

A Safety Response Mechanism for an Autonomous Moving Robot in a Small Manufacturing Environment using Q-learning Algorithm and Speech Recognition

Sonia KIANGALA (✉ sokiangala@gmail.com)

University of South Africa

Zenghui Wang

University of South Africa <https://orcid.org/0000-0003-3025-336X>

Research Article

Keywords: Autonomous moving robot , Obstacle-free path planning , Q-learning algorithm , Reinforcement learning (RL) , Safety response , Speech recognition

Posted Date: July 23rd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-703829/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

A Safety Response Mechanism for an Autonomous Moving Robot in a Small Manufacturing Environment using Q-learning Algorithm and Speech Recognition

KAHIOMBA Sonia KIANGALA (ORCID: 0000-0003-2994-0699) · Zenghui WANG (ORCID:0000-0003-3025-336X)

Received: date / Accepted: date

Abstract The industrial manufacturing sector is currently undergoing a tremendous revolution moving from traditional production processes to intelligent techniques. Under this revolution, known as Industry 4.0 (I40), a robot is no longer a static equipment but an active workforce to the factory production alongside human operators. Safety becomes crucial for humans and robots to ensure a smooth production run in such environments. Operators are subject to frequent safety inductions to react in emergencies but very little is done for robots. Our research proposes a safety response mechanism for a small manufacturing plant, through which an autonomous robot learns the obstacle-free trajectory to the closest safety exit in emergencies. We implement a reinforcement learning (RL) algorithm, Q-learning, to enable the path learning abilities of the robot. After obtaining the robot optimal path selection options with Q-learning, we code the outcome as a rule-based system for the safety response. We also program a speech recognition system for operators to react timeously, with a voice command, to an emergency that requires stopping all plant activities even when they are far away from emergency stops (ESTOPs) button. The factory emergency signal can be given by an ESTOP or a voice command sent directly to the factory central controller: an S7-1200 Siemens programmable logic controller (PLC) in this experiment. We simulate a simple and small manufacturing environment overview to test our safety procedure. Our results show that the safety response mechanism successfully generates paths without obstacles to the closest safety exits from all the factory locations.

Keywords Autonomous moving robot · Obstacle-free path planning · Q-learning algorithm · Reinforcement learning (RL) · Safety response · Speech recognition

Declarations

Funding: This research is supported partially by South African National Research Foundation Grants (No. 112108 and 132797) and Tertiary Education Support Program (TESP) of South African ESKOM.

Conflicts of interest/Competing interests: The authors declare to the best of their knowledge and believe that there are no conflicts of interest or competing interests in relation with this manuscript.

Availability of data and material: We implemented two types of programming materials: one python code designed in google colab to simulate the experimental manufacturing environment location overview (with Q-learning) and another python code for the speech recognition interface with a S7-1200 PLC. Data utilized is available in the programming code.

Code availability: Both Python codes developed to create the experimental safety mechanism procedure are available.

1 Introduction

Today's manufacturing automation trend, the so-called Industry 4.0 or smart manufacturing, is characterized by applying intelligent technologies and concepts to create smart factories [1]. Under Industry 4.0, modern factories face an increasing level of automation to achieve more efficiency and quality in production processes. One of the key elements

KAHIOMBA Sonia KIANGALA and Zenghui WANG
Department of Electrical and Mining Engineering, University of South Africa, Johannesburg 1710, South Africa
Corresponding Author: Sonia KIANGALA
E-mail: sokiangala@gmail.com

of smart manufacturing is the use of robotics technologies in various applications varying from basic to advanced tasks such as parts processing in production lines, product transportation, material handling, logistic warehousing, rescue services, inspection, surveillance, data collection, and more collaborative activities with human operators [2–4]. The implementation of collaborative robots (cobots) aims to enhance the cooperation and the shared workforce between humans and robots in smart manufacturing environments [5,6]. Cobots are usually programmed with an extent of autonomy where the robots themselves can perform certain duties without relying on human intervention. When dealing with a moving robot, the concept of autonomy becomes more critical since the robot must safely navigate through the factory from the starting point to the destination by avoiding any obstacle along the way [7]. Obstacle avoidance for moving autonomous robots has been a challenging topic among researchers. One of these researchers' goals is to provide robots with an intelligent behavior system to make appropriate decisions in their working environment. Robot obstacle avoidance falls within the robot path planning procedure [8]. Robot obstacle avoidance has two main categories: obstacle avoidance in a totally or partially unknown environment and obstacle avoidance in a predefined environment [9]. To give robots the ability to make autonomous decisions like human beings, researchers used the great potentials of machine learning to build intelligent robot systems. Machine learning (ML) is a field in the domain of artificial intelligence (AI) [10] that focuses on providing machines (robots) with learning abilities like humans. There are several AI segments and branches developed for implementation in different domains. We present in Figure 1 an overview on a few of the AI branches such as ML. ML has various eminent applications in the areas of medical treatment [11], speech recognition [12], natural language processing (NLP) [13, 14] image processing and signal processing [15].

Reinforcement learning (RL) is an ML approach widely used in robotics to allow autonomous robots to learn obstacle-free trajectories to their destinations in a complex and uncertain environment without necessarily prior knowledge of the working environment. With RL techniques, autonomous robots can improve their performance through trials-and-errors interactions with the environment learning from their experiences to make the most appropriate decision in the future without external assistance. Hence, RL provides the robot a tool to learn and adapt in a dynamic environment [7]. Several studies have been conducted to implement RL in robotics: Ribeiro T et al [9] implemented an obstacle avoidance scheme for a mobile robot using an RL algorithm called Q-learning. This study applied two different Q-learning methods with an extensive tuning of the algorithm hyperparameters to obtain the best obstacle avoidance approach. This research was part of robotic competition. Zhao M et al [8]

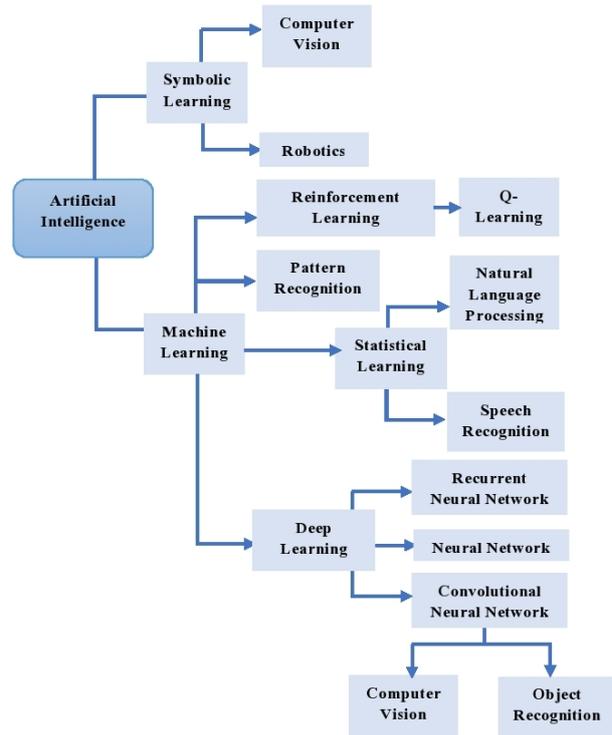


Fig. 1 Overview on a few AI branches

developed another study on robot path planning (including obstacle avoidance) using an advanced version of the Q-learning algorithm: an experience-memory Q-Learning (EMQL) through which the agent can receive two types of rewards (static and dynamic rewards). This method improves the robot's learning ability by utilizing two Q tables that allow the robot to remember a task in case of a change of destination. Another path planning research with Q-learning was carried out by Babu VM et al [16], where the robot can safely and effectively navigate in an unknown environment by capturing images (with an onboard robot camera) of the environment and computing the shortest path to the destination. This study applied machine learning to process the captured images. Research by Wiedemann T et al [17] applied RL has in robotics for information gathering in a hazardous gas leakage environment. They implemented an RL framework for the robot to learn how to get gas sources in a known environment. Bae H et al [18] extended RL application to a multi-robot system where several robots are interoperating and moving in a single environment performing different tasks without colliding with one another. They combined Deep Q learning and Convolution Neural Network (CNN) for image analysis to achieve their goal. In order to protect the robot from making fatal decisions during the learning process. Mannucci T et al. [19] defined a new RL approach with two safety algorithms to guide robots when learning their paths. This approach aimed to prevent robots from dan-

gerous states that could destroy them. Ou X et al [2] implements RL to solve a scheduling problem of robots moving from one cell to another loading and offloading production parts. RL allowed creating an effective moving scheduling policy for the robots in real-time to avoid production losses.

Another essential concept in the human-machine or human-robot interaction in smart factories is speech and language application via speech recognition [20]. Speech recognition [21] is another machine learning function through which machines can understand human speech and languages. Speech recognition eases the communication between humans and robots in a modern factory where the interaction between them is inevitable [23]. Thanks to speech recognition applications, voice instructions and commands can be given directly from human operators to robots. The speech recognition field for robotics technology has also been widely exploited with several works such as: Erol BA et al [22] who develop an intelligent system that gives robots the ability to detect human emotions by voice input or vision using speech recognition and deep neural networks. Rahat SA et al [24] created a system to control robot actions and give it instructions at a distance with voice commands. They built the system with a Raspberry Pi as the base controller to process the speech recognition algorithm and send commands to the robot. Another speech recognition for robot command work was done by Jevtić A et al [25]. This study aimed to develop an intelligent robot capable of dressing an individual (disabled people) with speech instructions. Speech recognition was just one of the multiple technologies included in this design, together with gesture and posture recognition. Jung SW et al [26] implemented a voice recognition system for an autonomous driving robot in a noisy environment. The system can effectively recognize different voice textures from adults to children despite the noise in the system. They used wireless microphones to capture the voice input and inserted multi-channel voice cancellers to reduce noise. A similar work designed to improve voice command quality in a speech recognition robotic system was conducted by Chan KY et al [27]. They improved the output signal of a speech recognizer installed in a control automation environment by applying an intelligent particle swarm optimization approach that tunes the speech recognizer and suppresses environmental noises.

