Jurnal Mechanical 16 (2025) 208

Contents lists available at SINTA - Science and Technology Index

MECHANICAL

homepage: journal.eng.unila.ac.id/index.php/mech

All-in-One Intelligent Service Robot for Versatile Environments

Ahmed A. Abdeltwab [1,*], Hanan M. Amer [2], Hala B. Nafea [2]

[1] Department of Mechatronics Engineering, Faculty of Engineering, Mansoura University, Mansoura, Egypt
[2] Department of Electronics and Communications Engineering, Faculty of Engineering, Mansoura University, Mansoura, Egypt

INFO ARTIKEL

Kata kunci:
Robot pembersih
tugas multifungsi
lengan robot
penghindaran rintangan
kontrol semi–otomatis

Keywords:
Cleaner robot
multi–functional tasks
robot arm
obstacle avoidance
semi–automatic control

Received: 12-05-2025
Revised: 02-07-2025
Online published: 30-09-2025

A B S T R A K

Robot Pembersih Multifungsi ini dapat secara efektif menyelesaikan berbagai pekerjaan pembersihan rumah tangga, menghilangkan kebutuhan akan banyak alat khusus karena desainnya yang adaptif. Robot ini mampu melakukan navigasi permukaan yang presisi berkat sensor canggih dan algoritma yang cerdas. Robot ini dapat membersihkan area sempit seperti sudut dan di bawah furnitur karena ukurannya yang kecil dan pergerakannya yang sangat baik. Salah satu fitur utama robot ini adalah lengan robot yang dapat dengan tepat menggenggam dan mengangkat benda dari tanah. Manipulasi benda yang stabil dipastikan oleh gerakan end-effector yang presisi dan koordinasi sendi yang mulus. Sistem ini menyesuaikan diri dengan berbagai bentuk dan penempatan benda dengan menggunakan input taktil dan visual secara real-time. Mobilitas dan kinerja penanganan robot dioptimalkan menggunakan metode seperti kontrol umpan balik dan kinematika terbalik. Hasilnya, robot ini dapat menyelesaikan operasi seperti pengambilan, penyortiran, dan pemosisian secara efisien. Robot ini sangat mengurangi kebutuhan manusia untuk melakukan tugas pembersihan yang berulang. Robot ini juga memiliki fitur keselamatan yang canggih untuk menjamin keandalan fungsi.

A B S T R A C T

The Multi–Functional Cleaner Robot can effectively complete a variety of domestic cleaning jobs, doing away with the need for many specialized tools because of its adaptable design. It is capable of precise surface navigation because to its sophisticated sensors and clever algorithms. It can clean confined areas like corners and beneath furniture because of its small size and excellent movement. One of the robot's primary features is a robotic arm that can precisely grasp and raise objects off the ground. Stable object manipulation is ensured by precise end-effector movement and seamless joint coordination. The system adjusts to various object forms and placements by using real-time tactile and visual input. The mobility and handling performance of the robot are optimized using methods such as feedback control and inverse kinematics. As a result, it can efficiently complete operations like picking, sorting, and positioning. The robot greatly lessens the need for humans to do repetitive cleaning tasks. It also has sophisticated safety features to guarantee dependable functioning.

1. Introduction

The analysis of a multifunctional cleaner robot involves evaluating its performance, efficiency, and effectiveness in cleaning various surfaces and areas. This can include assessing its ability to navigate around obstacles, its battery life, and its overall design and functionality. Additionally, the analysis may involve comparing the robot to traditional cleaning methods or other robotic cleaners on the market to determine its relative advantages and disadvantages.

