 DOI: 10.31276/VJSTE.2024.0019

Design and implementation assistant robot used in nuclear medicine: The first-step achievements

Tan Duoc Nguyen*, Thien Dang Hong, Le Dai Duong Nguyen

Department of Biomedical Physics, Nguyen Tat Thanh University, 300A Nguyen Tat Thanh Street, District 4, Ho Chi Minh City, Vietnam

Received 24 February 2024; revised 10 April 2024; accepted 20 May 2024

Abstract:

Nuclear medicine is a medical specialty that is increasingly gaining prominence. Although the radiation emitted by radiotracers enables visualisation of the functions of a patient's organs, it also poses risks to nearby individuals, particularly medical staff. The application of robots to assist or replace humans in such hazardous tasks offers a potential solution. This article presents the initial achievements of a project aimed at designing and implementing a robot for assistance in nuclear medicine. The robot is designed to facilitate transportation, provide audio-visual connections for telemedicine, and offer an interface for remotely interpreting survey meter readouts. Its robust and flexible movement system employs a differential steering mechanism. The software is based on a client-server architecture featuring multi-threaded processing and network-socket-based data exchange. The client-server design enables control of the robot via an online, cost-free console webpage. The radiation measurement interface, which integrates an Internet of Things (IoT) camera module with an optical character recognition (OCR) engine, efficiently incorporates conventional devices into a network platform. This article highlights the extensive efforts made to enhance both hardware and software design efficiency, alongside the initial achievements of the project.

Keywords: **assistant robot, Internet of Things, nuclear medicine, radiation measurement.**

Classification numbers: **2.3, 3.6**

1. Introduction

Nuclear medicine is a medical specialty that employs radioactive tracers for diagnostic imaging, cancer treatment, and *in vitro* testing [1]. Its use is becoming increasingly widespread. As of 2020, 134 of the 195 countries worldwide were employing nuclear medicine techniques [2]. In Vietnam, there are 41 clinics with nuclear medicine departments and 58 imaging scanners in operation [3]. Common nuclear medicine procedures include single-photon emission computed tomography (SPECT) with Tc-99m, positron emission tomography/computed tomography (PET/CT) with 18-FDG, and radiotherapy using I-131, P-32, and Y-90 radioactive microspheres.

The administration of radiotracers into patients or test samples is essential for these procedures. However, the radiation emitted by these sources can pose health risks to medical staff and the public if exposure exceeds safety thresholds. Additionally, these open radioactive sources can contaminate surfaces or individuals, leading to environmental pollution. Y.A. Khaled, et al. (2009) [4] conducted a study measuring radiation doses from patients treated with I-131 in hospital rooms. The dose rate at the patient's bedside was found to be 19.97±0.47 μGy/h, exceeding the 10 μGy/h safety limit for medical staff in the absence of protective equipment [5]. To mitigate these health risks, it is crucial to minimise exposure time [6] during the examination and care of patients in quarantine areas. The deployment of automation or remote-controlled robotic systems offers a promising solution.

The World Health Organisation (WHO) predicts that by 2030, the global healthcare workforce will need to increase to approximately 80.2 million - a 1.66-fold rise compared to 2013 - driven by population growth and climate change. Despite this, investment in healthcare human resources remains inadequate, particularly in developing countries [7]. As of 2020, WHO estimated that Vietnam required 99,351 doctors and 225,345 nurses [8]. However, by 2018, only 73,100 doctors and 135,845 nurses and midwives

*Corresponding author: Email: ntduoc@ntt.edu.vn

were available [9]. The COVID-19 pandemic has further exacerbated the shortage due to increased job resignations stemming from high-pressure conditions. Employing robotic systems to reduce work pressure on medical staff represents a sustainable approach to addressing these challenges.

Currently, robots in medicine are used primarily for the production of radioactive tracers, patient care and therapy, as well as surgical applications. For example, J.W. Brodack, et al. (1986) [10] explored the potential of laboratory robots in synthesising Fluorine-18 16α-Fluoroestradiol-17β for PET scanning, effectively reducing exposure time and radiation doses for staff. The THEODORICO 2 system by COMECER S.p.A. [11] automates the preparation of radioactive tracers for PET imaging, shielding staff from high-activity radiation sources. Similarly, G. Guevara, et al. (2015) [12] developed a robotic system for the preparation and dispensing of radioactive tracers. T. Alsennawy, et al. (2017) [13] designed a robot for transporting radioactive tracers within nuclear medicine departments, significantly reducing staff exposure.

In addition, robots also support patients and medical staff. In a review study, M. Kyrarini, et al. (2021) [14] listed 38 medical robot systems, of which 13 systems have been commercialised and 25 systems are in the research and testing stage. They are divided into 4 groups including patient care robots, nurse assistant robots, patient support robots, and rehabilitation therapy robots. Patient care robots aim to monitor the health and help elderly patients with daily tasks such as holding objects or eating. Robots also support the patient's spiritual life such as companionship, encouragement, or reminders of important tasks [14]. Typical examples of this type are the Pepper and Care-O-Bot 4 [15]. A nurse assistant robot is a robot designed to work under the control of a nurse. A robot assistant will act as a partner to help nurses reduce non-clinical tasks, so the nurses have more time to focus on taking care of patients [14]. In addition, assistant robots are also useful in jobs that require exposure to hazards such as radiation or dangerous pathogens. Typical robots of this type are Moxi [16] and ROBEAR [17].

Recognising the advantages of robotic systems, this project focuses on designing and implementing an assistant robot for nuclear medicine. The robot serves as a transporter for goods and medicine, a telemedicine support device, and a radiation measurement interface for survey meter data. This article presents the robot's hardware design, including dimensions, instrumentation, and schematic diagrams; its software architecture and algorithms; and performance evaluations from experimental trials. Improvement strategies are proposed in the discussion section.

2. Materials and methods

2.1. Hardware structure

The robot's frame and dimensions are illustrated in Fig. 1. It serves as the foundation for carrying and mounting other hardware and devices. The frame consists of a skeleton, a movement and steering unit, a power unit container, and a central control unit container. Table 1 provides the materials used to construct these components. The skeleton of the robot is constructed from steel bars and plates. The movement and steering unit are mounted at the base of the skeleton. The lowest houses the power unit container,

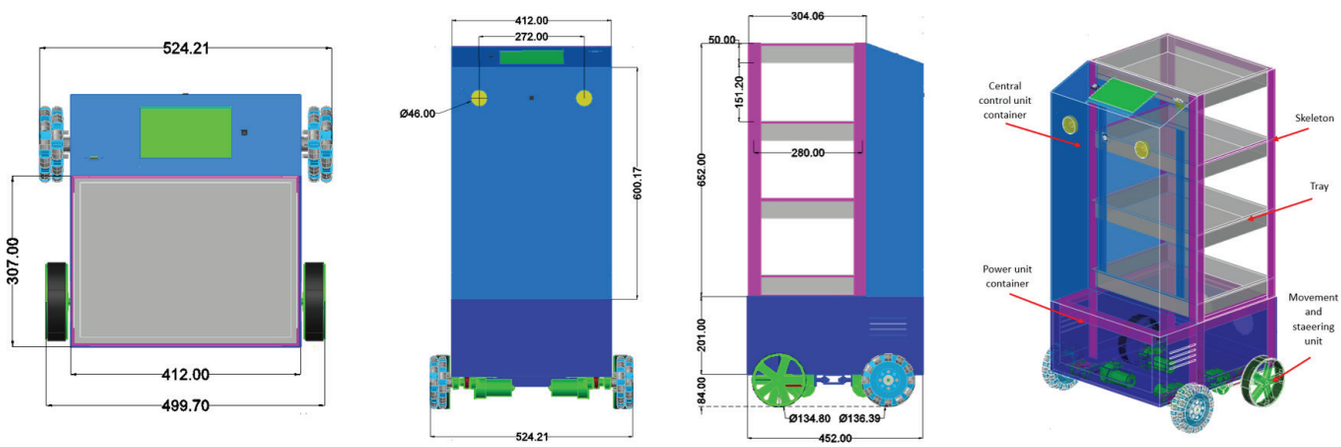

Fig. 1. The robot's frame and dimensions. The measurement unit is millimetre.

while the above are designated for carrier trays. These trays are machined from stainless steel for heavy loads, ease of cleaning, and compatibility with medical applications. The central control unit container is positioned at the front. This container is fabricated from aluminium composite panels selected for their lightweight properties and ease of machining. The container surfaces feature milled holes to accommodate the installation of peripheral devices.