In our research, we provide a procedure on how an autonomous robot could effectively respond to an emergency scenario in which the plant receives an evacuation signal. We use an RL algorithm called Q-learning to make an autonomous robot learn the most effective path to the closest safety exit. The safety exit can change depending on the robot's location when the emergency signal is received. We use speech recognition to create a voice command input to a plant programmable logic controller (PLC) as an additional stopping interlock that halts all plant operations and sends

an emergency signal in case of disaster. The voice command that controls the PLC directly is advantageous for the plant when the operator is far away from the system emergency stop (ESTOP) or stop buttons when the emergency signal needs to be activated. The voice command is an enhancement to an existing emergency stopping interlock.

1.1 Motivation of the study

Safety is a crucial factor in any working organization. In the industrial manufacturing environment, safety has become one of the highest priority areas to ensure an effective production system. Production factories contain several tools, pieces of machinery, and hazardous materials that could easily cause incidents without proper safety measures. An unsafe working environment results in frequent accidents, injuries, and machine failure [28]. One of the well-known emergency responses is to abruptly stop operations using system emergency stops (ESTOPs), often wired in control panels and other strategic locations. Unforeseen disastrous events are almost inevitable, requiring proper emergency responses to be observed by all factories personnel whenever they occur. In a smart factory context, where safety is a cyber-physical production system function [29], all production stakeholders should be involved in safety responses [30]. In an environment where humans and robots collaborate, the emergency procedure should apply to humans and robots. As much as humans go through safety inductions to be instructed on disasters' best safety measures, robots should comply with the same principle.

Our study proposes a safety enhancement measure for a small industrial manufacturing environment in which we equip an autonomous moving robot with an emergency procedure that is switched on in case of disaster. We also incorporate a voice command (via speech recognition) as an additional emergency stopping interlock to the central controller (a PLC), improving the emergency stop responses for situations where the operator is far from the ESTOP push button.

1.2 Contributions of the research

Our primary contribution is developing a safety response system in a small manufacturing environment where humans and machines (robots) collaborate. With the help of a reinforcement learning algorithm, Q learning, we make an autonomous moving robot to go through a "safety induction" to learn by itself the path to the closest safety exit based on its current location instead of relying on a single path to evacuate (that could be very distant from its location at the time of emergency) the place. In an emergency, the shorter

the traveling distance to the exit, the better as more incidents could happen along the way. Our experimental factory environment has a total of four safety exits. The destination of the autonomous robot is dynamic and depends on its current location. Our study also contributes to enhancing a PLC emergency stopping interlock by incorporating a voice command as an additional stopping input for the system in case of emergency. The voice command is captured from a microphone directly to the PLC via a python script without any other third-party controller like a Raspberry Pi.

1.3 Previous researches gaps summary

Most previous works on Q-learning (RL algorithm) in robotics have not tackled robots' safety regarding their working environment. Unlike in traditional factories, where robots are considered static production equipment, their role in a smart factory is more interactive, and they are part of the production workforce. It becomes essential to include them in safety responses in case of emergency. So far, the safety approaches covered in previous research have been on preventing robots from going into dangerous states while interacting with their environment. The fate of the robot in case of plant evacuation or emergency has not been widely explored. Various studies have been done on speech recognition applications in home environments, office environments, and even robotics for communication between humans and robots, but very few have been integrated directly into the process controllers' structures. A speech command incorporated into the central controller program has the advantage of controlling all other plants' operations in a single instruction. This feature could be convenient in emergencies. We address these gaps by creating a safety response mechanism (that we consider as a safety induction) for an autonomous moving robot to:

- Find an obstacle-free path to the closest safety exit based on its current location.
- Receive a voice command from the main plant PLC. The voice command is implemented as an additional stopping interlock for the PLC. It means that it can stop all ongoing plant operations in case of emergency.

The remaining part of this research is structured as follows: Section 2 summarizes the background and the theory on essential concepts used in this study, such as Reinforcement learning, Q-learning, Speech recognition, and programmable logic controllers in an I40 environment. Section 3 presents the methodology used to design our safety response mechanism. Section 4 describes the overall system architecture, the experimental factory, and the workflow diagram of our safety procedure. Section 5 highlights the experimental background, the testing settings and procedures,

the results, and our system's discussions. Section 6 is a conclusion to our study and a guideline for future works on this topic.

2 Background and theory: summary

2.1 Reinforcement Learning (RL)

The reinforcement learning (RL) [31–33] algorithm is a machine learning method created in the early 1990s. This algorithm is built based on the enhanced learning principle where an “agent,” the object undergoing the learning process, continuously tries all possibilities of actions and decisions to determine the most optimal one (the best strategy to adopt) [34]. The RL algorithm utilizes five main elements: the agent, the environment, the reward, the state, and the action. In its learning process, the agent performs several interactions with its environment by making some actions that will cause a change of state in the environment and result in a positive or a negative reward (penalty) [35]. The agent learns the impact of its actions based on the type of reward obtained. RL, therefore, falls into the unsupervised category [36], where the agent does not rely on external knowledge accumulated by an intelligent supervisor to make the correct decision [2, 37]. In short, it can be stated the agent learns the strategy to adopt through a series of trials-and-errors [9]. Over the years, RL has been the subject of many kinds of research in various applications such as chemical reaction [38], Resource Management [39], Traffic-light Management [40], Autonomous Driving [41], Dam Management [42], Surgery [43] and Robotics [44].

Our study focuses on the robotic area where we use RL to equip an autonomous robot, the agent, with an appropriate routine to learn the shortest and safest way to a secure location in case of a disaster that requires evacuation in its working environment. The RL algorithm has two main groups: model-based algorithms and value-based algorithms. In model-based algorithms, also known as an on-policy-based algorithm, the agent predicts its next state and the reward it will get before making an action based on a transition and a reward function. Value-based algorithms, also called model-free algorithms, are the opposite of model-based ones; they do not use modeled policies to determine actions to take. They rely on values obtained during the trial-and-error learning process to choose the optimal strategy [45]. One of the most popular model-free algorithms in reinforcement learning is the Q-learning algorithm. Model-free algorithms have been successfully implemented in robotics problems because of their ability to reuse information effectively [46]. Figure 2 is a summary of the working principle of the RL algorithm.

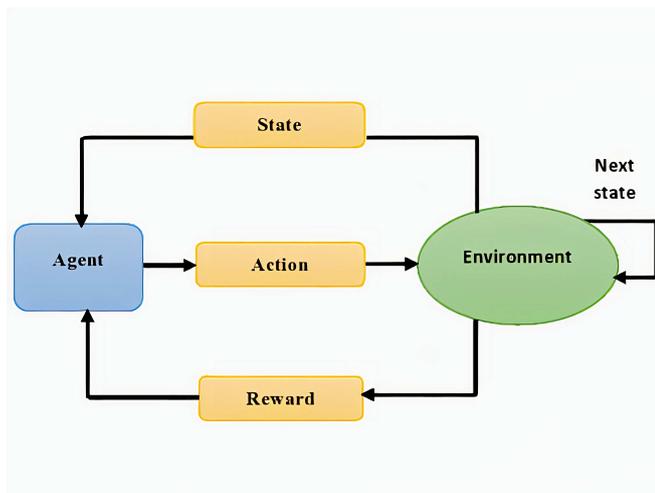


Fig. 2 RL operation principle [47]

2.1.1 Q-learning algorithm

The Q-learning (QL) algorithm is a model-free RL algorithm introduced by Watkins CJ and Dayan P [48]. It has been developed based on the psychological behaviourism of offering reward or encouragement for good actions and penalties or punishment for wrong actions. One of the most significant advantages of the QL algorithm is that it does not require prior knowledge of the environment to start its operations. It learns on the fly via trial-and-error. The agent performs all possible actions in every state and records the values generated in a Q table. The Q table is a matrix in which the rows represent the environment's states and the columns the corresponding actions of the agent at a state. The values in the Q table are computed based on a Bellman Equation [8]. Once the Q table is populated after several interactions, the agent uses the information in the Q table to choose the action giving the highest cumulative reward [35,49]. QL can be applied as a dynamic and incremental programming method to find the optimal strategy for a problem in a step-by-step learning mode. QL is the best solution for Markovian decision problems that do not clearly understand the transition probabilities [50]. In this research, we implement the QL algorithm to build a safety mechanism for an autonomous robot.

Limitation: It is worth mentioning that the QL algorithm is only suitable for discrete states and actions. Its learning processing time depends on the size of the Q table. The larger the Q table, the longer the learning processing time for the agent [36].

2.2 Speech recognition

Speech recognition is an AI process that converts a voice signal (usually a human voice) into instructions that com-

puters can understand and further manipulate for diverse applications. The voice signal's conversion involves a transformation of an analog signal (the original voice input) into a digital sequence of sentences or words from the voice instruction [51]. Speech recognition is a multidisciplinary field based on several research areas such as pattern recognition, signal processing, cognitive science, acoustics, digital signal processing, and linguistics. As previously highlighted, AI aims to give computers intelligence to behave like human beings and interact with them. If computer vision [52] is an AI method that allows computers to recognize patterns with the sense of vision or by seeing like humans, speech recognition permits machines or computers to interact with each other and with humans through language or speech. A speech recognition system contains several segments such as an acoustic model, a feature extraction algorithm, a search algorithm, and a language model [53].

In recent years, various applications using speech recognition technologies have been developed mainly to substitute traditional input methods like clicking or typing with voice commands. The creation of virtual assistants' voice-enabled applications such as Google Home assistant, Apple home pod, and Amazon Alexa has become quite popular in home and office environments. Nowadays, almost all smartphones are equipped with speech recognition applications to control device features like dialing contacts numbers, voice searching, and much more. Even in the automotive sector, modern cars can now be voice-controlled for various tasks that were previously manually controlled: switching on the headlights, putting in a destination address, calling in contact, to list some of them [51]. In all these applications, one of the ultimate goals of speech recognition remains to provide a way to ensure effective human-to-machine communication and to develop more robust systems that will ease this communication [54]. However, very few speech recognition applications have penetrated the industrial manufacturing sector. Our study aims to enhance an industrial manufacturing plant's security by incorporating a voice command as an additional emergency or safety input signal to a traditional PLC. Therefore, we apply the speech recognition process to an industrial manufacturing environment.

Limitation: One of the most significant limitations of speech recognition is that it requires a noise-free input command to carry out its process efficiently. Obtaining a clear input can be challenging because of different language accents and intonations that sometimes the system has difficulty recognizing. Several types of research exist to reduce the noise level in speech recognition input systems. It is not advised to use a speech recognition application as a single input for safety control [54]. In our study, we use the voice command input as an enhancement to an existing safety mechanism.

