timeout():
    # Simulate checking if the human exits the UV zone before timeout : Return True if human exits before timeout, False otherwise.
    return True  # This function will not be called in this scenario
def continue_welding():
    # Simulate continuing the welding operation.
    print("Welding continues.")  # This function will not be called in this scenario
def send_stop_signal():
    # Simulate sending a stop signal to the robot.
    print("Stop signal sent to the robot.")
def instruct_human_to_exit():
    # Simulate instructing the human to exit the UV zone.
    print("Human instructed to exit UV zone.")
def wait_for_manual_relaunch():
    # Simulate waiting for manual relaunch of the robot.
    print("Waiting for manual relaunch.")
def wait_for_automatic_relaunch():
    # Simulate waiting for automatic relaunch of the robot.
    print("Waiting for automatic relaunch.")
def switch_to_next_operation():
    # Simulate switching to the next operation after finishing.
    print("switching to the next opperation after finishing")
def reassess_risk():
    # Simulate AI reassessing the risk level.
    print("Reassess the risk.")
def main():
    # Start of the workflow
    print("Workflow started.")
        if detect_human_entry():
        if not assess_risk(): # Risk level is not acceptable
            send_stop_signal()
            instruct_human_to_exit()
            wait_for_manual_relaunch()
        else # Risk level is acceptable
            if determine_welding_point(): # Welding at a normal point
                send_stop_signal()
                instruct_human_to_exit()
                wait_for_automatic_relaunch()
            else: # Welding at a critical point
                continue_welding()
                if does_human_exit_before_timeout(): # Human exited before timeout
                    continue_welding()
                else: # Human did not exit before timeout, reassess risk level
                    if not assess_risk(): # Risk level is not acceptable after reassessment
                        send_stop_signal()
                        instruct_human_to_exit()
                        wait_for_manual_relaunch()
                    else: # Risk level is acceptable after reassessment
                        continue_welding()
if __name__ == "__main__":
    main()
```

Figure 5. Python Script of Scenario 2

```
Workflow started.
Stop signal sent to the robot.
Human instructed to exit UV zone.
Waiting for automatic relaunch.
```

Figure 6. Scenario 2 Response Result

3. **Scenario 3**: The system detects a human, assesses an acceptable risk level, and detects that the robot is working at a critical point (Figure 7). The robot continues working while warning the human. If the human exits the zone before the timeout, the robot continues welding (Figure 8).

```python
#Python Code Scenaio 3:
def detect_human_entry():
    # Simulate detection of human entry in the UV zone : Return True if human is detected, False otherwise.
    return True  # Human is detected
def assess_risk():
    # Simulate AI assessing the risk level : Return True if risk level is acceptable, False otherwise.
    return True  # Risk level is acceptable
def determine_welding_point():
    # Simulate determining if the welding point is normal or critical : # Return True if welding at a normal point, False if at a critical point.
    return False  # Welding at a critical point
def does_human_exit_before_timeout():
    # Simulate checking if the human exits the UV zone before timeout : Return True if human exits before timeout, False otherwise.
    return True  # Human exits before timeout
def continue_welding():
    # Simulate continuing the welding operation.
    print("Welding continues.")
def send_stop_signal():
    # Simulate sending a stop signal to the robot.
    print("Stop signal sent to the robot.")
def instruct_human_to_exit():
    # Simulate instructing the human to exit the UV zone.
    print("Human instructed to exit UV zone.")
def wait_for_manual_relaunch():
    # Simulate waiting for manual relaunch of the robot.
    print("Waiting for manual relaunch.")
def wait_for_automatic_relaunch():
    # Simulate waiting for automatic relaunch of the robot.
    print("Waiting for automatic relaunch.")
def switch_to_next_operation():
    # Simulate switching to the next operation after finishing.
    print("switching to the next opperation after finishing")
def reassess_risk():
    # Simulate AI reassessing the risk level.
    print("Reassess the risk.")
def main():
    # Start of the workflow
    print("Workflow started.")
    if detect_human_entry():
        if not assess_risk(): # Risk level is not acceptable
            send_stop_signal()
            instruct_human_to_exit()
            wait_for_manual_relaunch()
        else: # Risk level is acceptable
            if determine_welding_point(): # Welding at a normal point
                send_stop_signal()
                instruct_human_to_exit()
                wait_for_manual_relaunch()
            else: # Welding at a critical point
                continue_welding()
                if does_human_exit_before_timeout(): # Human exited before timeout
                    instruct_human_to_exit()
                    switch_to_next_operation()
                else: # Human did not exit before timeout, reassess risk level
                    if not assess_risk(): # Risk level is not acceptable after reassessment
                        send_stop_signal()
                        instruct_human_to_exit()
                        wait_for_manual_relaunch()
                    else: # Risk level is acceptable after reassessment
                        continue_welding()
if __name__ == "__main__":
    main()
```

Figure 7. Python Script of Scenario 3

```
Workflow started.