Table 1. Materials used to make the robot's frame sourced from local suppliers.

Components	Materials	Technical specifications
Skeleton	Equal angle bar steel	30.0x30.0x3.0 mm, alloy steel
	Plate steel	3.0 mm thickness, alloy steel
Central control unit/ power unit container	Aluminium composite panel	3.0 mm thickness, 2 sides of 2x0.1 mm aluminium
Trays	Stainless steel	SUS 304, 1.5 mm thickness

Figure 2 illustrates the configuration and positioning of the movement and steering unit (MSU). All components are mounted to the base of the skeleton. The unit consists of two regular wheels (145 mm diameter) and two omni wheels (139 mm diameter), which are attached to the skeleton using Φ10 mm axles with bearing-mounted supports. The wheels are powered by four independent 24 VDC gear motors, each with a maximum power of 60 W, a maximum torque of 62.5 kgf·cm, and a rotation frequency of 110 rpm. Power transmission from each motor to corresponding wheel is achieved via a system of pulleys and belts. The primary pulley has a smaller diameter than the secondary pulley, resulting in reduced wheel angular speed (ω_{wh}) and increased pulling force:

$$\omega_{wh} = \frac{d_{pl_1}}{d_{pl_2}} \times \omega_{mt} = \frac{d_{pl_1}}{d_{pl_2}} \times 2\pi f_{mt} \quad (1)$$

where d_{pl_1}, d_{pl_2} are diameters of the primary and secondary pulley, respectively; ω_{mt}, f_{mt} are angular speed and rotation frequency of the motor, respectively. So, the forward speed of the wheel (equally the speed of the robot) (v) can be calculated by:

$$v = \frac{d_{wh}}{2} \times \omega_{wh} = \pi \times \frac{d_{wh} \times d_{pl_1}}{d_{pl_2}} \times f_{mt} \quad (2)$$

where d_{wh} is the wheel diameter. With the chosen parameters, the robot's maximum speed is approximately 40 cm/s, which is one-third of a typical human walking speed [18]. This speed was selected to ensure safe and manageable operation.

The robot employs a skid-steering mechanism with four independent wheels. Unlike the Ackerman mechanism,

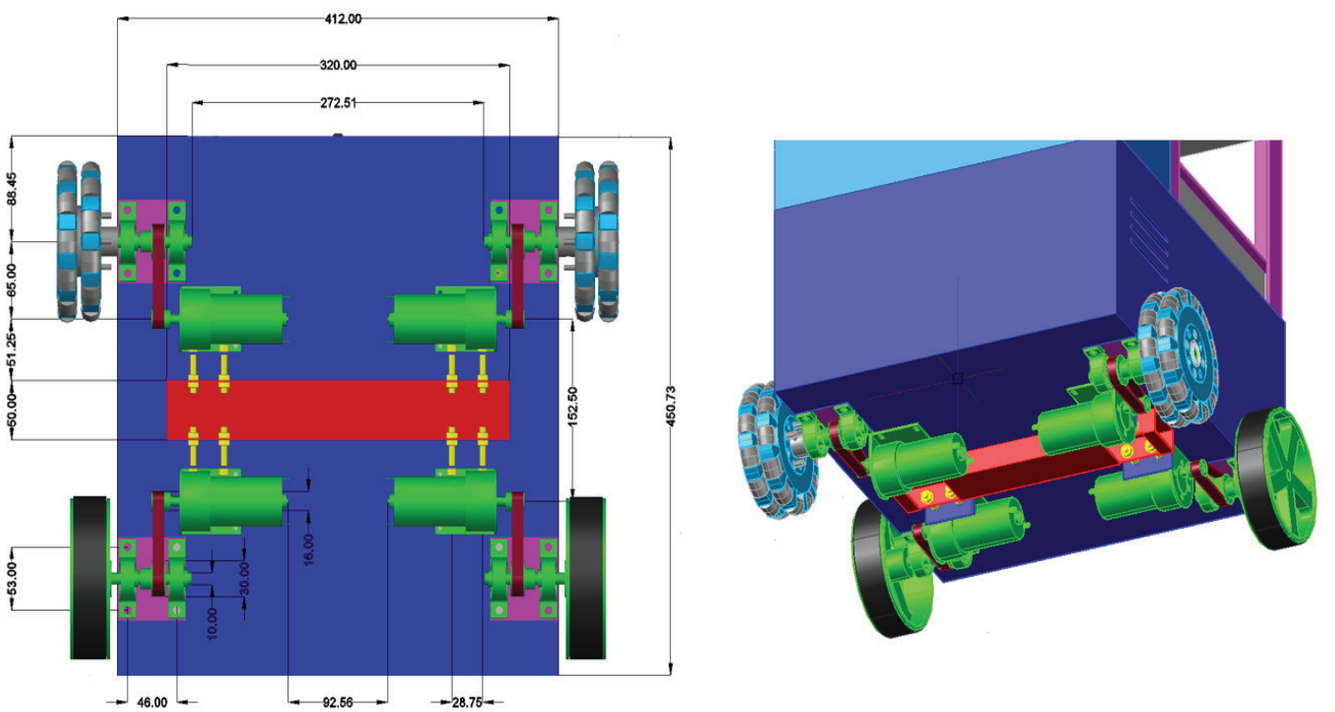

Fig. 2. Movement and steering unit appearance and dimensions. The measurement unit is millimetre.

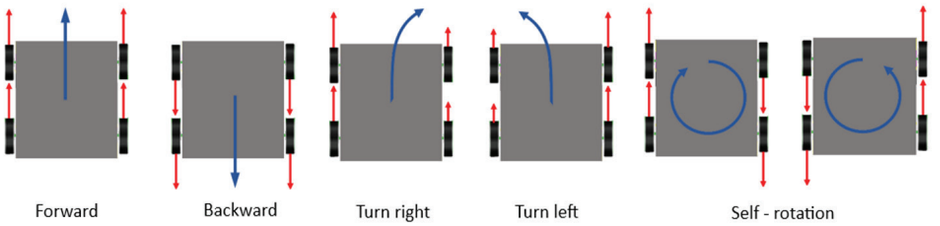

Fig. 3. Navigation of the robot. The red arrows demonstrate the velocities of each wheel. The blue arrows demonstrate the direction of the robot.

which alters the angles of the front wheels relative to the direction of motion, the skid-steering keeps all wheels fixed to the frame. The robot's navigation is determined by the differences in speed and rotation direction between the wheels. Fig. 3 demonstrates this kind of navigation. The robot could move forward or backward, turn left or right, and specialty rotates around its self-axis. These movements are managed by firmware via a microcontroller. The self-rotation capability allows the robot to manoeuvre flexibly in hospital lobbies and to enter or exit patient rooms with ease.

This robot utilises a centralised processing architecture for its control unit [19]. Fig. 4 presents a block diagram of the central control unit (CCU) and its peripheral devices.