2.3 Programmable Logic Controllers (PLC) functions in the era of Industry 4.0

Programmable Logic Controllers (PLCs) have been widely used in the industrial sector as the base element for controlling and monitoring field components and production processes. Not many researches talk about this powerful controller [55]. PLCs have been successfully established as the platform of choice for industrial process outline and factory automation conducted by the so-called IEC 61131 standard [57]. They provide reliability, safety, efficiency, and security to industrial applications. PLCs are famous for their ability to generate an ecosystem for simple programming software logic (the ladder logic), ruggedized hardware, robust systems, real-time control of network components, and field components like actuators and sensors. In summary, a PLC system contains three main sections: a programming section, a networking section, and a physical section. The physical section comprises the inputs segment (sensors) and the outputs segment (actuators) [55].

Although the industrial and manufacturing sector is undergoing a smart manufacturing revolution, encouraging the utilization of more general-purpose devices, tools, networks, and software, the use of PLCs is not yet ready to vanish. To this date, not so many operating systems or embedded controllers can offer the compact advantages that PLCs have in the industrial sector. However, the growing tendency of intelligent manufacturing technologies in Industry 4.0 increases PLCs' challenges to produce more intelligent and innovative solutions for factories. In smart manufacturing, the PLC role evolves from traditional controllers' tasks to decentralized and integrative functions [60], [61]. One of our study goals is to integrate a voice command input via speech recognition to a traditional PLC that initially does not support any voice command. Hence, we transform a traditional isolated PLC system into a smart controller in an Industry 4.0 environment.

3 Safety response mechanism methodology

Our experimental safety mechanism for an autonomous robot consists of:

- An automatic path finding section to safety exits in case of emergency: We build this section based on the Q-learning algorithm.
- A voice command input to a Siemens S7-1200 PLC to activate the emergency signal: Speech recognition is the key feature of this segment.

3.1 Q-learning algorithm to find obstacle free paths

Q-learning is a reinforcement learning algorithm. One of its working principles is that the agent (the autonomous robot in our research) learns the best policy to adopt for a given scenario based on its interactions with the environment and the rewards gained [35]. According to its current state, a policy in Q-learning is how the agent chooses to behave, react, or make decisions at a given time. Therefore, the policy is a result of the agent's action and state [57]. A policy can be represented by (1):

$$\beta = a|s. \quad (1)$$

where a is the action taken by the agent at a given time and s is the state of the agent when taking the action. β formulates the probability an agent takes an action a based on its state s .

During the learning process, the Q-learning algorithm computes values generated by pairing each possible action the agent may take based on different states. These calculated values are called Q-values. The agent picks the action that produces the highest Q value. The Q values are stored in a Q Table until the best possible policy is determined by the algorithm [58]. As mentioned previously, the QL algorithm originated from the Bellman equation. The Bellman equation can be applied in a deterministic environment where it is assumed that the agent has 100% chances to go through the desired direction and end up in the desired state when instructed to do so. (2) is the bellman equation for deterministic environment. The Bellman equation can also be implemented in a stochastic or non-deterministic environment in which there is a probability that the agent may end up in a different state than expected. (3) represents the Bellman equation for stochastic problems, also referred to as Markovian decision trees [8].

$$G(s) = \max_a (r(s, a) + \gamma G(s_n)) \quad (2)$$

where $G(s)$ is the value obtained at a given state s , $r(s, a)$ is the reward obtained by performing an action a at a state s , γ is an hyperparameter called the discount factor, and $G(s_n)$ is the value obtained at the next state s_n .

$$G(s) = \max_a (r(s, a) + \gamma \sum_{s_n} P(s, a, s_n) G(s_n)) \quad (3)$$

where $P(s, a, s_n)$ is the probability of getting into the next state s_n from the initial state s by performing an action a . The final Q-learning equation is presented by (4).

$$Q(s, a) = (1 - \alpha) Q(s, a) + \alpha [r(s, a) + \gamma (\max_{a_n} Q(s_n, a_n))] \quad (4)$$

where $Q(s,a)$ is the Q value computed by an agent performing an action a at its current state s . α is a Q-learning algorithm hyperparameter that represents the learning rate. Its value is defined in the interval $\alpha \in [0, 1]$ and it determines the impact of new information learnt compared to old stored ones. $r(s,a)$ is the reward obtained by performing an action a at a state s . γ is the discount factor. Its role in the Q-learning algorithm is to give importance in future rewards. Like its counterpart α , the value of $\gamma \in [0, 1]$ [9]. $Q(s_n, a_n)$ is the Q value of the next action a_n at the next state s_n . $\max_{a_n} Q(s_n, a_n)$ calculates the maximum Q value of the next action based on the next state. We can summarize the operations of the Q-learning algorithm (4) in Algorithm 1.

Algorithm 1 Q-learning operation summary

Input: Agent states s

Output: Q values

Initialize $Q(s,a)=0$ and $Q(s_n, a_n) = 0$

Set α and γ

1. Record a state s
 2. Choose an action a based on the state s
 3. Examine the reward received $r(s,a)$
 4. Record the next state s_n
 5. Find the maximum Q value based on (s_n, a_n)
 6. Find the value of $Q(s,a)$
 7. Go back to 1
-

The Q-learning algorithm allows our autonomous robot to find an obstacle-free path to the closest safety exit in an emergency. The choice of the safety exit is dynamic and depends on its current location. We create an algorithm (Algorithm 2) to find the route from the current robot location to its destination. The computed route should contain the indexes of all the locations through which the robot passes going to the destination. This algorithm utilizes Q values stored in the Q table.

Algorithm 2 Finding the obstacle free path from the current location to the safety exit

Input: Agent current location s

Output: Path to closest safety exit

Initialize the system, set the Q table

Read current location index

Set the current location as start

Determine the closest safety exit location index

Set the closest safety exit as Destination

while (start! = Destination)

1. next location = Location [max Q value (start)]

2. route = start index + next location index

3. start = next location index

end while

return Route

3.2 Speech recognition process to activate a voice command input to the PLC

Under the smart manufacturing era, the human-machine interface (HMI) concept has considerably expanded from simple push buttons, light indicators, and graphical user interfaces (GUI) to more advanced interfaces [62] driven by voice and gestures. A new concept called natural human-machine interface (NHMI) is developing enabled by gesture recognition, enhanced reality, and speech recognition [63]. Our research focuses on the implementation of speech recognition to create a new interface for a traditional PLC. The speech recognition principle's theoretical summary is the ability to recognize and identify patterns of information contained in a speech wave [64]. On the technical side, speech recognition aims to predict the best word sequence B based on a speech wave signal S and can be represented by (5) for its statistical expression [65]:

$$B^* = \arg \max_B P_{\Psi, \lambda}(B|S) \quad (5)$$

where Ψ is the acoustic model and λ is the language model. (5) can be transformed and presented as (6):

$$B^* = \arg \max_B p_{\Psi}(S|B)P_{\lambda}(B). \quad (6)$$

where $p_{\Psi}(S|B)$ represents the probabilistic quantity generated by the acoustic model and $P_{\lambda}(B)$ is the probabilistic quantity that originated from the language model.

(6) can be further expanded in (7) by assuming that a time sequence t is inserted and some observations of the word sequence b_t are generated with hidden states δ_t by the hidden Markov models (HMMs).

$$B^* = \arg \max_B P_{\lambda}(B) \sum_{\delta} \prod_{t=1}^T p_{\Psi}(b_t|\delta_t)P_{\Psi}(\delta_t|\delta_{t-1}) \quad (7)$$

The speech recognition system's first step is to process a feature extraction module, also known as the speech recognition front-end, that produces an acoustic feature. The second step is to transfer an acoustic feature into a language and an acoustic model to obtain the probability of the received initial word sequence. This step constitutes the back-end section of a speech recognition model. The back-end's output is a sequence of words generated from an acoustic and a language model [65].

We use the existing google application programming interface (API) to provide the language and acoustic models for the design of our speech recognition system. The operator's expected voice command goes through the speech recognition system and is converted into an action enabler

for emergency or evacuation cases. We present an outline of our voice command input system in Algorithm 3.

Algorithm 3 Voice command to PLC input via speech recognition

Input: Voice command

Output: Control PLC output

Import all important libraries (speech recognition and PLC connection)

Define write and/or read functions for PLC

Establish connection with the PLC

Endless loop

1. Listen to Microphone (sound source)

2. Recognize the language using Google API

if voice command == emergency command:

Write Instructions to PLC variable

A summary of the procedure of our overall safety mechanism combining the Q-leaning path finding method and the voice command input is presented in Algorithm 4. We assume that:

- θ is the emergency/evacuation signal.
- X is the autonomous robot.
- η is a safety exit.

Algorithm 4 Safety mechanism procedure summary

Input: Voice command and X current location

Output: System stop and robot path to closest η

1. Activate θ

2. Send θ to X from PLC.

3. X initiates the safety routine.

4. X determines the closest η .

5. X finds the obstacle free path to η

6. X reaches η

7. end

4 Experimental environment architecture and workflow diagram

We present our simple experimental manufacturing location overview in Figure 3 with its legend describing all essential components. The autonomous robot should travel from one location letter to another, avoiding obstacles until it reaches one of the safety exits. Our safety enhancement method using Q-learning enables the autonomous robot to choose the correct path to the closest safety exit (E, G, S, or W depending on its current location) when it receives an emergency evacuation signal. We assume that the emergency evacuation signal is generated by a push-button input to a Siemens S7-1200 PLC or a voice command from a microphone directly to the PLC. Figure 4 displays the conversion architecture from the voice command to the PLC for activation

of the emergency signal. Figure 5 summarizes the safety response workflow diagram for the autonomous robot once the evacuation signal is received.

5 Experimental results

5.1 Experimental background and assumptions

- Every industrial plant has or should have a safety protocol to activate in case of emergency. It could lead to evacuating the plant and pressing one of the emergencies stop interlocks (ESTOP or Stop buttons depending on the disaster’s severity) that usually halts the whole system. The stop interlocks are wired inputs of the plant controller (a PLC for this experiment) that are programmed to control several outputs. These stop buttons (including ESTOPS) are connected to the control panels and sometimes on different strategic locations.

How to palliate to emergencies where none of the operators is nowhere near to the stop buttons?