Welding continues.
Human instructed to exit UV zone.
switching to the next operation after finishing
```

Figure 8. Scenario 3 Response Result

4. **Scenario 4**: The system detects a human, assesses an acceptable risk level, and detects that the robot is working at a critical point. Human do not exist before the timeout (Figure 9), triggering a reassessment of the risk. If the risk becomes unacceptable, the robot is stopped; otherwise, it continues welding (Figure 10).

```python
#Python Code Scenaio 4:
def detect_human_entry():
    # Simulate detection of human entry in the UV zone : Return True if human is detected, False otherwise.
    return True  # Human is detected
def assess_risk():
    # Simulate AI assessing the risk level : Return True if risk level is acceptable, False otherwise.
    return True  # Initial risk level is acceptable
def determine_welding_point():
    # Simulate determining if the welding point is normal or critical : Return True if welding at a normal point, False if at a critical point.
    return False  # Welding at a critical point
def does_human_exit_before_timeout():
    # Simulate checking if the human exits the UV zone before timeout : Return True if human exits before timeout, False otherwise.
    return False  # Human does not exit before timeout
def continue_welding():
    # Simulate continuing the welding operation.
    print("Welding continues.")
def send_stop_signal():
    # Simulate sending a stop signal to the robot.
    print("Stop signal sent to the robot.")
def instruct_human_to_exit():
    # Simulate instructing the human to exit the UV zone.
    print("Human instructed to exit UV zone.")
def wait_for_manual_relaunch():
    # Simulate waiting for manual relaunch of the robot.
    print("Waiting for manual relaunch.")
def wait_for_automatic_relaunch():
    # Simulate waiting for automatic relaunch of the robot.
    print("Waiting for automatic relaunch.")
def switch_to_next_operation():
    # Simulate switching to the next operation after finishing.
    print("switching to the next opperation after finishing")
def reassess_risk():
    # Simulate AI reassessing the risk level.
    print("Reassess the risk.")
def main():
    # Start of the workflow
    print("Workflow started.")
    if detect_human_entry():
        if not assess_risk(): # Initial risk level is not acceptable
            send_stop_signal()
            instruct_human_to_exit()
            wait_for_manual_relaunch()
        else: # Initial risk level is acceptable
            if determine_welding_point(): # Welding at a normal point
                send_stop_signal()
                instruct_human_to_exit()
                wait_for_manual_relaunch()
            else: # Welding at a critical point
                continue_welding()
                if does_human_exit_before_timeout(): # Human exited before timeout
                    continue_welding()
                else: # Human did not exit before timeout, reassess risk level
                    if not reassess_risk(): # Risk level is not acceptable after reassessment
                        send_stop_signal()
                        instruct_human_to_exit()
                        wait_for_manual_relaunch()
                    else: # Risk level is acceptable after reassessment
                        continue_welding()
if __name__ == "__main__":
    main()
```

Figure 9. Python Script of Scenario 4

```
Workflow started.
Welding continues.
Reassess the risk.
Stop signal sent to the robot.
Human instructed to exit UV zone.
Waiting for manual relaunch.
```

Figure 10. Scenario 4 Response Result

The Python code for each scenario handles the decision-making logic using conditional checks, simulating the workflow discussed above. The system's operations depend on the real-time input from sensors (simulated in code) and the intelligent assessment by the AI module. Below is a brief explanation of the key components:

- **detect_human_entry():** Simulates the sensor detecting a human entering the UV zone.
- **assess_risk() / reassess_risk():** Simulates the AI evaluating the risk level based on various factors such as distance and exposure.
- determine_welding_point(): Identifies if the welding operation is at a critical or normal stage.
- **does_human_exit_before_timeout():** Simulates the system waiting for the human to exit the dangerous zone.
- **continue_welding() / send_stop_signal() / instruct_human_to_exit():** Functions that simulate the actual responses of the system based on the situation.

