

Portable Autonomous Mobile Robot In Multi Speciality Hospital

Rithvik. R, Prashanthini. K, Kishore Kumar. M, Paul Dhayanithi

UG Student B.E. Robotics and Automation,
Assistant.Professor(Sr.Gr) B.E. Robotics and Automation, Sri Ramakrishna Engineering College.

Abstract: - The abstract is to be in fully-justified, below the author information. Use the word “PORTABLE AUTONOMOUS MOBILE ROBOT IN MULTI SPECIALITY HOSPITAL” as the title, in 14-point Times New Roman, boldface type, centered relative to the column, all capitalized. Leave two blank lines after the abstract, and then begin the main text. All manuscripts must be in English.

Hospitals are busy places where staff needs to get supplies, medicines, and samples from one place to another quickly and safely. Instead of having people spend time running these errands, we built a simple robot that can do the job for them—either on its own or with a little help. Our main aim was to make a small, reliable robot that doesn’t cost too much and can be trusted to deliver lightweight medical items. The robot is easy to control, moves around on wheels, and uses sensors to avoid bumping into things. It also has a safe compartment to hold the items it carries. We wanted to show that using a robot like this can make hospital work easier. The robot follows set routes inside the hospital, so deliveries get done faster and with fewer mistakes. This way, nurses and other staff have more time to care for patients instead of handling deliveries.

KEYWORD:MEDICAL SUPPLY, POC PROJECT, OPENCV WITH ROS IN PYHTON TERMINAL, ARDUNIO NANO, REAL VNC IN MOBLE OPERATING CONTROL, AMR.

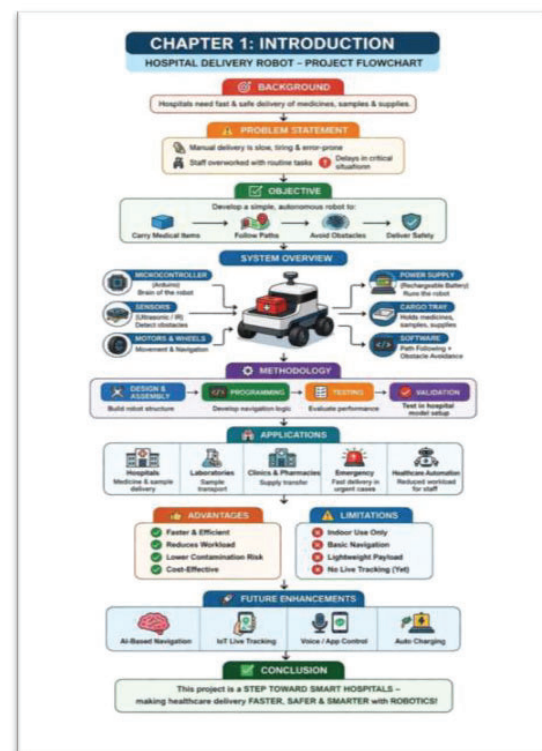
1. INTRODUCTION:

Robots and automation are changing the way many industries work. Hospitals, in particular, need to move things like medicines, test samples, and supplies from place to place, often quickly. Usually, hospital staff does these jobs by hand, but that can be slow, tiring, and sometimes causes delays when time is important. This project is about making a simple, working model of a robot that can carry and deliver small medical items in a hospital. The main goal is to build a small, efficient robot that can do some of these delivery jobs on its own, so doctors and nurses can spend more time caring for patients instead of running errands.