We answer this problem by incorporating a voice command as an additional stop interlock programmed in the PLC. The operator can use the command "emergency" to alt to the system via a microphone. We apply a speech recognition algorithm to convert the operator’s voice command to a PLC’s instruction. A Siemens S7-1200 PLC is used for testing the program via the TIA portal software. We develop the speech recognition code in a python integrated development environment (IDE).

- Significant modern buildings are constructed with different safety exits located at different areas to ensure smooth and efficient evacuations in case of emergencies. It could be more dangerous for people to always go out through a single safety exit that could be far away from their current location, risking getting more injured on the way.

What happens to autonomous robots in a smart manufacturing environment when a signal to evacuate the plant has been given, and they were busy performing one of their tasks at the plant’s location?

We address this problem by implementing the same principle of decentralized safety exits for an autonomous robot. We simulate different safety exits for the autonomous robot based on its current location, and we apply the Q-learning algorithm for the robot to learn the obstacle-free path to the closest safety exit when an evacuation signal is received. We, therefore, reduce the risk of undesired incidents on the robot in an emergency evacuation. We

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

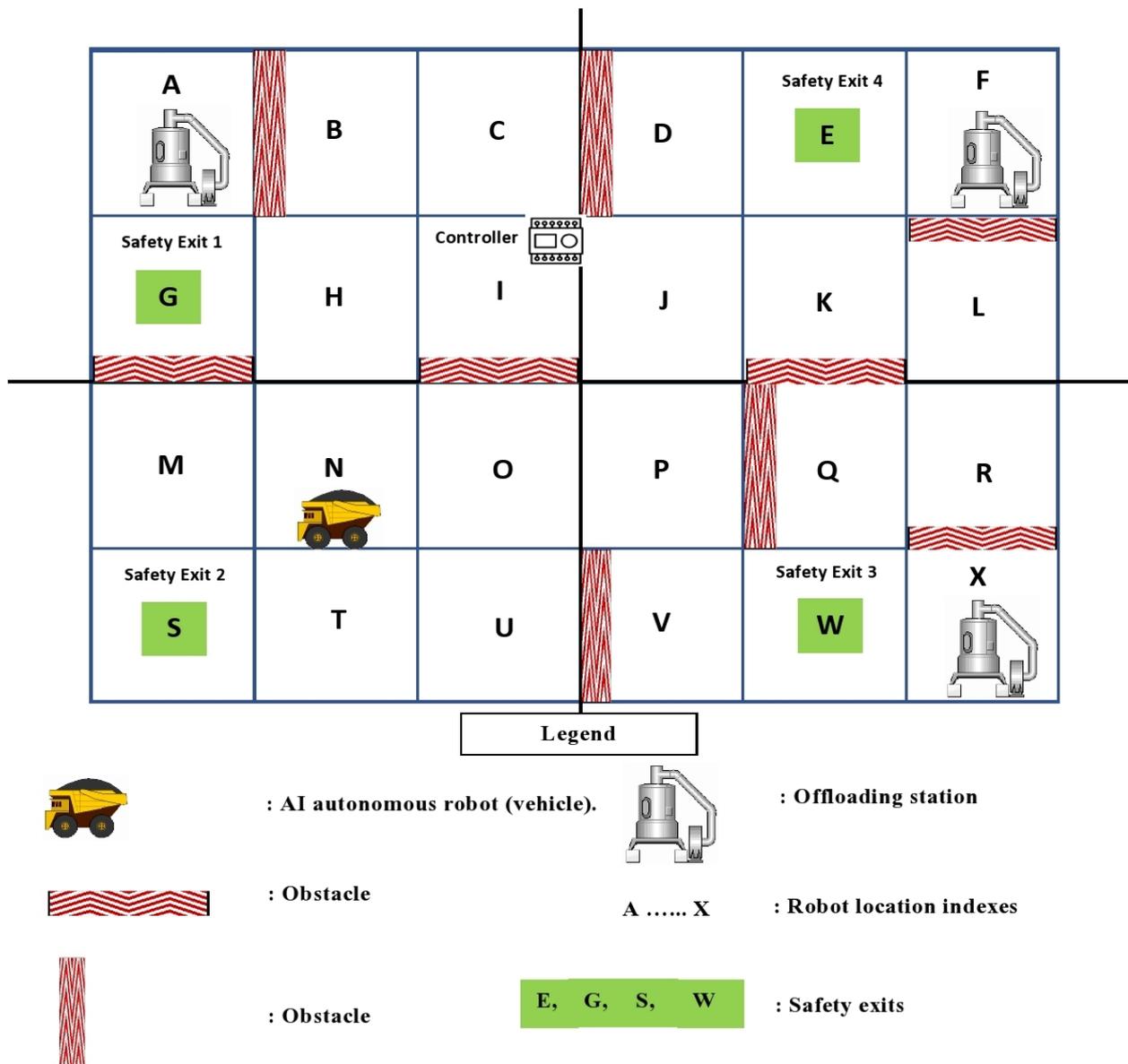


Fig. 3 Experimental manufacturing environment for autonomous moving robot

use python to program our Q-learning method for safety exits.

We make the following assumptions:

- In the design of our safety procedure response system, we assume that the autonomous robot has been equipped with onboard means like sensors and cameras to interact with other field components (such as the main plant PLC) and knowledge of its current location.
- We also presume that the microphone used to capture the voice command is powerful enough (wide range) to detect the operator’s emergency command from a long distance and has a noise suppression scheme.

- Another assumption we make in this research is that the autonomous robot has a designated route that does not interfere with human operators’ paths.

5.2 Finding Robot paths to the closest safety exits using Q-learning

From the simulated plant environment layout in Figure 3, we use the Q-learning algorithm to determine the path to the closest safety exit for the autonomous robot depending on its current location in case of an emergency signal that requires a complete exit. Finding the correct paths to the emergency exits include avoiding fixed obstacles in the plant. At the

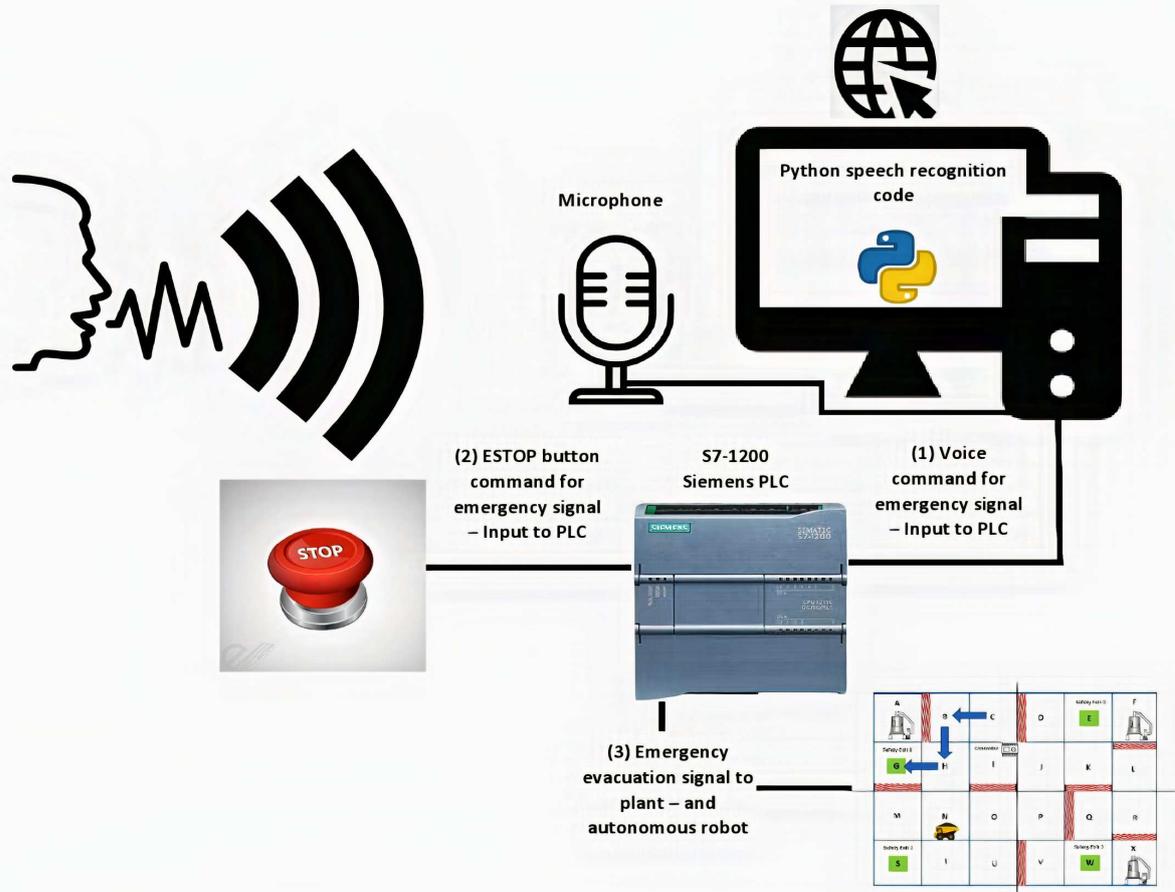


Fig. 4 From voice command to PLC emergency evacuation signal

beginning of our design, we assume that the autonomous robot does not know anything about the environment. We divide the plant floor into four segments having each a safety exit of its own. The four segments and their safety exits are as follows:

- For segment 1 - locations A,B,C,G,H and I: safety exit G
- For segment 2 - locations M,N,O,S,T and U: safety exit S
- For segment 3 - locations P,Q,R,V,W and X: safety exit W
- For segment 4 - locations D,E,F,J,K and L: safety exit E

One this design's vital steps is to translate every detail of the experimental plant environment, R , into a matrix of size $m \times m$, with m the plant's number of locations. The Q-learning algorithm is applied in this matrix format. As per Figure 3, the features of our experimental environment are as follows:

$$m = \{A, B, C, \dots, X\} = 24$$

$$R_{size} = 24 \times 24$$

In the matrix R , a '1' signifies two neighboring locations, in the intersection of a row and a column, with no obstacles. A '0' means that two locations (row and column) do not share a border or have an obstacle between them. A '10' represents the location designated as the safety exit for a specific segment. The value to insert in the safety location is arbitrary and must be greater than 0 and 1. Figure 6 is the corresponding environment matrix for segment 1 before applying the Q-learning algorithm. After applying the Q-learning algorithm to the matrix R , we obtain different reward values for each neighboring location. For segment 1, the safety exit location G has a higher reward value of 33. When moving from one location to another, the autonomous will try to use the path that gives the highest reward until it reaches the safety exit. From the Q-learning algorithm equation (4), we chose the Hyperparameters' following values: $\alpha = 1$ and $\gamma = 0.7$. After computing the Q-learning reinforcement learning, Figure 6 is converted to Figure 7.