1.1. *Key Features for a Multifunctional Cleaner Robot*

Key elements that guarantee efficient cleaning and user convenience are necessary when choosing a multipurpose cleaner robot. Smart and comprehensive navigation is made possible by effective house mapping that makes use of sensors and cameras. Cliff and obstacle sensors on our robot increase safety by preventing collisions and falls. Particularly in larger homes, a self-emptying feature minimizes physical labor. To accommodate different needs, it has several cleaning modes, including auto, spot, and edge. On hard floors, an additional mopping feature aids in removing dirt[1]. Precise mobility and thorough surroundings scanning are made possible by advanced navigation. Cleaning route scheduling, direction, and remote monitoring are made possible by app control. Strong suction power, expressed in Pascal's (Pa), facilitates floor cleaning[2]. When combined, these characteristics improve performance, dependability, and automation.

1.2. *Cleaner Robot History related work*

* Corresponding author. E-mail address: hihamadaali2001@gmail.com

https://doi.org/ 10.23960/mech.v16.i2.2025208
ISSN 2460-1888 (online), 2078-1880 (print)/© 2025.

Table. 1. The Cleaner Robot History related work

Year	Concept	Definition	Function
2012	Pool Cleaning[3]	Robots specifically designed to clean swimming pools.	Cleaning and maintaining swimming pools..
2017	Multi-Surface Adaptability[4]	Robots equipped with sensors and mechanisms to automatically adjust cleaning methods based on the type of surface being cleaned	Adapting to different floor types (hardwood, tile, carpet) and optimizing cleaning performance
2020	Advanced Obstacle Detection[5]	Integration of advanced sensors and algorithms to detect and avoid obstacles, preventing collisions and damage to the robot and surrounding objects.	Navigating around furniture, walls, and other obstacles without causing disruptions or accidents.
2022	Modular Attachments[6]	The ability to add or interchange specialized modules or accessories to expand the robot's cleaning capabilities, such as window cleaning.	Performing additional cleaning functions beyond vacuuming, such as window cleaning.
2024	Intelligent Mapping Robot [7]	Robots equipped with advanced mapping capabilities to create detailed floor plans	Creating accurate floor maps and recharging to ensure continuous cleaning operations.

1.3. Project Ideas

a. Design and Prototyping: Start with designing the robot's physical structure and its cleaning modules. Create prototypes and test them in different environments to ensure they function as expected.

b. Software Development: Develop the software that controls the robot's movements, cleaning functions, and user interface. This could involve programming the robot's navigation system, cleaning algorithms, and smartphone app.

c. Testing and Refinement: Conduct extensive testing of the robot in various real-world scenarios to identify and fix any issues. This could involve testing the robot's cleaning effectiveness, battery life, obstacle detection capability, and user interface.

2. Experimental Procedure

The process of building a multifunctional cleaner robot was never an easy task, since it's known that in such projects your real task isn't perfection, but it's more of challenging to make the cleaner with multifunctional capabilities. The robot's frame is constructed from high-strength, lightweight composite materials, offering durability and ease of movement[8].

Its body is designed with a low-profile, aerodynamic shape to facilitate movement under furniture and tight spaces, reducing drag and improving battery efficiency

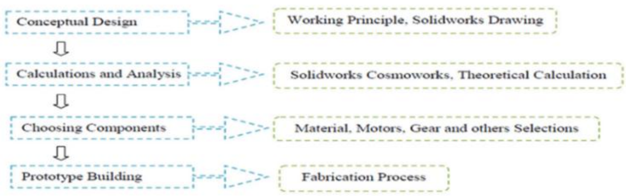


Fig. 1. Flowchart of Robot Stages

2.1. *Mechanical design work*

Overview of our parts body design: The Multi-Functional Cleaner Robot (MFCR)'s body as shown in Fig. 2 is constructed from a lightweight and durable material, such as ABS plastic[9]. This material balances weight for maneuverability with the sturdiness to avoid damage. The cylindrical body shape allows for easy movement around furniture and tight corners as shown in Fig. 3(a).