2. SYSTEM DESIGN AND IMPLEMENTATION:

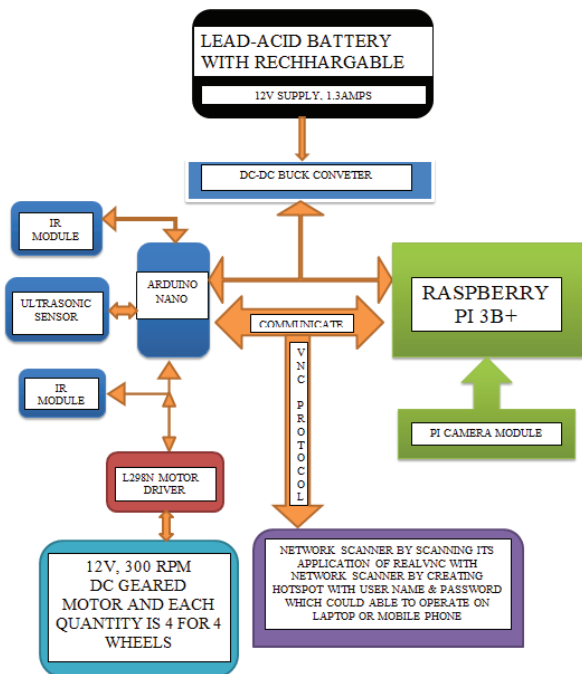
This chapter explains how the delivery robot was designed and built. It covers the steps taken to develop the robot, from choosing the right parts to putting everything together and

The robot uses basic ideas from robotics, like small computers, sensors to see obstacles, and wheels to move around. It follows set paths, like hospital hallways, and avoids bumping into things on the way. The robot has parts like a microcontroller, motor, and sensors, plus a safe spot to carry items. This first version is meant to show that these kinds of robots can really work in hospitals. Even though the model is simple, it already makes delivery easier and cuts down on mistakes and delays as shown in image 1.1. Even though healthcare technology has come a long way, most hospitals still depend on people to move things around inside. This leads to several problems: hospital staff will become overworked with these routine tasks, important deliveries sometimes get delayed, mistakes are more likely to happen, and highly trained workers spend valuable time on chores instead of patient care.



making sure it works as planned. To make sure the robot would actually be useful in a hospital, we started by listing the most important requirements. The robot needed to be small and light, able to move on its own, avoid obstacles, and carry medical items safely. We also made sure it would be easy to control and reliable for everyday use. We selected

hardware that was reliable and easy to find. The main parts included a microcontroller (such as an Arduino), DC motors and motor drivers for movement, ultrasonic and infrared sensors for obstacle detection, a sturdy frame, and a secure tray for carrying items. The robot also uses a rechargeable battery as its main power source. The process of how the robot operates can be shown as a flow chart.



Flowchart of image 2.2

3. SYSTEM ANALYSIS AND DETAILED DESIGN:

This chapter dives into the deeper details of how the hospital delivery robot system is analyzed and designed. It covers the step-by-step approach taken to make sure the robot would meet all the demands of a busy hospital setting, from understanding the requirements to selecting the best hardware and software solutions. Before starting any design, it's important to pin down exactly what the robot needs to do. For this project, that involved gathering feedback from hospital staff and observing typical delivery tasks. The key requirements included safe and reliable movement, the ability to carry various medical items, easy operation for staff, and strong safety features to prevent accidents in crowded corridors. The robot must be able to transport medicines and samples, follow pre-set routes, avoid obstacles, and stop at specific delivery points. It should also be easy to load and unload, recharge its battery with minimal hassle, and alert staff if there's a problem or when a delivery is complete.

3.1 HULL DESIGN SPECIFICATIONS

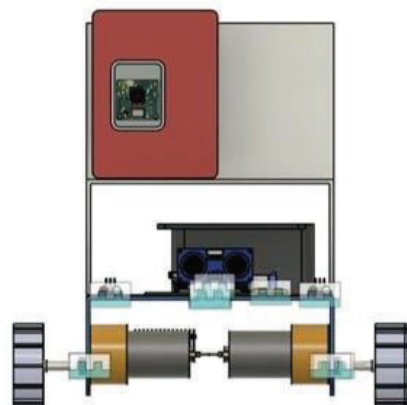
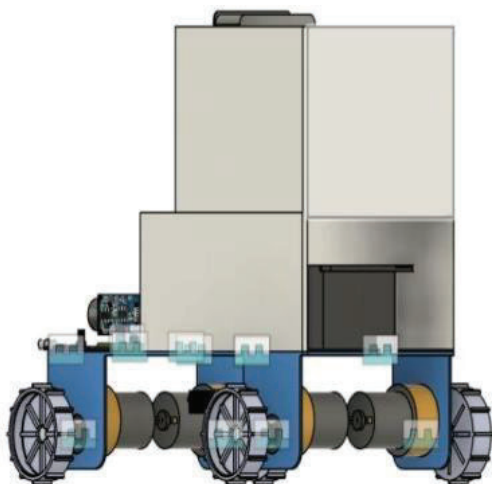
Designing the body—or hull—of the hospital delivery robot is a critical step in making sure the robot can do its job reliably and safely in a real-world healthcare environment. The hull isn't just the robot's outer shell; it's the foundation that holds and protects all the internal parts, from microcontrollers and batteries to motors and sensors. In a hospital, where cleanliness, safety, and durability are top priorities, the hull needs to balance strength and practicality with a clean, approachable appearance.