Values in Figure 7 are the rewards given to the autonomous robot for each location visited. The number 33 is the highest reward at location G, the safety exit. We apply the same

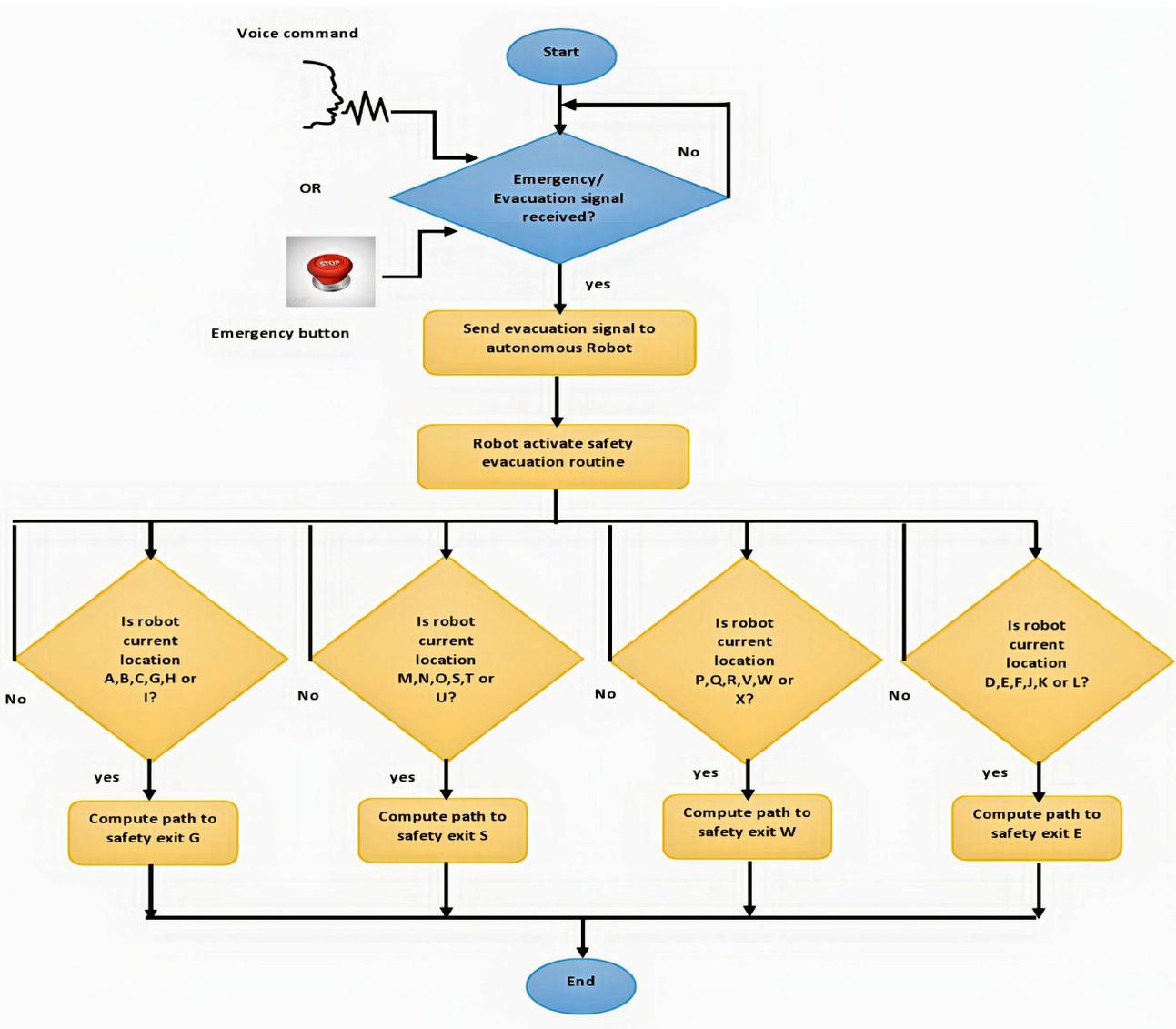


Fig. 5 Safety response mechanism flow diagram

principle for the three other segments where the main difference is the safety exit location. The safety exit moves from G to S for the second segment, from G to W for the third segment, and from G to E for the fourth segment. From the experimental environment in Figure 3, the rewards values from one location to another for segment 1, based on values in Figure 8, are:

- From A to B or from B to A - Reward value 0. The reward value is 0 because of the obstacle between the two locations
- From A to G - Reward value 24.
- From G to A - Reward value 17. The difference in reward is because going from G to A, the robot moves away from the safety location (G) but from A to G, it moves closer hence a higher reward.
- From B to C - Reward value 0.
- From C to B - The difference in reward is because going from B to C, the robot moves away from the safety location (G) but from C to B, it moves closer hence a higher reward.
- From C to D - Reward value 0. The reward value is 0 because of the obstacle between the two locations.
- From C to I - Reward value 13.
- From I to C - Reward value 10. the difference in reward is because going from I to C, the robot moves away from the safety location (G) but from C to I, it moves closer hence a higher reward.
- From I to J - Reward value 10.
- From I to O - Reward value 0. The reward value is 0 because of the obstacle between the two locations.
- From I to H - Reward value 17.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
A	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	1	0	0	0	0	0	100	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
I	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
K	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
N	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0
P	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
R	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fig. 6 Experimental environment matrix for segment 1 before Q-learning

- From H to I - Reward value 13. The difference in reward is because going from H to I, the robot moves away from the safety location (G) but from I to H, it moves closer hence a higher reward.
- From H to N - Reward value 13.
- From H to G - Reward value 24.
- From G to H - Reward value 13. The difference in reward is because going from G to H, the robot moves away from the safety location (G) but from H to G, it moves closer hence a higher reward.
- From G to M - Reward value 0. The reward value is 0 because of the obstacle between the two locations.

When learning the best path to the closest safety exit, the robot cumulates the best possible rewards to get in each location until it reaches the destination. We implement algorithm 2 in our python code to display paths selected by the autonomous robot based on any location it finds itself when the central controller receives the emergency exit sig-

nal. The simulation of the current location is done by receiving an input (one of the location letters) and observing the path displayed by our code after computation. This procedure is how the autonomous robot determines the path to the closest safety exit using Q-learning. The Q Table values in Figure 7 are critical elements of the algorithm operation. For an input location (variable loc) equals to location 'C,' the path in Fig. 9 is set forth. Fig. 10 is the graphical equivalent of the path. The same simulation can be tested for any of the 24 locations of the environment. A summary of algorithm 2 results for location C from the Q Table in Figure 7 can be described as follows: from row C, the highest reward (maximum reward) in line is 13 for both columns B and I. In other words, the autonomous robot can choose to go through location B or I on its way to the safety exit G. Since B is the first column it encounters in the Q table, it chooses B as the following location after C the starting point: C-B. After designating B as the following location, B becomes the new row

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
A	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	10	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	13	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	6	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	8	0	5	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	17	0	0	0	0	0	33	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	13	0	0	0	0	24	0	13	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0
I	0	0	10	0	0	0	0	17	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	8	0	0	0	0	13	0	8	0	0	0	0	8	0	0	0	0	0	0	0	0
K	0	0	0	0	6	0	0	0	0	10	0	6	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	5	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	8	0	0	0	0	0	0
N	0	0	0	0	0	0	0	17	0	0	0	0	10	0	10	0	0	0	10	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	13	0	8	0	0	0	0	8	0	0	0	0
P	0	0	0	0	0	0	0	0	0	10	0	0	0	0	10	0	0	0	0	0	0	6	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	5	0
R	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	5	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	10	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	8	0	8	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	10	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	5	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	6	0	5	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0

Fig. 7 Autonomous robot Q Table for safety response in segment 1

in which the maximum reward (highest reward) value is selected. As per Fig. 8, the highest reward in row B is 17 from column H. H is, therefore, the following location after B: C-B-H. The exact process is applied to H until the autonomous robot reaches the destination, the safety exit, G: C-B-H-G, as displayed in Figure 8 and Figure 9.

5.3 Using a voice command to control a Siemens S7-1200 PLC emergency stop interlock

An emergency stop interlock is usually programmed in a controller to abort a system’s operation in case of emergency or unforeseen events requiring an immediate stop. In a traditional installation, a stop button or an emergency stop button is wired in the control panel where the operator can stop operations whenever required. For more flexibility in the safety measure, if the operator is far away from the control panel in an emergency, we add a voice input command directly

to the PLC controller as an additional interlock to stop the system. Traditional PLCs that are still operational in several plants have not been designed to receive voice command inputs. Usually, third-party controllers are needed to receive the voice input and interact with the PLC. In this research, we do not use any external controller to receive the voice command; we compute the speech command from the microphone straight to the PLC via a python speech recognition code. We use an S7-1200 Siemens PLC for the simulation in which a simple ladder logic program is written, as displayed in Figure 10, for the stopping and starting interlock. We explain the function and the role of each variable in Table 1.