This simulation code ensures that all the potential scenarios are covered, where either the system stops the robot for safety reasons or continues the welding process when it is safe to do so.

Results

The safety control system successfully demonstrates a flexible and robust decision-making process that guarantees human safety without significantly hindering the welding process. The integration of AI-based risk assessment with real-time monitoring provides an efficient solution to collaborative human-robot environments in welding applications.

Key Outcomes

- **Safety First:** The system prioritizes human safety in all scenarios, whether the risk is initially acceptable or not. The system ensures that if the risk level is unacceptable, the robot is immediately stopped.
- **Efficiency in Critical Processes:** For critical welding points where stopping would result in damage or waste, the system intelligently allows the robot to continue working while issuing warnings to humans.
- **Adaptive Response:** The system adapts based on real-time inputs and reassessments. If a human stays in the hazardous zone too long, the system re-evaluates the risk dynamically, ensuring continuous safety monitoring.
- **Manual Control:** After a safety stop, the robot can only be relaunched manually because the human operator decides whether the workpiece can be resumed or if it is already wasted where they need to launch another new operation.
- **Automatic Control:** After an efficiency stop, the current welding sample is at a normal stage, where there is no need for human interaction to make the decision, instead the robot relaunches the process by itself.

CONCLUSION

In this research, we developed a safety-first control system for collaborative human-robot work environments in welding applications, integrating Artificial Intelligence (AI) with real-time monitoring and decision-making processes. The system prioritizes human safety while maintaining operational efficiency, particularly in scenarios involving unexpected human intrusions into hazardous zones like the UV area generated by a welding robot. The aim was to strike a balance between safety, quality, and production rate without compromising the welding process. The control system design incorporates AI-driven decision-making, real-time monitoring, and robust safety protocols, with a Python implementation successfully simulating various workflow scenarios and demonstrating the system's adaptability to different risk situations. Future advancements could involve integrating more advanced sensor technologies and AI models to further enhance risk prediction accuracy and overall system reliability.

REFERENCES

[1] J. T. Kahnamouei and M. Moallem, Advancements in control systems and integration of artificial intelligence in welding robots: A review, Ocean Engineering, 2024, vol. 312, pp. 119294.

[2] B. Wang, S. J. Hu, L. Sun and T. Freiheit, Intelligent welding system technologies: State-of-the-art review and perspectives, Journal of Manufacturing Systems, 2020, vol. 56, pp. 373-391.

[3] Q. Guo, Z. Yang, J. Xu, Y. Jiang, W. Wang, Z. Liu, ... and Y. Sun, Progress, challenges and trends on vision sensing technologies in automatic/intelligent robotic welding: State-of-the-art review, Robotics and Computer-Integrated Manufacturing, 2024, vol. 89, pp. 102767.

[4] M. Marecek-Kolibisky, S. Janik, M. Mikva, P. Szabo, and G. Czifra, Human-Machine Co-Working for Socially Sustainable Manufacturing in Industry 4.0, Acta Polytechnica Hungarica, 2024, vol. 21, no. 2.

[5] Y. Cao, Q. Zhou, W. Yuan, Q. Ye, D. Popa and Y. Zhang, Human-robot collaborative assembly and welding: A review and analysis of the state of the art, Journal of Manufacturing Processes, 2024, vol. 131, pp. 1388-1403.

[6] S. I. Wahidi, S. Oterkus and E. Oterkus, Robotic welding techniques in marine structures and production processes: A systematic literature review, Marine Structures, 2024, vol. 95, pp. 103608.

[7] Z. Wang, The active visual sensing methods for robotic welding: review, tutorial and prospect, 2024, arXiv preprint arXiv:2405.00685.

[8] A. Mehta and H. Vasudev, Advances in welding sensing information processing and modelling technology: an overview, Journal of Adhesion Science and Technology, 2024, pp. 1-45.

[9] J. T. Kahnamouei and M. Moallem, Advancements in control systems and integration of artificial intelligence in welding robots: A review, Ocean Engineering, 2024, vol. 312, pp. 119294.

[10] G. D. Putnik, P. B. Petrovic and V. Shah, Spatial Visual Feedback for Robotic Arc-Welding Enforced by Inductive Machine Learning, Journal of Manufacturing Science and Engineering, 2024, vol. 146, no. 4.

[11] A. Hyllbrant C. Janpechpanao, Exploring Machine Learning Approaches for Predicting Stops in Welding Robot Operarions, 2024.