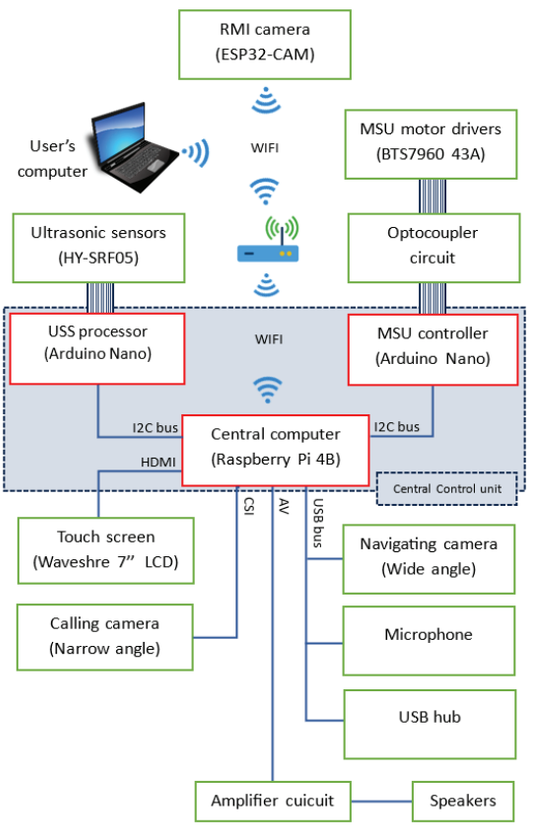

Fig. 4. Block diagram of the robot. The central control unit contains elements inside the dashed polyline.

At the core of the CCU is a central computer based on the Raspberry Pi 4B single-board computer. The Raspberry Pi connects to two Arduino Nano circuits via a dedicated I²C bus. These circuits function as controllers for the MSU and processors for ultrasonic sensors (USS). This configuration offers significant advantages by protecting the central computer from electrical interference that may propagate from peripheral devices. Additionally, it provides a safeguard against electrical shocks that could arise from short circuits in any of the connected analogue devices.

The MSU controller executes movement and steering requests from the central computer by managing four BTS7960-43A motor drivers. Each driver is connected to a DC gear motor in the MSU through an electrical interference filter circuit. The Arduino controls each driver via three pins that are isolated by an optocoupler circuit.

Another Arduino Nano, also on the I²C bus, works as the ultrasonic sensor (USS) processor. Four HY-SRF05 sensors are located on the robot's four sides. The Arduino Nano handles these sensors, calculates the distances between the sensors and obstacles, and periodically sends the distance data to the central computer. Each sensor is connected to the processor via two pins.

The radiation measurement interface (RMI) camera is an integral component of the RMI. It is based on an ESP32-CAM microcontroller with a built-in OV2640 camera. This circuit operates independently and connects to the central computer via a Wi-Fi connection. The camera captures images of the Radiation Alert Ranger® survey meter screen and streams them to an optical character recognition (OCR) processing module in the central computer. Using the OCR method, the meter readout can be recognised without relying on the manufacturer's software or hardware. To securely position the camera on the survey meter screen, a holder was designed and fabricated using 3D printing. The holder is made from acrylonitrile butadiene styrene (ABS) thermoplastic. Fig. 5 shows the holder dimensions and the assembled device.

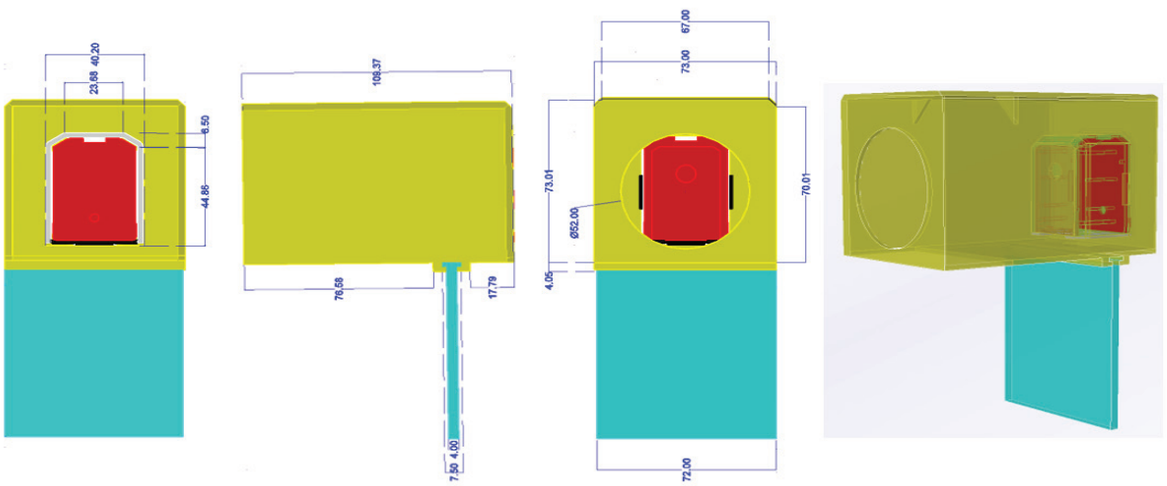

Fig. 5. The radiation measurement interface device and dimensions. The measurement unit is millimetre.

All off-the-shelf components and devices used in the robot's hardware are listed in Table 2.

Table 2. Components and devices used in the robot.

Components	Model/technical specifications	Amount
Touch screen	Waveshare 7" HDMI Capacitive Touch Screen LCD©	1
Raspberry Pi SBC	Raspberry Pi 4 Model B 1 GB RAM	1
Arduino nano	Arduino Nano CH340	2
I^2C voltage-level translator	Bi-directional 3V3-5 V	1
Audio amplifier	PAM8610 10W Hifi 2.0 Class D	1
Ultrasonic sensor	HY-SRF05	4
DC-DC buck converter	Input 7-32 VDC; output 0.8-28 VDC; 100 W - 12 A	2
Motor driver DC	BTS7960 43A high-power	4
USB hub	Type-C USB input, 4 ports type-A USB output	1
Mini USB microphone	100-16 KHz, -67 dBV/pBar, -47 dBV/Pascal +/- 4 dB	1
Camera (calling mode)	OV3660 3 MP USB 110 degree/2.1 mm	1
Camera (navigating mode)	OV5647 for Raspberry Pi 200 degree/3.6 mm	1
Speaker	8 Ohm - 5 W	2
ESP-32 CAM	Ai-Thinker ESP32-S	1
Gear motor	DCM50-775 24 VDC 110 rpm 116 K	4
Battery	12 V - 20 Ah	3
Battery-charged controller	XH-M604	2
Transformer	220 V - 12 V/24 V - 10 A	1
Omni wheel	Φ136 mm, Φ 10 mm of axel	2
Friction wheel	Φ145 mm, aluminium, Φ10 mm of axel	2

2.2. Software architecture

Figure 6 describes the robot's software structure, which includes three layers of software: the user layer, the server layer, and the firmware layer. Each layer is deployed on separate CCU hardware and contains several modules.