The start and stop interlock in Figure 10 is written in a “latch” programming format where a push button signal (ON-OFF) starts the system (via M0.0) and the outputs (Q0.0 and Q0.2) remain activated until one of the stop commands is activated (M0.2 or M0.1). We present a summary of im-

```

1 #Print the route based on the selected location
2
3
4 if loc == 'A' or loc == 'B' or loc == 'C' or loc == 'G' or loc == 'H' or loc == 'I':
5     print(route(loc, 'G'))
6
7 elif loc == 'M' or loc == 'N' or loc == 'O' or loc == 'S' or loc == 'T' or loc == 'U':
8     print(route(loc, 'S'))
9
10 elif loc == 'P' or loc == 'Q' or loc == 'R' or loc == 'V' or loc == 'W' or loc == 'X':
11     print(route(loc, 'W'))
12
13 else:
14     print(route(loc, 'E'))
15
16 ['C', 'B', 'H', 'G']

```

Fig. 8 Selected route displayed from location C

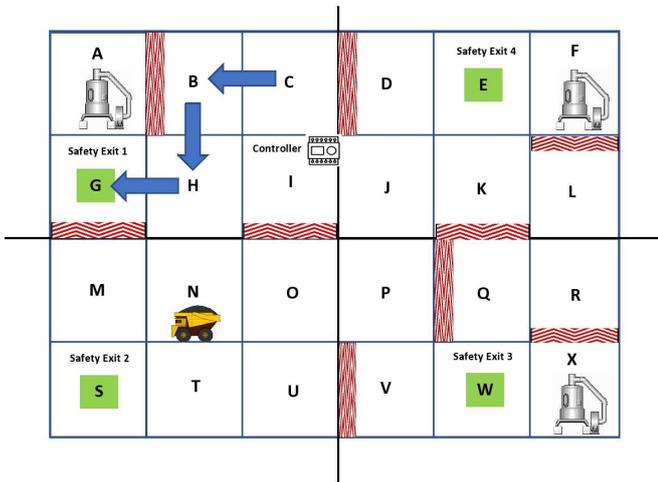


Fig. 9 Path from C to safety exit G graphic

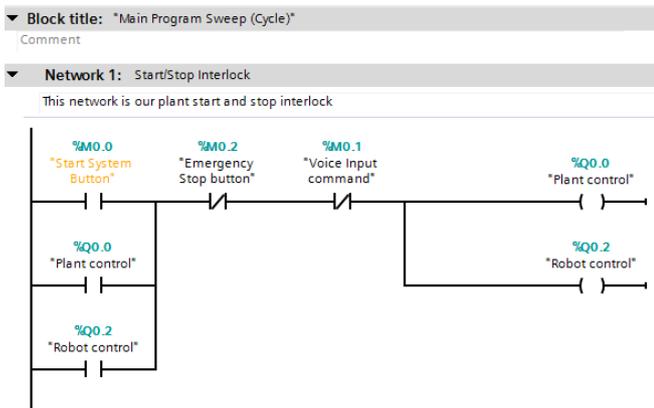


Fig. 10 Stopping and starting interlock PLC programming code

Table 1 PLC Start and Stop interlock variables

Variable address	Variable function
M0.0	This is the start signal for the system. It is usually a ON-OFF action directly wired to a PLC input IX.X. For this simulation, we use a memory bit M. When the memory bit is ON the system starts if M0.1 and M0.2 are OFF.
M0.1	This is the memory bit controlled externally by our python speech recognition code that converts the voice command to a 'ON' bit value. When M0.1 is ON the system is in stop mode.
M0.2	This is the memory bit controlled by an emergency stop signal. In real plant applications it is usually an input signal IX.X. For simulation, we use a memory bit variable M.
Q0.0	This is a physical PLC output that controls the plant.
Q0.2	This is a physical PLC output that controls the plants robots.

portant experimental settings for the S7-1200 PLC in Table 2.

We translate the voice command into an input to the PLC via a speech recognition python code (using a google Application Programming Interface (API)) and a python to siemens PLC library. The Speech recognition library converts the voice signal from the microphone into a text that we manipulate in our code to set the PLC variable M0.1. The required settings to achieve successful speech recognition command to the S7-1200 PLC in the python code are shown in Table 3.

Figure 11 is a representation of a healthy running system monitored from a live S7-1200 PLC when no stop signal is received. Table 4 is an explanation summary of each

Table 2 S7-1200 PLC settings for speech recognition process

Settings	Values	Comment
PLC IP address and MAC address	172.16.2.254	This is the experimental IP address; it can be adapted to any other IP address.
Python PC IP address and MAC address	172.16.2.202(with internet access)	Internet connection needed for the Google speech API.
PLC security/protect level	Permit access with PU/GET communication – No protection.	This is to allow the external python code to modify the variable value.
Speech recognition variable	M0.1	This is a physical PLC output that controls the plant.

Table 3 Python IDE settings for speech recognition to a Siemens S7-1200 PLC

Settings	Values	Comment
Python version	Idle (Python 3.7 32-bit)	
Libraries	Python-snap7 speechrecognition wheel pyttsx3 pipwin pyaudio playsound gTTs	Install libraries in Python IDE.
plc = c.Client()	plc.connect (172.16.2.254,0,1)	This is the IP address of the PLC.
rpausethreshold	0.5	The Speech recognition system listens after 0.5 seconds.
Voice source	Microphone as Source	Select the Microphone as the source of the voice.
Write Command	WriteMemory (plc,0,1, S7WLBit,1)	Writing command to M0.1 variable.

variable’s value in a healthy running system (as per Figure 11).In case of a fault that requires an emergency shutdown, the voice command can be given via the microphone and directly affect the PLC code. Figure 12 and Figure 13 display respectively a portion of the speech recognition code and output from the python console to stop the system after a successful voice is received. The result in the S7-1200 PLC after a voice command is received to stop the system is displayed in Fig. 15. The system is now in stop mode. A summary of the variables updated values after the stop voice command is presented in Table 5. If we assume that the voice input command is also programmed as a plant evacuation signal, the PLC that controls operations in the whole plant sends a signal to the autonomous robot (via Q0.2 turned OFF) to start the emergency procedure finding their closest path to the safety exit.

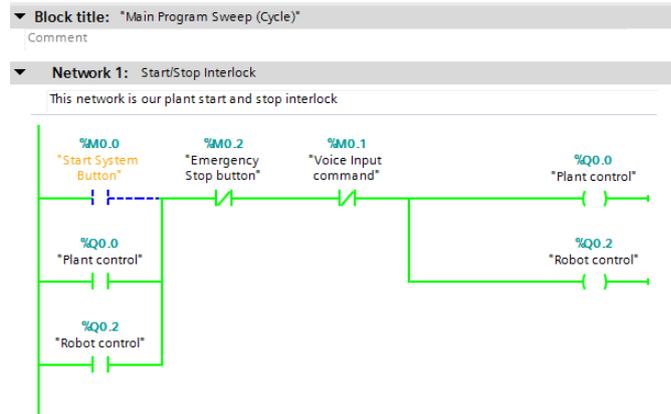


Fig. 11 Healthy running plant PLC program

Table 4 PLC healthy running system start-stop interlock variables values meaning

Variables	Values	Comment
M0.0	0	Because of the latch programming format. M0.0 does not have to remain ON ('1') for the system to run continuously.
M0.1	0	No stop signal has been sent by the voice command.
M0.2	0	No stop signal has been sent by the emergency stop button.
Q0.0	1	The plant control signal has been activated.
Q0.2	1	The robot control signal has been activated.

```
def takeCommand(self): #Method to give Input as Voice
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print('Listening')
        r.pause_threshold = 0.7
        audio = r.listen(source)

    try:
        print("Recognizing")
        Query = r.recognize_google(audio, language='en-
        print("the query is printed=", Query, "")

    except Exception as e: #handle Exception
        print("Please Speak Again")
        return "None"
    return Query

def sysstop(self): #Method to stop the system
    take = self.takeCommand()
    choice = take

    #Stopping the system
    if choice == 'emergency':
        print("Shutting down Your System")
        WriteMemory(plc,0,1,S7WLBit,1)
        self.Speak("System stopped")

if __name__ == '__main__': #Run the Code
    Maam = Shutdown()
    Maam.sysstop()
```

Fig. 12 Speech recognition segment in python code

```

1 Window Help
2 Python 3.7.9 Shell
3 File Edit Shell Debug Options Window Help
4 Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:01:5
5 (Intel)] on win32
6 Type "help", "copyright", "credits" or "license()" for mor
7 >>>
8 = RESTART: C:\Users\GigaSol\AppData\Local\Programs\Python\
9 ()
10 _Voice with googleAPI working_04032021_2015.py
11 Listening
12 Recognizing
13 the query is printed=' emergency '
14 Shutting down Your System
15 >>> |
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

```

Fig. 13 Speech recognition code output after successful voice command

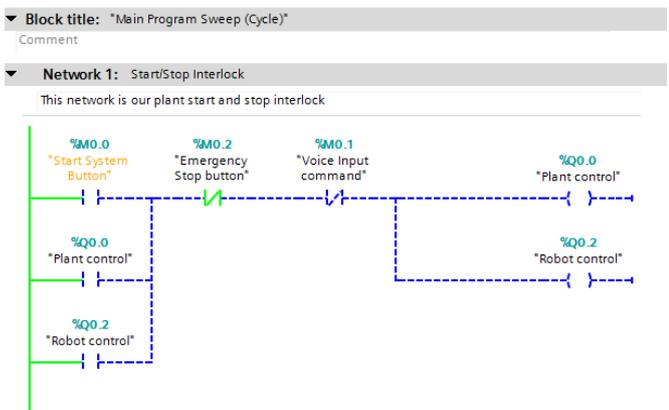


Fig. 14 PLC system stopped after emergency voice command received

Table 5 PLC healthy running system start-stop interlock variables values meaning

Variables	Values	Comment
M0.0	0	Because of the latch programming format. M0.0 does not have to remain ON ('1') for the system to run continuously.
M0.1	1	An external voice command is received to stop the system.
M0.2	0	No stop signal has been sent by the emergency stop button.
Q0.0	0	The plant control signal is stopped.
Q0.2	0	The robot control signal is stopped.

5.4 Results summary and discussion

Our study was about providing additional practical measures to enhance the safety procedures of a smart manufacturing plant. Safety in a manufacturing plant remains a priority for every organization to ensure a smooth run of their production activities [66]. In an era of smart manufacturing and digitalization, the word safety is no longer limited to employees or operators working in the plant but to machine interacting and collaborating with them daily, such as collaborative robots (cobots) and autonomous robots. Firstly, we tackled the safety of autonomous robots moving in a manufacturing plant to perform specific tasks, such as offloading parts to

different working stations, by implementing the robot’s Q-learning algorithm to learn the best trajectory to the closest safety exit in case of disaster. Secondly, we incorporated a voice command, with the speech recognition algorithm, as an additional interlock to stop the operation of a plant directly via its control PLC in case of an emergency. The Q-learning algorithm allows the autonomous robot to learn the most efficient route to the desired target by accumulating the highest rewards along the way. The target has the highest reward. Our experimental environment has twenty-four locations (from A to X) through which the autonomous robot travels. As a strategy to find the closest safety exit for the robot, we divided the environment into four sections where each has its safety exit. In that way, the robot does not have to travel very long distances to get to a designated safety exit when an emergency signal is received from the central controller. During the learning phase, the four targets have the highest reward for the autonomous robot to find its way to any of them depending on its current location. We assigned every location to its specific safety exit. Our Q-learning method’s simulation results demonstrate that an autonomous robot, on its own, finds the best path to the closest safety exit from any location it is based in case of emergency. The trajectory computed by the algorithm is obstacle-free. The emergency routine can be activated as soon as the central controller (PLC) sends an emergency signal.