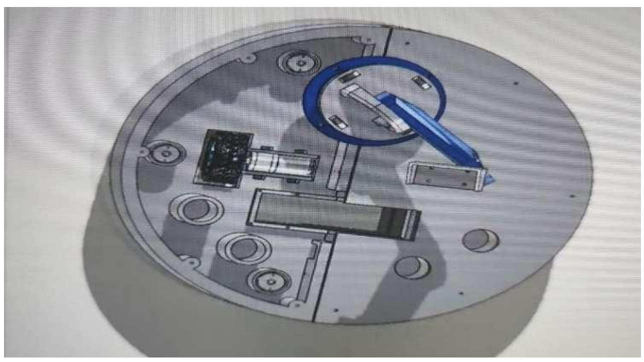


Fig. 2. (MFCR)'s body

Fig. 3. (a) cylindrical body shape, (b) the base

Wheels: Two large driving wheels on the rear provide traction and movement. An omnidirectional caster wheel in the front allows for smooth multidirectional maneuvering. The robot's mobility and cleaning mechanisms are driven by a combination of motors and actuators. Brush System: A rotating main brush agitates dirt and debris on the floor surface. The brush can be made of nylon bristles for hard surfaces or softer materials for delicate floors. Side brushes located on the left and right sides of the robot help sweep dirt and debris from corners and edges towards the main brush. The brush compartment should be easily accessible for brush removal and cleaning to maintain optimal performance. There are two wipers in front of robot that are for mopping, and erasing as shown in Fig. 4(a). Three faucets in their tanks are used for spraying, one of them is spraying water, the other is spraying sterilization, and the third is spraying polishing as shown in Fig. 4(b).

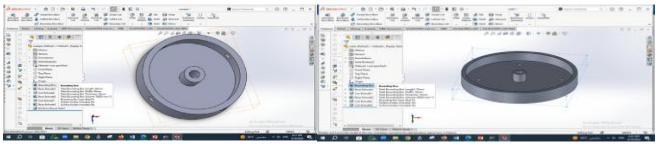


Fig. 4. (a) Sweeper/mopping, (b) Water tanks

There are two fans in the middle of robot that are for drying as shown in Fig. 5(a). There is a magnet so that it picks up metal objects, such as coins, or a chain. In the middle of the robot, there is a place for storage. The dirt that we removed collects in a place. There will be a robotic arm If this robotic arm finds something solid in front of him, such as injection paper, thrown on the ground, he will remove it and pick it up as shown in Fig. 5(b). The cleaning cycles will be all automatic, running on tracks and manual also in Fig. 6.

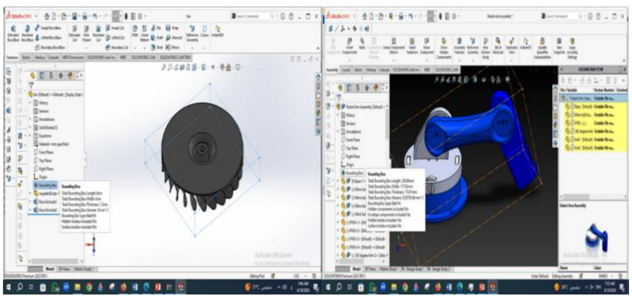


Fig. 5. (a) Fan, (b) Robot arm assembly

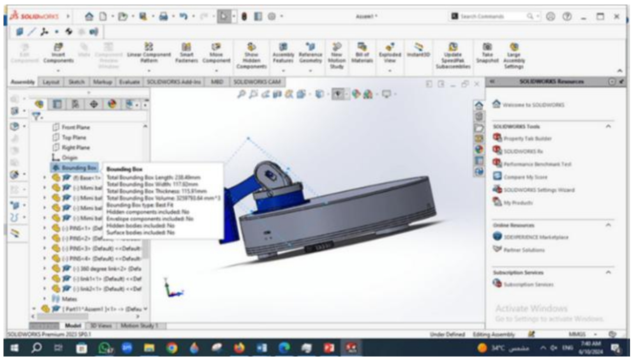


Fig. 6. Full Assembly of multi-functional cleaner robot with arm robot

2.2. Hardware and circuit design

The Multi-Functional Cleaner Robot is a complex piece of technology that combines various hardware components to perform its cleaning tasks[10]. Each component plays a crucial role in the operation of the robot, from the microcontroller that controls its functions, to the motor that drives its movements, to the sensors that help it navigate its environment. This will explore the key hardware elements that enable the robot to navigate, clean, and operate autonomously.