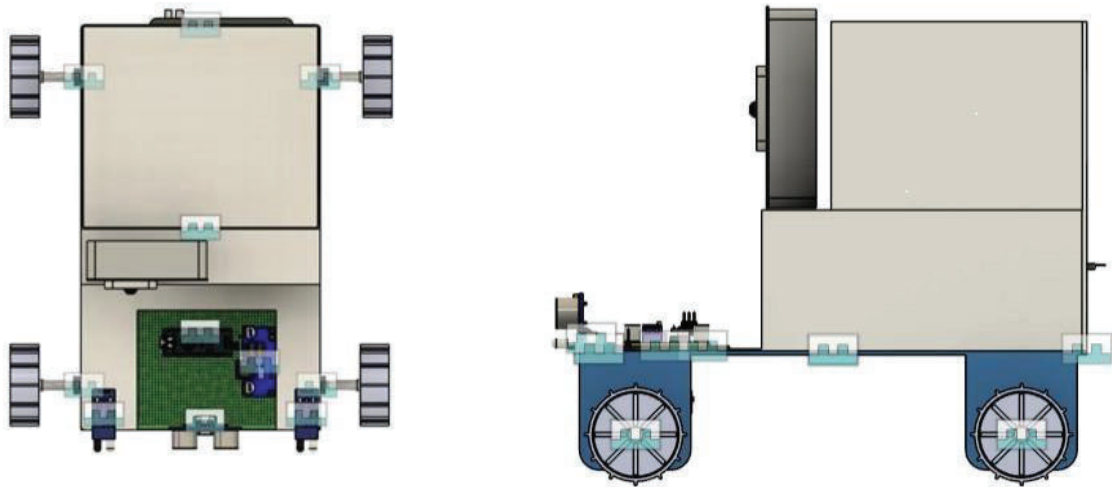
Hull Design Specifications Table: Parameter	Specification	Description / Justification
Material	Foam Board / Aluminium	Lightweight, corrosion-resistant, easy to fabricate
Dimensions (L × W × H)	400 mm × 300 mm × 150 mm	Compact for indoor navigation, fits through hospital corridors
Chassis Type	Rectangular flat base with side supports	Provides stable base, evenly distributes weight
Wheel Configuration	4-wheel drive with 65 mm diameter wheels	Ensures stability and smooth maneuverability on indoor surfaces
Load Capacity	2–3 kg	Sufficient for carrying small medical equipment
Mounting Provisions	Slots and holes for Arduino Nano, Raspberry Pi, L298N, sensors, and battery	Allows secure installation of electronic components
Battery Placement	Centrally located	Maintains low center of gravity, improves stability
Sensor Placement	Front-mounted IR and ultrasonic sensors	Provides accurate line-following and obstacle detection
Modular Tray Support	Detachable platform	Allows easy loading/unloading of medical items
Weight	Approximately 2–2.5 kg (excluding payload)	Lightweight for efficient motor performance
Structural Reinforcement	Cross-braces at base	Reduces chassis bending and improves rigidity
Surface Finish	Smooth edges, rounded corners	Reduces snagging or injury risk in hospital environments

Component	Specification	Function / Description
Arduino Nano	Microcontroller: ATmega328P, 14 digital I/O pins, 8 analog inputs, 5V operation, 16 MHz	Processes sensor inputs, controls motor driver, executes control algorithms
Raspberry Pi 3B+	Quad-core 1.4 GHz CPU, 1 GB RAM, Wi-Fi, Bluetooth, 4 USB ports, GPIO pins	Handles advanced computation, remote monitoring, communication, and VNC interface
L298N Motor Driver	Dual H-bridge, max 2A per channel, 5–35V operation	Controls DC motor speed and direction based on Arduino commands