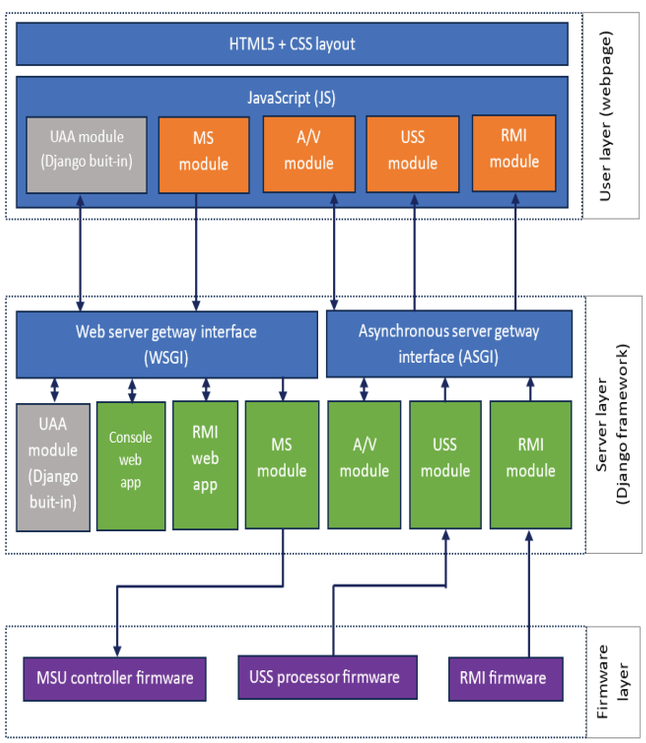

Fig. 6. The robot software structure. Abbreviations: UAA: user authentication and authorisation; MS: movement and steering; A/V: audio and video; USS: ultrasonic sensor; RMI: radiation measurement interface; MSU: movement and steering unit. Arrows indicate connections and data flow.

The user layer runs on a client computer, which acts as the controlling computer and connects to the central computer via a Wi-Fi connection. It is a web-based application that is laid out by HTML5/CSS and interacts with the server layer by JavaScript. This layer contains five modules: UAA, MS, AV, USS, and RMI. UAA is a Django built-in module that is responsible for user account creation, authorisation, and user authentication. MS module is a front-end event handler that constantly listens for navigation buttons or keyboard presses to generate movement and steering requirements. While the UAA and MS modules interface with the server layer via WSGI, the AV, USS, and RMI modules exchange data with the principal via ASGI. AV module exchanges data with the server layer in both sending and receiving directions. USS module only listens to events to obtain distance data. RMI module is responsible for receiving radiation readout data and deployed on a separate webpage.

The server layer operates on the central computer and is implemented using the Django web framework. Django provides two network interfaces: the Web Server Gateway Interface (WSGI) and the Asynchronous Server Gateway Interface (ASGI). WSGI serves only HTTP requests. The connection between client and server is created, used for request or response exchange, and terminated immediately after completion. WSGI is suitable for low data flow and non-real-time processes but performs poorly for high-flow exchanges such as AV, USS, and RMI data due to frequent HTTP connection creation and termination. ASGI, on the other hand, establishes a persistent connection (referred to as a channel in Django or a WebSocket in JavaScript) at the start of a session. This connection is used for data exchange throughout the session and is terminated only when the client or server ends the session. This channel is highly efficient for high-flow data, making it ideal for AV, USS, and RMI data. Beneath these interfaces are web applications and modules corresponding to those in the user layer.

The firmware layer, the lowest level of the robot's software, is installed on the Arduino and ESP32-CAM circuits. Acting as intermediaries between analogue devices and the digital server layer, these circuits are programmed in C++. The MSU controller firmware listens to movement and steering requests from the central computer and resolves them into a set of controlling signals on Arduino pins to execute the required movement direction. The USS processor firmware periodically checks four ultrasonic sensors to obtain distance data and then sends it to the central computer. The RMI camera firmware, installed in the ESP32-CAM, controls the built-in camera to capture image frames of the survey meter screen and transmit them to the central computer. By deploying controlling firmware on separated microcontrollers, the central computer retains more resources for higher-level processes.

2.3. Algorithms of the main modules

2.3.1. Moving and steering modules

The MS modules consist of components at the client, server, and firmware layers as well. Their primary function is to listen to and resolve user navigation requests. Fig. 7 illustrates this process step by step. The user initiates a command by pressing an arrow on the console or keyboard. These actions trigger events on the EventListener. A FormData object is created containing the necessary data and then transmitted to the server layer via the HTTP protocol. At the server layer, the WSGI receives the HTTP packages, invokes the resolving functions to process the data, and translates them into commands for the I²C-bus object. This object forwards the commands to the corresponding module on the MSU firmware. Commands are queued in a stack, where they are processed sequentially by the Executing functions.

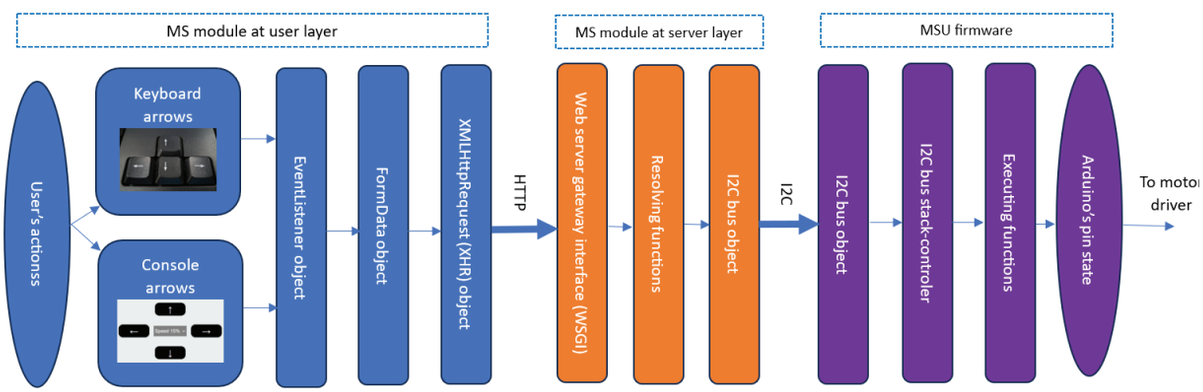

Fig. 7. Moving and steering module algorithms. MS: movement and steering; MSU: movement and steering unit.

In this process, the MS module at the server only forwards user commands to the MSU firmware. Most of the processing and hardware control tasks are handled by the MSU processor (Arduino), which reduces the load on the central computer. This approach optimises processing times and enhances the system's real-time performance. However, the Arduino's limited resources can become overwhelmed if too many command packages accumulate in the I²C buffer, potentially causing firmware crashes. This issue is mitigated by the I²C-bus-stack controller that ensures the stack length remains below a defined limit by deleting outdated commands. This first-in-first-out (FIFO) management reduces latency between the user's command and the robot's actual movement.

2.3.2. Audio and video modules

The AV modules facilitate the capture, exchange, and rendering of audio and video data between the robot and the user. Fig. 8 presents the flowchart for this process. At the user layer, Web Audio APIs [20] are used to handle audio streams while Canvas object and Web APIs are utilized for video processing. At the server layer, audio data is managed using the PyAudio API [21], and video processing is primarily handled by the OpenCV-Python API (CV2) [22]. The input and output streams interface with the corresponding processes at the user via ASGI. These streams are handled on separate threads on the central computer for concurrent processing. Python multi-threading help to optimise the system's real-time performance because the AV modules interact with devices such as cameras, screens, microphones, and sound cards, which require time to sample or render data. In addition, other processes, such as constructing/opening data URLs and decoding/encoding images, are also time-intensive.

Audio waveform data is sampled at microphones in sequential packages (called chunks) at a common sampling rate (f_s) of 44.100 frames per second. The interval of each chunk (T_{ch}) is calculated by:

$$T_{ch} = \frac{1}{f_s} \times L_{ch} \tag{3}$$

where L_{ch} is the length of a chunk or number of frames per chunk. A chunk is generated and will be fed to the appropriate process at frequency:

$$f_{fd} = \frac{1}{T_{ch}} = \frac{f_s}{L_{ch}} \tag{4}$$

The higher f_{fd}, the lower latency and smoother the sound, but the load on the central computer increases. Vice versa, a low f_{fd}, however, decreases the load but worsens the sound quality. To balance sound smoothness, real-time performance, and system efficiency, L_{ch} should be 2048 frames per chunk.