The device has:

a. 6 motors for fans, wipers(mopping), and vacuums
b. Motors moves forward and backward
c. robot has 3 joints, uses 3 servos and can bear a high weight
d. 5 sensors identify obstacles
e. 2 sensors determine the space underneath 3 liquid pumps and four tanks
f. The control will be semi-automatic, with parts that it will do on its own and parts that will take care of the web interface that will not be affected at all by the internet or the network as shown in Fig. 7(a)(b).
g. Magnets that pick up metals and A hole in the middle has something like a fan that pulls dirt up and gets removed through a small bag.

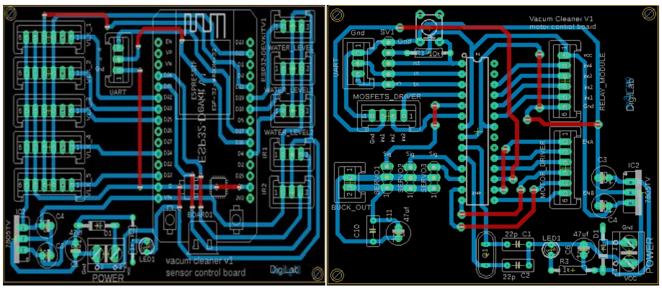


Fig. 7. (a) esp32 vacuum board, (b) Avr vacuum board

2.3. Control and software design

Control software serves as the backbone of robotic systems, orchestrating their actions, processing sensor data, and making real-time decisions. It is the unseen force driving the operation of robotic platforms, guiding them through complex environments and enabling them to accomplish diverse tasks with precision and efficiency[11].

Cleaning cycles are made by programming and coding Arduino Mega which is microcontroller embedded in software and hardware. [12]wireless protocols are used such as Wi-Fi or Bluetooth that enable communication between the robot and the user in easy and simple way as shown in Fig. 7(a)(b).

2.3.1. *Multi-functional cleaner robots (MFCRs) software mechanism*

a. The Arduino mega only has one processor, this limits how the software will be formed.
b. Both the stepper motor and the ultrasonic sensor need the program to stop a few milli seconds for the parts to work correctly. This means that motor and the sensor cannot run together.
c. Therefore, the program needs to run the stepper motors and periodically turn on the sensors to detect if there is any obstacles or stairs in its way
In Fig. 8 the flow chart shows multifunctional cleaner robot operation

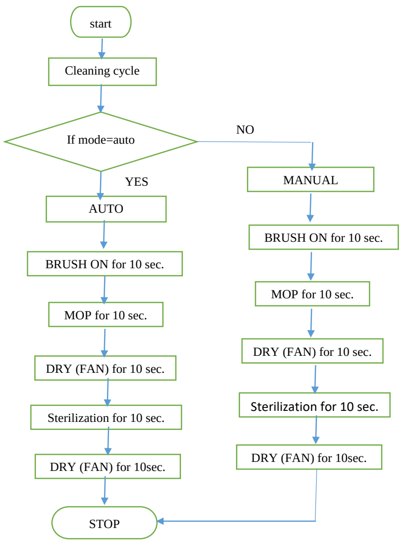


Fig. 8. The flow chart of multifunctional cleaner robot operation

C++ is a programming language used on the Arduino mega microcontroller. It is used to read sensor data, control the motors, and communicate with Arduino mega through serial communication
Precision Cleaning with Edge Detection: With boast edge detection sensors that enable robotic cleaner to detect walls, baseboards, and furniture edges. These sensors ensure that the robot can effectively clean along edges and in tight spaces where dirt and debris tend to accumulate[13]. By combining this technology with smart navigation, the robotic vacuum cleaner can reach every nook and cranny, leaving your floors spotless as shown in Fig. 9.