An emergency stop (An ESTOP) button is a critical component of a plant. It is helpful to stop operations in case of emergency or disaster. It is usually wired on the central controller’s panel and bigger plants in different locations to maximize accessibility. Hence, the closer the operator to the ESTOP at the time of emergency, the quicker the response. We implemented the speech recognition algorithm to use a voice command as a stopping input for the plant if the operator is not close to the ESTOP. In this simulation, the word “emergency” is the command that activates the stopping process. We implemented the speech recognition algorithm directly from a python script running in a computer to a Siemens S7-1200 PLC connected in the same network without going via a third-party device such as a Raspberry Pi.

5.5 Research implications

We summarize the implications of our research as follows:

- *In society:* Some studies [59, 67] highlighted that safety measures and procedures in a manufacturing environment have a direct impact on the frequency of accidents in the plant, to the overall production performance (which in return affects the economy) and to the workers’ well-being. Operators are performant in an environment where they feel safe and comfortable working. Our research

offers additional practical strategies for manufacturing plants to enhance their safety procedures in an intelligent manufacturing scenario where human operators and intelligent equipment such as autonomous robots are at risk in emergencies.

- *In research:* On one hand, our study converts the theoretical concept of the Q-learning algorithm into a practical application directed in the area of safety where much more applications can be developed. On the other hand, we make a traditional controller (a PLC) more intelligent by enabling its ability to receive external voice commands using the Speech-recognition algorithm. The research area of speech recognition for PLCs is still quite open and presents many opportunities for the enhancement of traditional processes in an era of smart manufacturing. [67] states that there is a crucial need for more practical research in manufacturing safety in the way people and external factors or situations can influence the overall safety. Our study is a new addition to this research area.

5.6 Research limitations

Our research has the following limitations:

- We based our study purely on software simulations. It was not tested on a physical manufacturing factory. Our safety response mechanism is suitable for planning purposes before implementation to a real-world application.
- The experimental manufacturing environment we design to test our safety mechanism procedure is a simple locations overview of a small manufacturing factory. It does not reflect a real-world manufacturing plant. Its purpose is only to demonstrate that, using, our safety procedure, an autonomous robot can successfully learn an obstacle-free path to the closest safety exit.
- The voice command to the PLC is an enhancement to an existing ESTOP system. It should not be installed as the only emergency stopping mechanism.

6 Conclusion

In this research, we presented a safety response mechanism in the form of a safety induction for a moving autonomous robot by developing an experimental manufacturing environment through which the robot learns the best decision to make in case of an emergency that requires factory evacuation. Our platform allows the robot to generate a trajectory free of obstacles from its location at the time of evacuation to the closest safety exit. We implemented the Q-learning RL algorithm for the robot to grasp through trials and errors the best path to one of our experimental environment's

four safety standards. During the safety induction learning process, our safety exit locations have been given the best rewards than any other factory location. It pushes the robot to prioritize the safety exits as a preferred destination in case of emergency. The robot determines the path to its destination by choosing a location with the best rewards cumulating them until it gets to the exit. We also integrated a voice command, developed with a speech recognition system, as an additional stopping interlock in the plant PLC (a Siemens S7-1200). The voice command improves operators' responses when designated ESTOP buttons are not immediately accessible to them in emergencies. The voice command is captured by a microphone processed by a python script in a network computer connected to the S7-1200 Siemens PLC with no additional hardware. In an era of smart manufacturing, the inclusion of autonomous moving robots in production processes will considerably increase, and the need to keep robots safe from possible working environmental disasters will become a necessity. This study offers an effective safety response mechanism or safety induction for autonomous moving robots to allow them to evacuate safely from the factory whenever needed. The result of our simulation proves that using our safety procedure mechanism; the robot finds all the time the best route to the closest safety exit regardless of its current location.

For future works, our safety response mechanism could be adapted to applications wherein their regular routines; autonomous robots can identify the closest charging spots whenever their batteries are running low. Depending on the working environment's design, the robot could choose to go to the closest charging location instead of aborting its mission or deciding to go back to the source location.

Acknowledgements This research is supported partially by South African National Research Foundation Grants (No. 112108 and 132797) and Tertiary Education Support Program (TESP) of South African ES-KOM.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Bajic B, Rikalovic A, Suzic N, Piuri V (2021) Industry 4.0 Implementation Challenges and Opportunities: A Managerial Perspective. in *IEEE Systems Journal*, 15 (1): 546-559. <https://doi.org/10.1109/JSYST.2020.3023041>.
2. Ou X, Chang Q, Arinez J, Zou J (2018) Gantry Work Cell Scheduling through Reinforcement Learning with Knowledge-guided Reward Setting. in *IEEE Access*, 6: 14699-14709. <https://doi.org/10.1109/ACCESS.2018.2800641>.

3. Matheson E, Minto R, Zampieri EGG, Faccio M, Rosati G (2019) Human–Robot Collaboration in Manufacturing Applications: A Review. *Robotics*, 8 (100): 1-25. <https://doi.org/10.3390/robotics8040100>.
4. Hasan R, Asif Hussain S, Azeemuddin Nizamuddin S, Mahmood S (2018) An autonomous robot for intelligent security systems. in Proc. 9th IEEE Control Syst. Graduate Res. Colloq. (ICSGRC): 201–206.
5. Prianto E, Kim M, Park JH, Bae JH and Kim JS (2020) Path Planning for Multi-Arm Manipulators Using Deep Reinforcement Learning: Soft Actor-Critic with Hindsight Experience Replay. *Sensors (Basel)*, 20 (20): 1-22. <https://doi.org/10.3390/s20205911>.
6. Colgate JE, Edward J, Peshkin MA, Wannasuphprasit W (1996) Cobots: Robots for Collaboration with Human Operators. In Proceedings of the 1996 ASME International Mechanical Engineering Congress and Exposition: 433–439.
7. Muhammad J, Bucak IO (2013) An improved Q-learning algorithm for an autonomous mobile robot navigation problem. The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE): 239-243. <https://doi.org/10.1109/TAECE.2013.6557278>.
8. Zhao M, Lu H, Yang S, Guo F (2020) The Experience-Memory Q-Learning Algorithm for Robot Path Planning in Unknown Environment. in IEEE Access, 8: 47824-47844. <https://doi.org/10.1109/ACCESS.2020.2978077>.
9. Ribeiro T, Gonçalves F, Garcia I, Lopes G, Ribeiro AF (2019) Q-Learning for Autonomous Mobile Robot Obstacle Avoidance. IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC): 1-7. <https://doi.org/10.1109/ICARSC.2019.8733621>.
10. Peres RS, Jia X, Lee J, Sun K, Colombo AW, Barata J (2020) Industrial Artificial Intelligence in Industry 4.0 - Systematic Review, Challenges and Outlook. in IEEE Access, 8: 220121-220139, 2020. <https://doi.org/10.1109/ACCESS.2020.3042874>.
11. Shvets AA, Rakhlin A, Kalinin AA, Iglovikov VI (2018) Automatic instrument segmentation in robot-assisted surgery using deep learning. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA): 624–628.
12. Graves A, Abdel-rahman M, Geoffrey H (2013) Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing:6645–6649.
13. Xie Q, Zhou X, Wang J, Gao X, Chen X, Liu C (2019) Matching Real-World Facilities to Building Information Modeling Data Using Natural Language Processing. in IEEE Access, 7: 119465-119475. <https://doi.org/10.1109/ACCESS.2019.2937219>.
14. Wang D, Su J, Yu H (2020) Feature Extraction and Analysis of Natural Language Processing for Deep Learning English Language. in IEEE Access, 8: 46335-46345. <https://doi.org/10.1109/ACCESS.2020.2974101>.
15. Yu D, Deng L (2011) Deep Learning and Its Applications to Signal and Information Processing [Exploratory DSP]. *IEEE Signal Processing Magazine*, 28 (1): 145-154. <https://doi.org/10.1109/MSP.2010.939038>.
16. Babu VM, Krishna UV, Shahensha SK (2016) An autonomous path finding robot using Q-learning. 10th International Conference on Intelligent Systems and Control (ISCO): 1-6. <https://doi.org/10.1109/ISCO.2016.7727034>.
17. Wiedemann T, Vlaicu C, Josifovski J, Viseras A (2021) Robotic Information Gathering with Reinforcement Learning Assisted by Domain Knowledge: An Application to Gas Source Localization. in IEEE Access, 9: 13159-13172. <https://doi.org/10.1109/ACCESS.2021.3052024>.
18. Bae H, Kim G, Kim J, Qian D, Lee S (2019) Multi-Robot Path Planning Method Using Reinforcement Learning. *Appl. Sci.*, 9 (15): 1-16. <https://doi.org/10.3390/app9153057>.
19. Mannucci T, van Kampen E, de Visser C, Chu Q (2018) Safe Exploration Algorithms for Reinforcement Learning Controllers. in *IEEE Transactions on Neural Networks and Learning Systems*, 29 (4): 1069-1081. <https://doi.org/10.1109/TNNLS.2017.2654539>.
20. Varchacskaia P, Fitzpatrick P, Breazeal C (2001) Characterizing and processing robot-directed speech. in Proc. Int. IEEE/RSJ Conf. Human. Robot.: 229–237.
21. Nassif AB, Shahin I, Attili I, Azzeh M, Shaalan K (2019) Speech Recognition Using Deep Neural Networks: A Systematic Review. in IEEE Access, 7: 19143-19165. <https://doi.org/10.1109/ACCESS.2019.2896880>.
22. Erol BA, Majumdar A, Benavidez P, Rad P, Choo KR, Jamshidi M (2020) Toward Artificial Emotional Intelligence for Cooperative Social Human–Machine Interaction. in *IEEE Transactions on Computational Social Systems*, 7 (1): 234-246. <https://doi.org/10.1109/TCSS.2019.2922593>.
23. Krejsa J, Vechet S (2011) Mobile Robot Motion Planner via Neural Network. 17th International Conference on Engineering Mechanics, Engineering Mechanics 2011: 327-330.
24. Rahat SA, Imteaj A, Rahman T (2018) An IoT based Interactive Speech Recognizable Robot with Distance control using Raspberry Pi. International Conference on Innovations in Science, Engineering and Technology (ICISSET): 480-485. <https://doi.org/10.1109/ICISSET.2018.8745656>.
25. Jevtić A, Valle AF, Alenya G, Chance G, Caleb-Solly P, Dogramadzi S, Torras C (2019) Personalized Robot Assistant for Support in Dressing. in *IEEE Transactions on Cognitive and Developmental Systems*, 11 (3): 363-374. <https://doi.org/10.1109/TCDS.2018.2817283>.
26. Jung SW, Sung KW, Park MY, Kang EU, Hwang WJ, Won JD, Lee WS, Han SH (2013) A study on precise control of autonomous driving robot by voice recognition. *IEEE ISR 2013*: 1-3. <https://doi.org/10.1109/ISR.2013.6695640>.
27. Chan KY, Yiu CKF, Dillon TS, Nordholm S, Ling SH (2012) Enhancement of Speech Recognitions for Control Automation Using an Intelligent Particle Swarm Optimization. in *IEEE Transactions on Industrial Informatics*: 8 (4): 869-879. <https://doi.org/10.1109/TII.2012.2187910>.
28. Dhouchak D, Biban LK (2017) Applications of Safety in Manufacturing Industry. *International Journal of Scientific Research in Science, Engineering and Technology*, 3 (6): 498 -500.
29. Verl A, Lechler A, Schlechtendahl J (2012) Globalized cyber physical production systems. *Prod. Eng.*, 6 (6): 643–649.
30. Robla-Gómez S, Becerra VM, Llata JR, González-Sarabia E, Torre-Ferrero C, Pérez-Oria J (2017) Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments. in *IEEE Access*, 5: 26754-26773, 2017. <https://doi.org/10.1109/ACCESS.2017.2773127>.
31. Wang YC, Usher JM (2005) Application of reinforcement learning for agent-based production scheduling. in *Eng. Appl. Artif. Intell.*, 18 (1): 73–82.
32. Sutton RS, Barto AG (1998) *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press.
33. Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: A survey. *J. Artif. Intell. Res.*, 4(1): 237–285.
34. Jiang J, Kamel M, Chen L (2004) Reinforcement learning and aggregation. in Proc. IEEE Int. Conf. Syst., Man Cybern.: 1303–1308
35. Wang N, Hsu W (2020) Energy Efficient Two-Tier Data Dissemination Based on Q-Learning for Wireless Sensor Networks. in *IEEE Access*, 8: 74129-74136. <https://doi.org/10.1109/ACCESS.2020.2987861>.
36. Wicaksono H (2011) Q learning behavior on autonomous navigation of physical robot. 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI): 50-54. <https://doi.org/10.1109/URAI.2011.6145931>.
37. Weiss G (1999) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, 1st ed. Cambridge, MA, USA: MIT Press.