2.3.3. Ultrasonic sensors modules

The USS modules manage ultrasonic sensors, acquire distance data, transfer these data over ASGI/WebSocket, and render the results on the USS radar displayed on the console screen. Fig. 9 illustrates the data flow. The GetDistance function on the firmware layer controls the ultrasonic sensors and calculates the distances. The distance data is sent to the server layer via the I²C bus. The Controlling function passes this data to ASGI and regulates the frequency of data updates from the USS firmware. Periodically, it sends a request to the firmware to obtain distance data, at which point the GetDistance function is executed. Since the I²C bus is shared between the USS and MSU processors, careful control is required to ensure system performance. Distance data is updated twice per second, which provides a balance between real-time responsiveness and system efficiency. Like other ASGI-based modules, the USS module at the server layer operates on a separate thread.

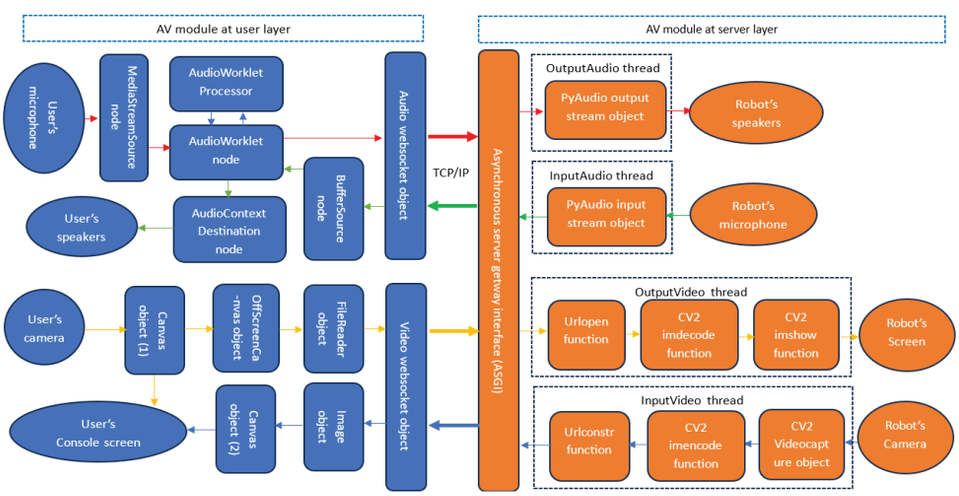

Fig. 8. Audio and video module algorithms.

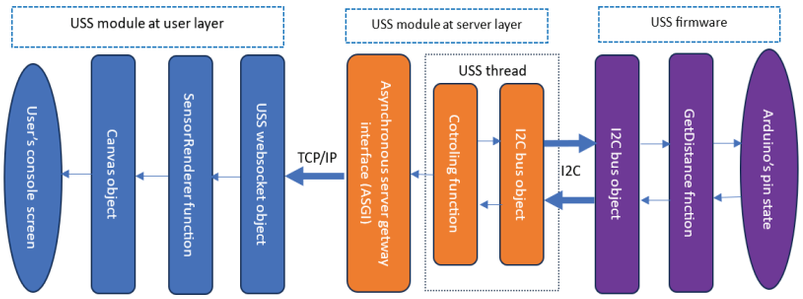

Fig. 9. Ultrasonic sensors module algorithms.

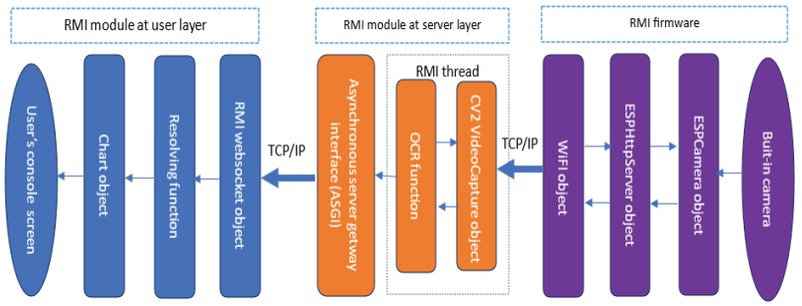

Fig. 10. Radiation measurement interface module algorithms.

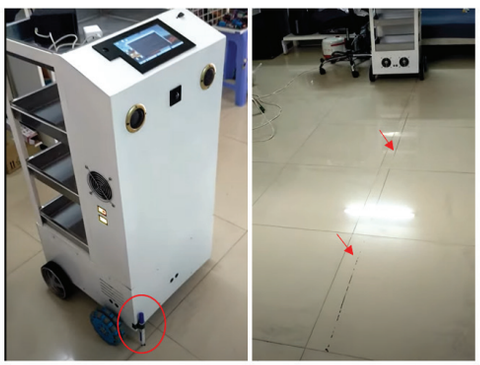

Fig. 11. Horizontal shift testing.

2.3.4. Radiation measurement interface modules

The RMI modules serve as an interface for capturing the readouts from a Radiation Alert Ranger® survey meter, extracting the values using an OCR algorithm, and displaying the results on a webpage as a time-based chart or recording them in a CSV file. Fig. 10 illustrates the step-by-step process. On the ESP32-CAM circuit, a mini server is implemented to stream image frames from the ESPCamera object to the RMI module at the server layer. At this layer, the RMI module operates on a separate thread to process the incoming data. The CV2 API is utilised to capture the stream and sample image frames at 1 frame per second to avoid redundant calculations from identical frames. The sampling frames are then fed to the OCR function to extract the readouts. The Tesseract OCR engine [23] is the core of this function. To optimise Tesseract's performance for digit-only images, custom-trained data provided by Shreeshrii (2016) [24] is used. To enhance OCR quality, pre-processing is required [25]. It involves fundamental image processing techniques, including subtraction, cropping, padding, thresholding, and morphological operations. These steps improve image contrast, reduce noise, and isolate the value-containing area from the background. At the end of the OCR process, only the results with confidence factors higher than 90% will be accepted. This data is sent to the user layer via ASGI. At the user layer, the Resolving function processes the data and forwards it to the Chart object, which leverages the Chart.js API [26] to render the data on the user screen.

2.4. Performance evaluation

2.4.1. Robot performance testing

The performance testing was conducted in an office environment. The trial scenario included the robot's navigational motions, movement in narrow spaces, support for doctor-patient remote conversations, and assistance with radiation measurements. Most of the testing components were evaluated visually. The linearity of forward and backward motions was assessed by measuring the horizontal shifts (relative to the movement direction) between the start and end points along a path drawn on the floor (Fig. 11). The test distance for each trial was 3.0 m.

2.4.2. Radiation measurement interface precision testing

The precision of the OCR function is the primary determinant of the RMI's accuracy. Therefore, this testing focused on the OCR function. Each time the OCR function was executed, the original image frames, embedded with their corresponding OCR values, were stored on the central computer. These images were manually analysed by comparing the meter readouts with the OCR-extracted values to classify them into a true group and a false group. Precision was calculated as the ratio of the size of the true group to the total sample size. The sample size (n) is calculated using the formula [27]:

$$n = \frac{t_{\alpha}^2 \times p(1-p) \times N}{(N-1) \times \alpha^2 + t_{\alpha}^2 \times p(1-p)} \quad (5)$$

where N is the population size, p is the expected proportion, α is the significance level, and t_{α} is the normal distribution value corresponding to the confident interval ($CI=1-\alpha$). The Radiation Alert Ranger® screen can display a maximum of 4 digits (0 to 9) and 3 dots (one or none visible). Therefore, the readout population size (N) is 10^4 in non-dot measured units (cumulative counts - CC, counts per minute - CPM…), or 3×10^4 in dot measured units ($\mu Sv/h$, counts per second - *CPS*, *mR/h*,…). p is chosen to be 0.5 by assuming the same proportion of the true and false cases. *CI* is chosen to be 95%. Finally, the minimum sample size is n_1=370 for the non-dot case and n_2= 380 for the dot case.