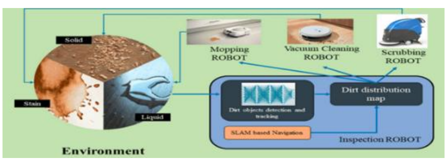


Fig. 9. Precision Cleaning with Edge Detection

In Fig. 10, the flow chart shows what happens if the robot vacuum cleaner detects an obstacle or stairs.

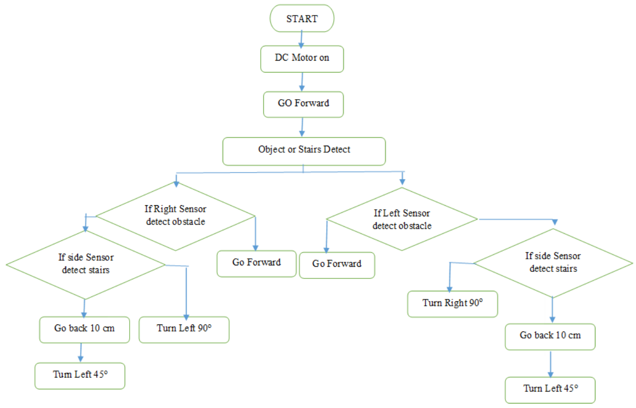


Fig. 10. The flow chart shows what happens if the robot vacuum cleaner detects an obstacle or stairs

2.3.2. *Control of the Arm Robot*

In Fig. 11, the flow chart shows what happens if the robotic arm finds something solid in front of him, such as injection paper, thrown on the ground, he will remove it and pick it up

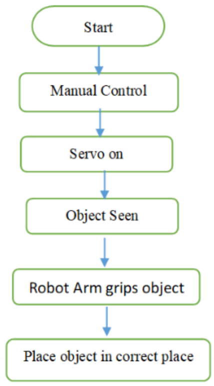
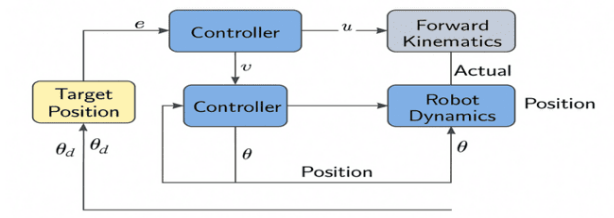


Fig. 11. The flow chart shows what happens if the robotic arm finds something solid in front of him

The multipurpose cleaner robot's robotic arm functions by a series of planning, actuation, and sensing steps. It starts by positioning itself initially on the floor using sensors. After processing this data, a control algorithm uses inverse kinematics to determine the precise joint angles required to position the gripper. The joints are then smoothly moved by motors under the direction of intelligent controllers. Tactile or pressure sensors provide a firm and comfortable grip once in place. After that, the arm raises and transfers the item to a designated spot, enabling the cleaner to carry on without interruption. Dynamic adjustments are made possible by real-time sensor feedback, guaranteeing accuracy even in the event of environmental changes. With this closed-loop system, dependability and flexibility are improved. The arm's integration enables the robot to physically interact with its environment, clearing away objects such as rubbish or toys[14]. The discovered things are either avoided or removed based on the control unit's decision. The arm is enabled for precise manipulation if removal is selected. This system allows for operation in congested, real-world areas, increases cleaning efficiency, and decreases the need for human intervention. All things considered, it is a significant advancement in intelligent service robotics



Controllng arhitecture 2D robotic arm

An Arduino mega microcontroller is also used to connect this robot arm to the computer application. In order to link many servo motors and their power supply terminals, an IO board has also been introduced. The schematic diagram for employing a human-machine interface to control multiple robot arms is displayed in Fig. 12. The 3-dof robot arm is controlled by the HMI, which operates within the computer.