38. Zhou Z, Li X, Zare RN (2017) Optimizing chemical reactions with deep reinforcement learning. *ACS central science*,3(12): 337–344.
39. Mao H, Alizadeh M, Menache I, Kandula S (2016) Resource management with deep reinforcement learning. In *Proc. of ACM Workshop on Hot Topics in Networks*: 50-56.
40. El-Tantawy S, Abdulhai B, Abdelgawad H (2013) Multi-agent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto. in *IEEE Transactions on Intelligent Transportation Systems*, 14 (3): 1140-1150. <https://doi.org/10.1109/TITS.2013.2255286>.
41. Leurent E, Mercat J (2019) Social attention for autonomous decision-making in dense traffic. In *Machine Learning for Autonomous Driving Workshop at NeurIPS 2019*: 1-11.
42. Wang RY, Xu L (2012) Multi-Agent Dam Management Model Based on Improved Reinforcement Learning Technology. *Applied Mechanics and Materials*, (198–199): 922–926.
43. Hashimoto DA, Rosman G, Rus D, Meireles OR (2018) Artificial intelligence in surgery: promises and perils”, *Annals of surgery*, 268 (1): 1-17. <https://doi.org/10.1097/SLA.0000000000002693>
44. Gu S, Holly E, Lillicrap T, Levine S (2017) Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. *IEEE International Conference on Robotics and Automation (ICRA)*: 3389-3396. <https://doi.org/10.1109/ICRA.2017.7989385>.
45. Chen YJ, Chang DK, Zhang C (2020) Autonomous Tracking Using a Swarm of UAVs: A Constrained Multi-Agent Reinforcement Learning Approach. in *IEEE Transactions on Vehicular Technology*, 69 (11): 13702-13717. <https://doi.org/10.1109/TVT.2020.3023733>.
46. Nguyen H, La H (2019) Review of deep reinforcement learning for robot manipulation. in *Proc. 3rd IEEE Int. Conf. Robotic Comput.:* 590–595.
47. Su Y, Lu X, Zhao Y, Huang L, Du X (2019) Cooperative Communications With Relay Selection Based on Deep Reinforcement Learning in Wireless Sensor Networks,” in *IEEE Sensors Journal*, 19(20): 9561-9569. <https://doi.org/10.1109/JSEN.2019.2925719>.
48. Watkins CJ, Dayan P (1992) Technical note: Q-learning,” *Mach. Learn.*, 8 (3–4): 279–292.
49. Zhang Z, Xu JM (2005) A dynamic route guidance arithmetic based on reinforcement learning. *International Conference on Machine Learning and Cybernetics*, 6 : 3607-3611. <https://doi.org/10.1109/ICMLC.2005.1527567>.
50. S. Haykin (2001) *Neural Networks A Comprehensive Foundation*, Second Edition, Tsinghua Publishing House, Beijing.
51. Jat DS, Limbo AS, Singh C (2020) Chapter 11 - Speech-based automation system for the patient in orthopedic trauma ward”, in *Advances in ubiquitous sensing applications for healthcare, Smart Biosensors in Medical Care*, Chaki J, Dey N, De D, Eds: Academic Press: 201-214. <https://doi.org/10.1016/B978-0-12-820781-9.00011-5>.
52. Guo L, Zhou D, Zhou J, Kimura S, Goto S (2018) Lossy Compression for Embedded Computer Vision Systems. in *IEEE Access*, 6: 39385-39397. <https://doi.org/10.1109/ACCESS.2018.2852809>.
53. Liu H (2020) Chapter 1 – Introduction” in *Robot Systems for Rail Transit Applications*, Liu H, Eds: Elsevier: 1-36. <https://doi.org/10.1016/B978-0-12-822968-2.00001-2>.
54. Husnjak S, Perakovic D, Jovovic I (2014) Possibilities of Using Speech Recognition Systems of Smart Terminal Devices in Traffic Environment. *Procedia Engineering*, 69: 778-787. <https://doi.org/10.1016/j.proeng.2014.03>.
55. Sehr MA, Lohstroh M, Weber M, Ugalde I, Witte M, Neidig J, Hoeme S, Niknami , and E.A. Lee (2021) Programmable Logic Controllers in the Context of Industry 4.0. in *IEEE Transactions on Industrial Informatics*, 17(5): 3523-3533. <https://doi.org/10.1109/TII.2020.3007764>.
56. International Electrotechnical Commission (2017) *International Standard IEC 61131: Programmable Controllers*, 4th ed., Geneva, Switzerland.
57. Zhang Q, Lin J, Sha Q, He B, Li G (2020) Deep Interactive Reinforcement Learning for Path Following of Autonomous Underwater Vehicle. in *IEEE Access*,8: 24258-24268. <https://doi.org/10.1109/ACCESS.2020.2970433>.
58. Tang J, Song J, Ou J, Luo J, Zhang X, Wong K (2020) Minimum Throughput Maximization for Multi-UAV Enabled WPCN: A Deep Reinforcement Learning Method. in *IEEE Access*, 8: 9124-9132, 2020. <https://doi.org/10.1109/ACCESS.2020.2964042>.
59. Rajkumar I, Subash K, Raj Pradeesh T, Manikandan R, Ramaganesh M (2021) Job safety hazard identification and risk analysis in the foundry division of a gear manufacturing industry. *Materials Today: Proceedings*. <https://doi.org/10.1016/j.matpr.2021.02.326>.
60. Li L, Li S, Zhao S (2014) QoS-aware scheduling of services-oriented Internet of Things. *IEEE Trans. Ind. Informat.*, 10 (2): 1497–1505.
61. Da Xu L, He W, Li S (2014) Internet of Things in industries: A survey. *IEEE Trans. Ind. Informat.*: 10 (4): 2233–2243.
62. Oviatt S, Cohen P (2000) Perceptual user interfaces: multimodal interfaces that process what comes naturally”, *Commun. ACM*, 43 (3): 45-53. <https://doi.org/10.1145/330534.330538>
63. Gorecky D, Schmitt M, Loskyll M, Zühlke D (2014) Human-machine-interaction in the industry 4.0 era. In *12th IEEE INDIN: 289-294*. <https://doi.org/10.1109/INDIN.2014.6945523>.
64. Garcia MAR, Rojas R, Gualtieri L, Rauch E, Matt D (2019) A human-in-the-loop cyber-physical system for collaborative assembly in smart manufacturing. *Procedia CIRP*, 81: 600-605. <https://doi.org/10.1016/j.procir.2019.03.162>.
65. Li J, Deng L, Haeb-Umbach R, Gong Y (2016) Chapter 2 - Fundamentals of speech recognition”, in *Robust Automatic Speech Recognition*, Li J, Deng L, Haeb-Umbach R, Gong Y, Eds: Academic Press: 9-40, <https://doi.org/10.1016/B978-0-12-802398-3.00002-7>.
66. Dabous SA, Ibrahim F, Feroz S and Alsyouf I (2021) Integration of failure mode, effects, and criticality analysis with multi-criteria decision-making in engineering applications: Part I – Manufacturing industry. *Engineering Failure Analysis*, 122. <https://doi.org/10.1016/j.engfailanal.2021.105264>.
67. Hald KS (2018) Social influence and safe behavior in manufacturing. *Safety Science*, 109: 1-11. <https://doi.org/10.1016/j.ssci.2018.05.008>.