For the non-dot case, a systematic sampling method is employed. Although multiple units exist within this category, the OCR function processes only the numerical digits of the readout, disregarding the unit itself. To simplify systematic sampling and ensure that the sample is independent and identically distributed (IID), the cumulative counting (CC) unit is chosen. A Cs-137 source with an activity of 180.61 µCi (decay corrected at the experiment date) is used to generate the counts. Fig. 12 illustrates the experimental setup. For the dot case, a convenience sampling method is utilised. Although this is a non-probability sampling method that does not yield an independent and identically distributed (IID) sample, it was chosen to avoid extensive time-consuming due to the large of possibilities. Two samples are drawn in each case to calculate the mean and standard deviation of the precisions.

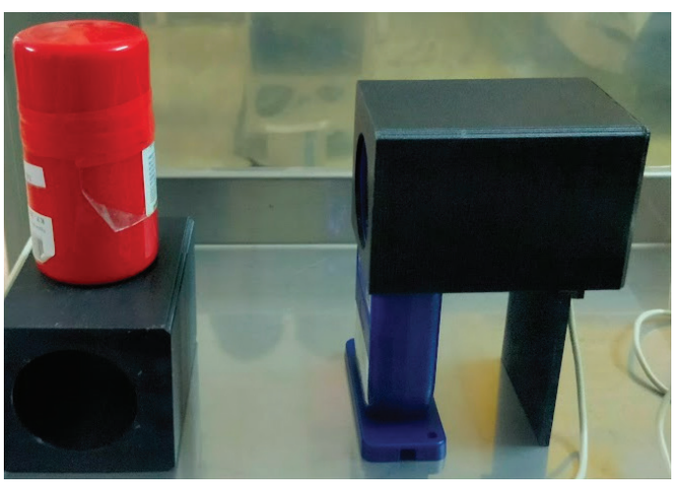

Fig. 12. Testing radiation measurement interface with a Cs-137 source.

3. Results and discussion

Figure 13 shows the robot's appearance, and Table 3 outlines its technical specifications. The robot utilises a skid-steering mechanism with four motorised wheels, providing strength, stability, and flexible movement. It can move forward or backward at a maximum speed of 40 cm/s and self-rotate with a minimum turning radius of only 43 cm. These features enable it to manoeuvre easily in narrow spaces such as lobbies and patient rooms. The robot is manually operated by staff using an intuitive online console (Fig. 14). The console is a webpage accessible via any computer, tablet, or smartphone equipped with a standard web browser. The robot is equipped with four stainless steel trays, which can be used to transport daily necessities such as meals, water, clothes, and medicines into radiation quarantine areas to support patients. Fig. 15 demonstrates the robot performing transportation tasks. In some cases, it can also transport radiopharmaceuticals between a hot lab and an administering room. As a telemedicine support device, the robot facilitates communication and examination between patients and medical staff. It is equipped with a narrow-angle camera to capture patient video and audio, which are transmitted to the console. Similarly, the doctor's video and audio can be streamed to the robot's screen and speakers during interactions with patients.

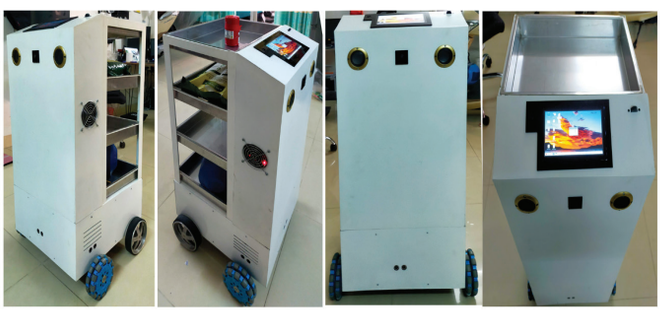

Fig. 13. The robot's appearance.

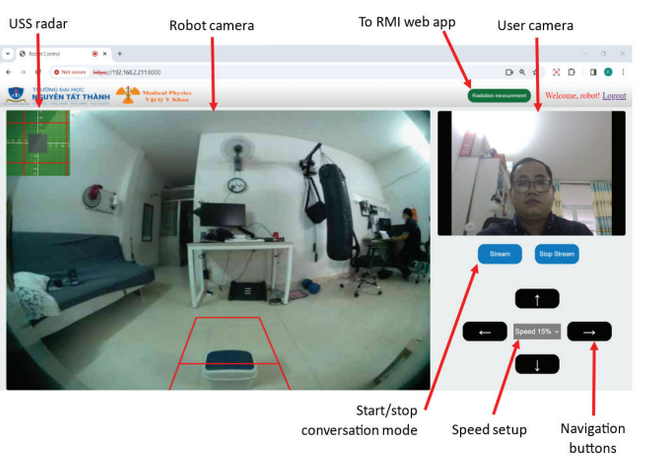

Fig. 14. Robot console in navigation mode.

Table 3. The robot's technical specifications.

Dimensions (LxWxH)	45.5x41.5x95.5 cm^3
Ground clearance	10.5 cm
Weight	45 kg
Max load	50 kg
Tray	4 pieces
Motor max torque (at 24 V)	62.5 kgf.cm
Motor max rotation frequency	110 RPM
Wheel diameter	14.5 cm (rears); 13.9 cm (fronts)
Max speed	40 cm/s
Min turning radius	43.0 cm
Horizontal shift per meter of path	2.21±1.44 cm (forward motion) 4.58±1.86 cm (backward motion)
Camera resolution	5 MP
Camera view-angle	200° (navigating); 160° (calling)
Screen	1024x600 px; 16.5x12.4 cm^2 (7"); capacitive touchscreen
Collision sensor	4 x Ultrasonic sensor; Measurement range: 3-50 cm; Response time: ~0.5 sec
Speaker	2x8 Ω/5 W
Power supply	220 V - 50 Hz
Wireless	WIFI 2.4 GHz/5 GHz IEEE 802.11ac; Bluetooth 5.0, BLE
Peripheral port	USB 3.0
Radiation measurement interface	Available for Radiation Alert Ranger ®; Response time: ~1.0 sec

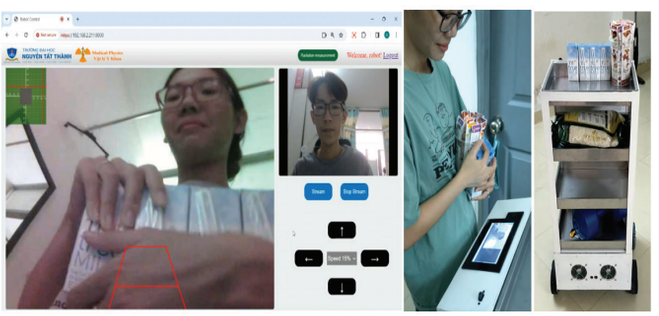

Fig. 15. The robot carrying food to a quarantine area to serve a patient. The staff and patient are conversing via robot.

Additionally, the robot can carry medical equipment such as thermometers, sphygmomanometers, and cardiographs, allowing patients to use these under tele-instruction. This enables the collection of basic health indices. Patients can also collect samples for testing under staff guidance, and the robot can transport these samples to a laboratory. This setup allows medical staff to perform a wide range of telemedicine tasks with minimal effort, increased convenience, and complete protection from radiation exposure.