12V 1.3Ah Lead-Acid Battery	Rechargeable, 12V output, provides power for motors and electronics	Main power source for the robot
DC-DC Buck Converter	Input 12V, output 5V/3.3V, efficiency >85%	Converts battery voltage for low-voltage electronics like Arduino and sensors
IR Sensor Module	Detection range: 2–30 cm, analog/digital output, operating voltage: 3.3–5V	Line detection for path following, detects black/white contrast
Ultrasonic Sensor (HC- SR04)	Range: 2–400 cm, accuracy ±3 mm, operating voltage: 5V	Measures distance to obstacles, enables collision avoidance
Wiring and Connectors	AWG 22–24 wires, insulated connectors	Ensures stable electrical connections and safety
PWM Control Signals	0–5V digital PWM from Arduino Nano	Controls motor speed through L298N driver
Serial Communication	UART, 9600–115200 bps, Arduino ↔ Raspberry Pi	Enables data exchange between microcontroller and processor.

View	Description	Illustration Placement
Left side view	Shows robot from left side, able to view chassis layout, wheel placement and modular tray. Ideal for understanding overall footprint and component alignment.	fig. 1
Right side view	3D view of the right side showing depth, relative placement of components, and spatial relationships	fig. 2
Top view	Displays the robot from the top, highlighting sensor orientation, wheel alignment, and chassis width	fig. 3
front view	front side profile showing wheel positions, camera , carrying tray, and electronics placement	fig. 4





These are the given image of isometric view of 3D CAD

4. COMPARATIVE ANALYSIS, USER IMPACT AND FUTURE PROSPECTS:

It provides a comprehensive evaluation of the medical equipment transport robot by comparing its performance and impact against traditional manual delivery methods commonly used in hospitals. This chapter also explores user feedback, operational challenges, cost-effectiveness, and the future potential of robotic logistics in healthcare. One of the most important measures of the robot's value is how it stacks up against existing manual delivery practices. Traditionally, hospital staff—including nurses, attendants, or porters—are responsible for transporting medicines, lab samples, and equipment across departments. While this system is straightforward, it can be slow, inconsistent, and sometimes error-prone. The introduction of the robot brought about significant improvements. Looking ahead, the potential for further innovation is significant.

Advanced AI and Navigation: The integration of artificial intelligence will enable the robot to learn from its environment, optimize routes, and handle unexpected situations with greater autonomy.

Broader Roles: Beyond deliveries, future robots could assist with inventory management, environmental monitoring, or even basic patient interactions.

Sustainability: Using recyclable materials and optimizing power systems will make future robots more environmentally friendly.

Improved Communication:

Features such as multilingual interfaces, real-time tracking, and integration with mobile devices will further enhance usability and user satisfaction.

Chapter 4 has demonstrated that the medical equipment transport robot represents a major advancement over traditional delivery methods in terms of speed, reliability, cost savings, and staff satisfaction. While some challenges remain, the overwhelming benefits and positive user response pave the way for broader adoption and continued.

5. RESULT AND RECOMMENDATIONS WITH USING CODING METHOD:

5.1 CODING

Camera command file creating in raspberry pi 3b+:

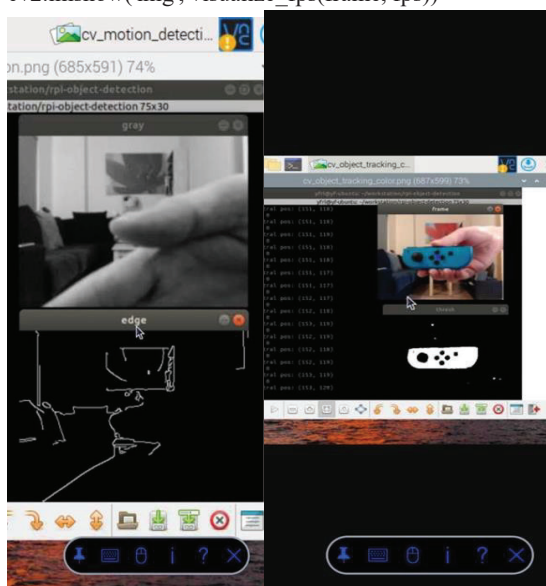
```
import os
import sys
import cv2
import time
import numpy as np
import time
# Add src directory to the path
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(file))))
from utils.picamera_utils
import is_raspberry_camera, get_picamera
CAMERA_DEVICE_ID = 0
IMAGE_WIDTH = 320
IMAGE_HEIGHT = 240
IS_RASPI_CAMERA = is_raspberry_camera()
fps = 0
base_dir = os.path.dirname(os.path.abspath( file ))
print("Using raspi camera: ", IS_RASPI_CAMERA)
def visualize_fps(image, fps: int):
    if len(np.shape(image)) < 3:
        text_color = (255, 255, 255) # white
    else:
        text_color = (0, 255, 0) # green
    row_size = 20 # pixels
    left_margin = 24 # pixels
    font_size = 1
    font_thickness = 1
    # Draw the FPS counter
    fps_text = 'FPS = {:.1f}'.format(fps)
    text_location = (left_margin, row_size)
```

```
cv2.putText(image, fps_text, text_location,
cv2.FONT_HERSHEY_PLAIN, font_size, text_color,
font_thickness)
return image

# Load the cascade
face_cascade = cv2.CascadeClassifier(os.path.join(base_dir,
'haarcascade_frontalface_default.xml'))

# To capture video from webcam. if IS_RASPI_CAMERA:
cap = get_picamera(IMAGE_WIDTH, IMAGE_HEIGHT)
cap.start()
else:
# create video capture
cap = cv2.VideoCapture(CAMERA_DEVICE_ID) # set
resolution to 320x240 to reduce latency cap.set(3,
IMAGE_WIDTH)
cap.set(4, IMAGE_HEIGHT)

# To use a video file as input
# cap =
cv2.VideoCapture('filename.mp4')
while True:
#
.....
# record start time
start_time =
time.time()
# Read the frames from a camera if IS_RASPI_CAMERA:
frame = cap.capture_array() else:
_, frame = cap.read() # Convert to grayscale
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #
Detect the faces
faces = face_cascade.detectMultiScale(gray, 1.1, 4) # Draw
the rectangle around each face
for (x, y, w, h) in faces:
cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2) #
Display
cv2.imshow('img', visualize_fps(frame, fps))
```

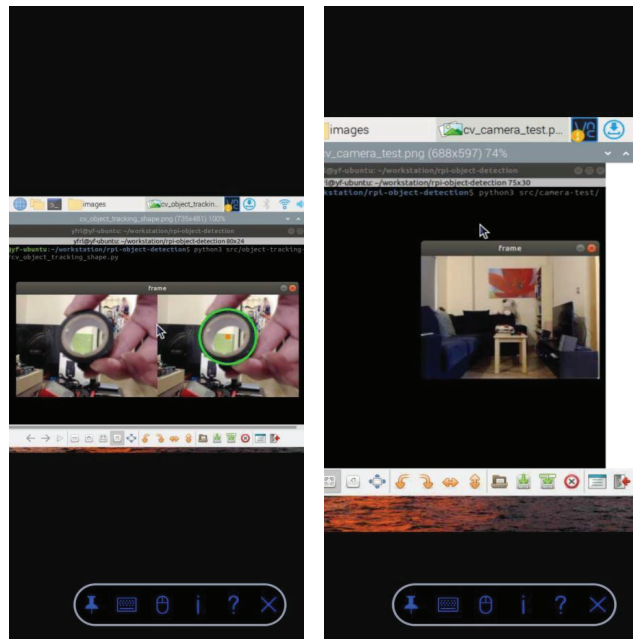


These are the given image of fig 5.1.1 and 5.1.2 cv_command.