[12] D. A. Suciu, E. H. Dulf, and L. Kovacs, Low-Cost Autonomous Trains and Safety Systems Implementation, using Computer Vision, Acta Polytechnica Hungarica, 2024, vol. 21, no. 9.
[13] Y. Chu, K. Ma, L. Zhao, J. Xu, W. Zhou, X. Wang, ... and Y. Zhang, Structural design and adaptive tracking control of automatic welding robot for liquefied natural gas containment system, Discover Applied Sciences, 2024, vol. 6, no. 3, pp. 118.
[14] M. A. Nasser and M. M. Asy, Virtual numerical control: an approach towards autonomous manufacturing with a case study in welding, The International Journal of Advanced Manufacturing Technology, 2024, pp. 1-19.
[15] S. Kumar, Introductory Chapter: Welding in the Era of Industry 5.0, In Welding-Materials, Fabrication Processes, and Industry 5.0. IntechOpen, 2024.
[16] Kollár, Csaba ; Ványa, László: Szerethetők-e a robotok?: Az ember-robot interakció humán oldalának empirikus aspektusa. HADTUDOMÁNY: A MAGYAR HADTUDOMÁNYI TÁRSASÁG FOLYÓIRATA 27 : 1-2 pp. 163-177. , 15 p. (2017)
[17] Kollár, Csaba: Szerethetők-e a robotok: Az ember-robot interakció humán oldalának teoretikus aspektusa. HADTUDOMÁNY: A MAGYAR HADTUDOMÁNYI TÁRSASÁG FOLYÓIRATA 26 : különszám pp. 142-154. , 13 p. (2016)
[18] Zhang, H, Optimization and Efficiency Improvement of Robot-based Industrial Production Process, International Journal of New Developments in Engineering and Society, 2024, vol. 8, no. 2.
[19] D. K. Naik, V. P. Sharma and R. Dinesh Kumar, Automation in Welding Industries, Automation in Welding Industry: Incorporating Artificial Intelligence, Machine Learning and Other Technologies, 2024, pp. 37-48.
[20] I. M. Sarivan, O. Madsen and B. V. Wæhrens, Automatic welding-robot programming based on product-process-resource models, The International Journal of Advanced Manufacturing Technology, 2024, vol. 132, no. 3, pp. 1931-1950.
[21] A. Biber, R. Sharma, U. Reisgen, Robotic welding system for adaptive process control in gas metal arc welding, Welding in the World, 2024, pp. 1-10.
[22] D. Curiel, F. Veiga, A. Suarez and P. Villanueva, Advances in robotic welding for metallic materials: Application of inspection, modeling, monitoring and automation techniques, Metals, 2023, vol. 13, no. 4, pp. 711.
[23] M. Amarnath, N. Sudharshan and P. Srinivas, Automatic detection of defects in welding using deep learning-a systematic review, Materials Today: Proceedings, 2023.
[24] D. Say, S. Zidi, S. M. Qaisar and M. Krichen, Automated categorization of multiclass welding defects using the x-ray image augmentation and convolutional neural network, Sensors, 2023, vol. 23, no. 14, pp. 6422.
[25] S. Perri, F. Spagnolo, F. Frustaci and P. Corsonello, Welding defects classification through a Convolutional Neural Network, Manufacturing Letters, 2023, vol. 35, pp. 29-32.
[26] S. Ma, Z. Chen, D. Zhang, Y. Du, X. Zhang and Q. Liu, Interpretable Multi-task Neural Network Modeling and Particle Swarm Optimization of Process Parameters in Laser Welding, Knowledge-Based Systems, 2024, pp. 112116.

[27] G. A. Gourzoulidis, C. A. Bouroussis, A. Achtipis, M. Kazasidis, D. Pantelis, A. Markoulis, ... and F. V. Topalis, Photobiological hazards in shielded metal arc welding, Physica Medica, 2023, vol. 106, pp. 102520.

[28] S. S. Murugan and P. Sathiya, Analysis of welding hazards from an occupational safety perspective, Vietnam Journal of Science, Technology and Engineering, vol. 66, no. 3, 2024, pp. 63-74.

[29] A. S. Shote, S. A. Aasa and H. O. Adeyemi, Environmental Trends Due to Arc Welding Activities: Sensitivity of some Typical Workplaces, Arid Zone Journal of Engineering, Technology and Environment, 2024, vol. 20, no. 3, pp. 619-624.