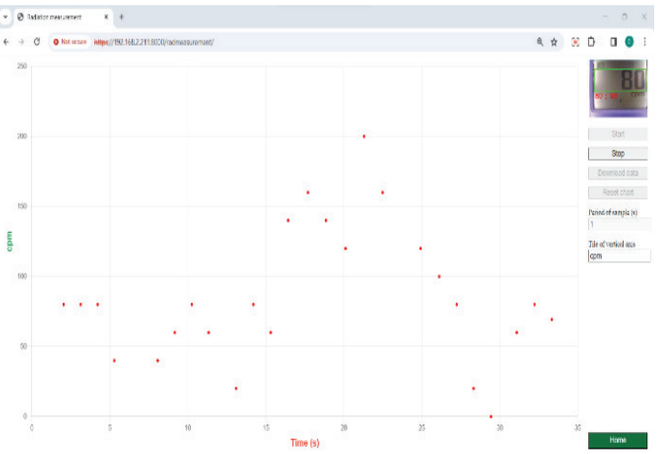

Fig. 16. The radiation measurement interface web app.

Figure 16 shows the RMI web app. The scatter graph displays radiation measurement data over time, functioning as a live chart that updates data points in real-time. Additionally, a frame presents the meter screen along with the corresponding OCR result. After the measurement, users can download a CSV file containing the recorded data. This feature allows users to obtain measurement results without waiting on-site, ensuring they remain free from radiation exposure. This functionality is particularly valuable for radioactive pollution inspections and monitoring the remaining activity of patients.

Table 4. Radiation measurement interface precision testing result.

	Non-dot case		Dot case	
	Sample 1	*Sample 2*	*Sample 1*	*Sample 2*
Sample size	421	395	380	380
False group size	9	15	210	99
True group size	412	380	170	281
Precision	97.86%	96.20%	44.74%	73.95%
Mean	97.03%		59.34%	
Standard deviation	1.17%		20.65%	

Table 4 shows the results of the RMI precision testing. The non-dot case achieved a good precision of (97.03±1.17)%, while the dot case exhibited a lower precision of (59.34±20.65)%. Fig. 17 demonstrates typical false OCR results. There are three types of failures for both cases. (A) occurs at the pre-processing step that over-detects a digit contour causing digit duplication. (B) occurs when the original images are unclear such as when images are captured during the transition of the readout. (C) is a failure related to the Tesseract's recognition ability. It may be improved by training the engine with relevant data.

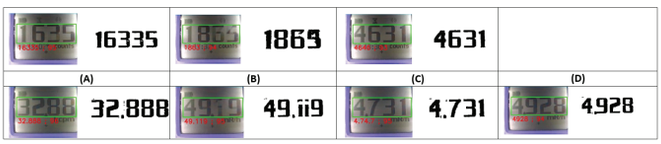

Fig. 17. Typical optical character recognition failure results. The first row is the non-dot case, the second row is the dot case.

However, re-training Tesseract is complicated and cost-consuming. Another failure for the dot case is (D) dot under-recognition. This occurs when dots are linked to adjacent digits, forming a single block on the image. Since the positions of the dots are fixed on the meter screen, a simpler solution is to verify these positions directly to determine whether a dot is present. Additionally, using a higher-resolution camera to produce higher-quality images could resolve most of these issues.

During the trial operating scenario, all functions generally performed as per the planned design. Video demonstrating the robot's operation can be found at https://doi.org/10.5281/zenodo.14826115. The robot executed all navigational motions smoothly. The horizontal shifts per metre of path were 2.21±1.44 cm for forward motion and 4.58±1.86 cm for backward motion. There was no significant difference between the horizontal shifts of these motions ($p=0.09$) at a significance level of 0.05. The horizontal shift reflects the robot's deviation from the planned path. Over longer paths, the deviation becomes more pronounced, which may be attributed to several factors, such as (1) Differences in friction forces between the four wheels and the floor, (2) Variations in the performance of the four gear motors, and (3) Inconsistencies in the toe angles of the wheels. Using feedback from sensors to dynamically adjust the path during movement [28] may solve the first issue. To address the second issue, offsets can be applied to slower motors to synchronise their speeds. For the third issue, fine adjustments to the toe angles may help.

Although the latency of navigation, as well as AV signals, has not been precisely measured, it is acceptably low based on visual evaluations. During the trial operation, users were generally able to control the robot's navigation comfortably and engage in effective conversations. On the other hand, the response time of the USS and RMI modules are theoretically estimated using the corresponding parameters of the algorithms. The response times are 0.5 and 1.0 sec, respectively. In the trial operation, they are around a second. Latency is the most important factor of a real-time system. Many techniques have been deployed as mentioned in each algorithm focusing on using multi-threading, APIs in the JavaScript runtime environment, multi-layer software architecture, optimisation of processing steps and parameters, and utilising raw data. For example, experience with optimising audio latency has shown that transmitting raw data (e.g., audio waveform data) via network sockets and rendering it at the destination without encoding or decoding (e.g., avoiding data URL generation and parsing) significantly reduces latency.

Originally designed for teaching computer science in schools, the Raspberry Pi has increasingly been applied across diverse fields [28, 29]. Our research demonstrates the feasibility of using compact and cost-effective microcontroller circuits such as Raspberry Pi and Arduino to develop assistant robots for medical applications. This project aligns with recent research on using Raspberry Pi for smart healthcare monitoring systems [30] and healthcare digitalisation systems [31, 32].

The robot developed in this project offers significant potential for future enhancements. Autonomous navigation and AI-based algorithms could be implemented on the existing MSU and CCU systems to enable independent operation. As the differential steering mechanism allows precise control of wheel speed and direction, automating the robot's movement would primarily involve software updates rather than hardware modifications. Such upgrades would, however, require additional sensors, such as LiDAR (light detection and ranging), to model surrounding obstacles [33], along with using the existing USS sensor system [34]. Artificial intelligent algorithms could further optimise path planning, obstacle avoidance, stability, efficiency, and safety [28, 35]. The robot's server-client architecture and IoT capabilities provide a foundation for building a healthcare monitoring system. Medical devices such as thermometers, sphygmomanometers, and cardiographs can generate data that is transmitted to medical staff via the ASGI/WebSocket protocol. The modular software design allows new modules to interface with additional medical devices, each operating independently on separate threads to avoid interference with existing functions. The RMI module's first-step achievements highlight the feasibility of interfacing with peripheral devices. Combining the RMI module with autonomous navigation could enable the creation of radiation exposure maps for nuclear medicine departments or hospitals [36]. Additionally, the IoT-based architecture of RMI devices allows them to be attached to multiple survey meters across a hospital, facilitating a centralised radiation monitoring system.

4. Conclusions

This article presents the initial achievements of designing and implementing an assistant robot for nuclear medicine. The robot integrates hardware and embedded software, featuring a movement and steering unit with omni-wheels and a skid-steering mechanism for flexible and efficient transportation in nuclear medicine departments. The robot also serves as telemedicine support, enabling AV communication between doctors and patients. The integrated radiation measurement interface enhances its utility for inspection tasks. Moreover, the robot's expandable software capabilities make it suitable for future developments, including autonomous navigation, AI integration, and interfacing with medical devices. These features also position it for use in autonomous radiation measurement and centralised monitoring systems.

CRediT author statement

Tan Duoc Nguyen: Literature review, Conceptualisation, Constructing hardware, Programming software for CCU and USS, Testing and Debugging, Data analysis, Writing - Reviewing and Editing; Thien Dang Hong, Le Dai Duong Nguyen: Conceptualisation, Constructing hardware, Programming firmware for MSU and Console webpage, Testing, and Image preparation.

ACKNOWLEDGEMENTS

This research is funded by Nguyen Tat Thanh University, Ho Chi Minh City, Vietnam under grant number 2023.01.119/HD-KHCN.

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] S.R. Cherry, J.A. Sorenson, M.E. Phelps (2003), *Physics in Nuclear Medicine*, Saunders, 523pp.