5.2 SETUP FUNCTION:

Commands to run Image Processing to detect faces:

```
source ~/.profile
workon cv
cd ~/Desktop/rpi-object-detection-master/src/face-
detection/ python face-detection.py
source ~/.profile
workon cv
cd ~/Desktop/rpi-object-detection-master/src/camera-test
python cv_camera_test.py
```



These are the given image of fig 5.3 and 4.4 setup function command.

5.3 Comparison with Conventional Methods

The experimental robot is compared with traditional manual delivery of medical items.

5.3.1 Discussion

After thoroughly testing the medical equipment transport robot, it is important to reflect on the results, interpret the findings, and put them into context. The experimental study demonstrated that the robot could navigate the simulated hospital environment reliably, deliver items with high accuracy, and interact safely with both static and moving obstacles. However, as with any new technology, the journey from prototype to real-world deployment brings valuable insights, lessons learned, and areas for further improvement.

One of the most significant observations was the robot's adaptability. Despite encountering a variety of obstacles, lighting conditions, and delivery tasks, the robot consistently performed well. The use of ultrasonic and infrared sensors provided reliable obstacle detection, while robust algorithms allowed for quick recalibration and rerouting when paths were blocked.

The controlled acceleration and deceleration routines minimized the risk of damaging sensitive medical items and ensured safe operation in busy corridors.

Another point of discussion concerns the robot's interaction with human users. The ease of loading and unloading cargo, straightforward user interface, and clear signalling were praised by staff during usability trials. This positive user feedback is crucial, as technology adoption in healthcare often depends not only on technical performance but also on how comfortable staff feel working alongside new systems. Nevertheless, some limitations were noted. The robot's line-following abilities, while highly accurate on clearly marked routes, were less reliable on faded or poorly maintained floor markings. Additionally, extremely crowded or cluttered environments sometimes required manual intervention. These findings highlight opportunities for further sensor refinement and software enhancements, such as integrating vision-based navigation or machine learning for better route planning.

5.3.2 Summary

In summary, the experimental validation of the medical equipment transport robot confirmed that autonomous delivery systems can bring real benefits to hospital logistics. The robot offered high accuracy in navigation and delivery, improved staff efficiency by taking over routine tasks, and maintained a strong safety record throughout all tests. User feedback was largely positive, reinforcing the idea that robotics can blend smoothly into the existing workflow when designed with real end-users in mind. The study also identified specific areas for future development, ensuring that the next generation of hospital robots will be even more reliable, flexible, and user-friendly.

5.4 NAVIGATION TIME AND DEFFICIENCY

Feature	Manual Delivery	Robot Performance
Speed	1–2 min per 10 m	30 sec per 10 m
Accuracy	Human error possible	±5 mm deviation consistently
Safety	Risk of dropping items	Collision avoidance prevents damage
Labor Requirement	Requires staff	Fully autonomous
Load Capacity	Limited to one tray	2–2.5 kg per trip

5.4.1 Limitations and Paths Forward:

Parameter	Result
Track Length	20 centimeters
Average Navigation Time	30 seconds
Conventional Manual Time	1–2 minutes
Deviation from Track	±5 mm
Repetition Consistency	High – minimal variation across trials

Despite these successes, the study also identified some areas where further improvements would be beneficial. The robot's line-following mode was slightly less reliable on heavily worn or poorly marked floors, suggesting that future versions could benefit from more advanced visual navigation systems. In rare cases, highly congested areas caused the robot to pause for longer periods, indicating an opportunity to refine obstacle avoidance algorithms or add communication features to alert staff when the robot is temporarily delayed.

Overall, the navigation time and efficiency analysis demonstrates that autonomous robots can bring significant value to hospital logistics. Their consistency, adaptability, and energy efficiency set a new standard for rapid, reliable delivery in complex environments, while freeing staff to concentrate on patient care and other high-priority tasks.

6. CONCLUSION

In closing, the medical equipment transport robot project stands as a testament to the value of thoughtful robotics integration in healthcare. Through careful design, user-centered development, and rigorous testing, the project proved that robots can meaningfully improve efficiency, safety, and staff satisfaction in hospital logistics. By embracing innovation and a culture of continuous improvement, healthcare institutions can unlock new levels of productivity and patient care—ushering in a future where technology and humanity work hand in hand.