[2] D. Paez, F. Giammarile, P. Orellana (2020), "Nuclear medicine: A global perspective", *Clinical and Translational Imaging*, **8(2)**, pp.51-53, DOI: 10.1007/s40336-020-00359-z.

[3] A.T. Hoang, N.D. Do (2022), "Assessing the current status and prospects of nuclear medicine technology application in Vietnam", *Vietnam Journal of Science and Technology - MOST*, **64(1)**, pp.59-64, DOI: 10.31276/VJST.64(1).59-64.

[4] Y.A. Khaled, A. Sandouqa, I. Haddadin (2009), "Radiation exposure from radioactive iodine 131I in and surrounding the patients' room", *Radiation Protection Dosimetry*, **135(1)**, pp.64-68, DOI: 10.1093/rpd/ncp093.

[5] Ministry of Health, Ministry of Science and Technology (2018a), *Circular No. 13/2018/TT-BKHCN Dated September 5, 2018 of The Minister of Science and Technology Amending and Supplementing a Number of Articles of Joint Circular No. 13/2014/TTLT-BKHCN-BYT Dated June 9, 2014 of The Minister of Science and Technology and The Minister of Health Regulating Radiation Safety in Medicine* (in Vietnamese).

[6] The International Commission on Radiological Protection (2007), *The 2007 Recommendations of The International Commission on Radiological Protection*, Elsevier, 35pp.

[7] World Health Organization (2016a), *Global Strategy on Human Resources for Health: Workforce 2030*, 64pp.

[8] World Health Organization (2016b), *Human Resources for Health Country Profiles: Viet Nam*, 54pp.

[9] Ministry of Health (2018b), *Health Statistics Yearbook*, 277pp (in Vietnamese).

[10] J.W. Brodack, M.R. Kilbourn, M.J. Welch, et al. (1986), "Application of robotics to radiopharmaceutical preparation: Controlled synthesis of fluorine-18-estradiol", *Journal of Nuclear Medicine*, **27(5)**, pp.714-721.

[11] COMECER (2023), *THEODORICO 2 Robotic Dispenser for Radiopharmaceuticals*, https://www.comecer.com/theodorico-2-robotic-dispenser-for-radiopharmaceuticals/, accessed 15 May 2023.

[12] G. Guevara, J. Lima, P. Leitão, et al. (2015), "Development of a robotic prototype system for the preparation and partition of radioactive products", *2015 IEEE International Conference on Industrial Technology (ICIT)*, pp.173-178, DOI: 10.1109/ICIT.2015.7125095.

[13] T. Alsennawy, W. Alonazy, I.E. Saad (2017), "Robotic radioactive doses delivery system in nuclear medicine facility", *IOSR Journal of Applied Physics*, **9(6)**, pp.34-40, DOI: 10.9790/4861-0906013440.

[14] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, et al. (2021), "A survey of robots in healthcare", *Technologies*, **9(1)**, DOI: 10.3390/technologies9010008.

[15] Aldebaran Robotics (2023), *Pepper*, https://www.aldebaran.com/en/pepper, accessed 15 May 2023.

[16] Diligent Robotics Inc. (2022), *Moxi*, https://www.diligentrobots.com/moxi, accessed 15 May 2023.

[17] J. Wilkinson (2022), "The strong robot with the gentle touch", *Riken*, https://www.riken.jp/en/news_pubs/research_news/pr/2015/20150223_2/, accessed 15 May 2023.

[18] R.C. Browning, E.A. Baker, J.A. Herron, et al. (2005), "Effects of obesity and sex on the energetic cost and preferred speed of walking", *Journal of Applied Physiology*, **100(2)**, pp.390-398, DOI: 10.1152/japplphysiol.00767.2005.

[19] P. Kazanzides, P. Thienphrapa (2008), "Centralized processing and distributed I/O for robot control", *2008 IEEE International Conference on Technologies for Practical Robot Applications*, pp.84-88, DOI: 10.1109/TEPRA.2008.4686678.

[20] MDN Web Docs (2024), *Web Audio API*, https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API, accessed 5 January 2024.

[21] Massachusetts Institute of Technology (2024), *PyAudio Documentation*, https://people.csail.mit.edu/hubert/pyaudio/docs/, accessed 4 January 2024.

[22] Open Source Computer Vision (2024), *Introduction to OpenCV-Python Tutorials*, https://docs.opencv.org/4.x/d0/de3/tutorial_py_intro.html, accessed 2 January 2024.

[23] R. Smith (2007), "An overview of the Tesseract OCR engine", *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, **2**, pp.629-633, DOI: 10.1109/ICDAR.2007.4376991.

[24] Shreeshrii (2024), *Finetuned Traineddata Files for Tesseract 4.0.0 for Testing*, https://github.com/Shreeshrii/tessdata_shreetest?tab=readme-ov-file#readme, accessed 3 January 2024.

[25] Tesseract Documentation (2024), *Improving The Quality of The Output*, https://tesseract-ocr.github.io/tessdoc/ImproveQuality.html, accessed 3 January 2024.

[26] Chartjs.org (2024), Chart.js Getting Started, https://www.chartjs.org/docs/latest/getting-started/, accessed 3 January 2024.

[27] M.R.D. Águila, A.G. Ramírez (2014), "Sample size calculation", *Allergologia et Immunopathologia*, **42(5)**, pp.485-492, DOI: 10.1016/j.aller.2013.03.008.

[28] F. Gul, W. Rahiman, S.S.N. Alhady (2019), "A comprehensive study for robot navigation techniques", *Cogent Engineering*, **6(1)**, DOI: 10.1080/23311916.2019.1632046.

[29] H.K. Kondaveeti, S. Raman, P. Raj (2020), "Prototyping with Raspberry Pi in healthcare domain", *International Journal of Advanced Computer Science and Applications*, **11(10)**, DOI: 10.14569/IJACSA.2020.0111038.

[30] S. Naik, E. Sudarshan (2019), "Smart healthcare monitoring system using Raspberry Pi on IoT platform", *ARPN Journal of Engineering and Applied Sciences*, **14(4)**, pp.872-876.

[31] A. Saha, M.M.H. Bhuiyan, D. Saha, et al. (2023), "Medbot-design and development of medical robot for healthcare digitalization", *AIUB Journal of Science and Engineering*, **22(1)**, pp.1-8, DOI: 10.53799/ajse.v22i1.257.

[32] M.A. Hossain, M.E. Hossain, M.A. Rahaman (2021), "Multipurpose medical assistant robot (Docto-Bot) based on internet of things", *International Journal of Electrical and Computer Engineering*, **11(6)**, DOI: 10.11591/ijece.v11i6.pp5558-5567.

[33] G. Haddeler, A. Aybakan, M.C. Akay, et al. (2020), "Evaluation of 3D LiDAR sensor setup for heterogeneous robot team", *Journal of Intelligent & Robotic Systems*, **100(2)**, pp.689-709, DOI: 10.1007/s10846-020-01207-y.

[34] G. Csaba, L. Somlyai, Z. Vámossy (2018), "Mobil robot navigation using 2D LIDAR", *2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics*, DOI: 10.1109/SAMI.2018.8324002.

[35] H. Beomsoo, A.A. Ravankar, T. Emaru (2021), "Mobile robot navigation based on deep reinforcement learning with 2D-lidar sensor using stochastic approach", *2021 IEEE International Conference on Intelligence and Safety for Robotics*, pp.417-422, DOI: 10.1109/ISR50024.2021.9419565.

[36] A. Miller, R. Machrafi, A. Mohany (2015), "Development of a semi-autonomous directional and spectroscopic radiation detection mobile platform", *Radiation Measurements*, **72**, pp.53-59, DOI: 10.1016/j.radmeas.2014.11.009.