7. REFERENCE

The references section provides a comprehensive list of scholarly articles, books, technical manuals, standards, and online resources that have been consulted throughout the design, modelling, fabrication, and experimental validation of the medical equipment transport robot. Proper referencing ensures credibility, acknowledges original authors, and allows readers to explore related work for deeper understanding.

Books and Textbooks

1. **Groover, M. P. (2020).** *Automation, Production Systems, and Computer-Integrated Manufacturing*. 5th Edition, Pearson.
 - o Provides foundational concepts on automation systems, robotic design principles, and integration with manufacturing processes. This book guided the understanding of system architecture and automation control used in the project.
2. **Craig, J. J. (2018).** *Introduction to Robotics: Mechanics and Control*. 4th Edition, Pearson.
 - o A key resource for robotic kinematics, dynamics, sensor integration, and motion control. The text was particularly useful for designing the robot's movement algorithms and understanding sensor-based navigation.
3. **Siciliano, B., & Khatib, O. (2016).** *Springer Handbook of Robotics*. Springer.
 - o Provides in-depth coverage of modern robotic systems, control algorithms, and sensor technologies. Used extensively for guidance on integrating multiple sensors and electronics in autonomous robots.
4. **Hall, D. V., & Hall, D. (2019).** *Mechatronics: Principles and Applications*. 2nd Edition, Cengage Learning.
 - o Focused on integration of mechanical systems, electronics, and control, which was critical for

developing the modular robotic chassis and electrical subsystem.

Technical Manuals and Datasheets

1. **Arduino Nano Datasheet.** Arduino, 2021.
 - o Provided pin configurations, electrical specifications, and programming guidelines essential for sensor interfacing and motor control.
2. **Raspberry Pi 3B+ User Guide. Raspberry Pi Foundation, 2018.** Provided hardware specifications, GPIO details, and guidance for communication and control software integration.
3. **L298N Motor Driver Datasheet.** STMicroelectronics, 2020. Used for designing the motor control circuitry and implementing PWM-based speed regulation.
4. **12V 1.3Ah Rechargeable Lead Acid Battery Datasheet.** Exide Industries, 2021. Specifications used for power supply calculations, load testing, and battery life estimation.
5. **HC-SR04 Ultrasonic Sensor Datasheet.** 2020. Provided detailed specifications for detection range, accuracy, and interfacing.
6. **IR Sensor Module Datasheet.** 2020. Guidance on line-following sensor calibration and integration with Arduino Nano.

SOFTWARE AND TOOLS REFERENCES

- Autodesk Fusion 360.** Autodesk Inc., 2021. Used for CAD modeling, assembly design, simulation, and generating four-view renders. Documentation was referenced for tutorials on parametric modeling and joint constraints.
- Real-VNC Viewer.** Real-VNC Ltd., 2020. Used for remote monitoring and controlling Raspberry Pi during experimental trials. Reference manuals helped in setting up secure and stable communication.
- Arduino IDE.** Arduino.cc, 2021. Programming environment used to develop microcontroller code, sensor interfacing routines, and motor control algorithms.
- Python 3.9. Python Software Foundation, 2021.** Used on Raspberry Pi for serial communication, data logging, and sensor processing.

Terminal Emulator Tools. PuTTY, 2020. For debugging, monitoring sensor outputs, and establishing communication between Raspberry Pi and Arduino Nano.

Standards and Guidelines

- 1. **ISO 13482:2014.** Robots and robotic devices – Safety requirements for personal care robots. o Provided safety standards for designing and testing autonomous robots in environments with humans.
-

- 2. **IEEE 1872-2015.** Standard Ontologies for Robotics and Automation. o Guidelines for modelling robotic systems and defining sensor and actuator interfaces.

Online Resources

1. **Arduino Official Documentation** – <https://www.arduino.cc/>
2. **Raspberry Pi Official Documentation**– <https://www.raspberrypi.org/documentation/>
3. **Fusion 360 Tutorials and Forum Discussions** – <https://knowledge.autodesk.com/>
4. **ResearchGate Articles on Healthcare Robotics** – <https://www.researchgate.net/>