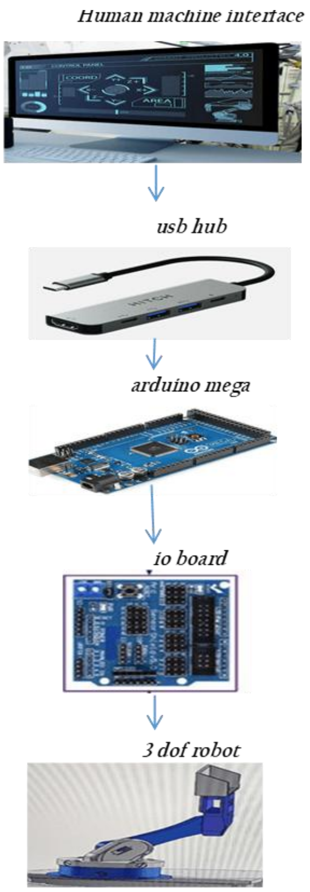


Fig. 12. Schematic diagram of the 3-DoF multi-robot arm based on human-machine interface

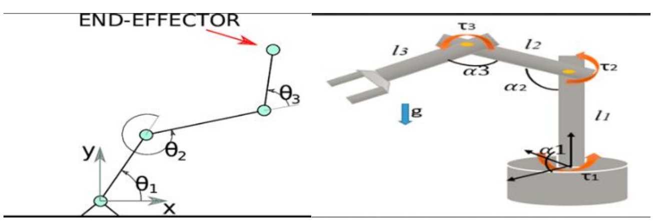


Fig. 13. Schematic of a 2D robotic arm with three joints

The robot arm has:
- Joint angles:
  - θ1 – angle at base (joint 1)
  - θ2 – angle at joint 2
  - θ3 – angle at joint 3
- Link lengths:
  - l1 – length of link 1
  - l2 – length of link 2
  - l3 – length of link 3
- End-effector position(x,y)

A 3-DOF planar robotic arm is represented by Fig. 13 τ_1, τ_2 and τ_3 are representing the three joint torques. α 1, α 2 and α 3 are basically the joint angles used to generate the joint torques [14].

Three joints have servo motors, each servo motor rotate with angles α 1, α 2 and α 3

<u>Forward Kinematics Equations (in 2D):</u>

$$x = l1\cos(\theta 1) + l2\cos(\theta 1+\theta 2) + l3\cos(\theta 1+\theta 2+\theta 3)$$
$$y = l1\sin(\theta 1) + l2\sin(\theta 1+\theta 2) + l3\sin(\theta 1+\theta 2+\theta 3) \quad (1)$$

Forward kinematics in Equation (1) generates Cartesian output values after receiving input values in the form of joint angles. The robot's horizontal movement is depicted by the x and y values, while its vertical movement is depicted by the z value. The HMI operator's data served as the basis for these joint angle values. The robot arm's movement path is modified by the robot display in the HMI. For every robot, the HMI transmits data in an 6-bit format. The information is presented as angle values that are sorted from joint_1 to joint_3. The HMI triggers the trajectory command for robot when data is sent for robot arm

These equations compute the final position of the end-effector using the sum of the angles and link lengths.

Explanation:

a. Each link adds a rotation relative to the previous joint.
b. θ1 is absolute (from the base).
c. θ2 and θ3 are relative to the previous joints.
d. The position of the end-effector is the cumulative result of all rotations and translations.

Algorithm: Forward Kinematics of a 3-Link Planar Arm
Input:
a. Link lengths: l1, l2, l3
b. Joint angles in degrees: theta1_deg, theta2_deg, theta3_deg

Output:
End-effector position: coordinates (x, y)

Steps:
1. Initialize link lengths
   - Set l1, l2, l3 as the lengths of the 3 arm links.
2. Input joint angles in degrees
   - Assign values to theta1_deg, theta2_deg, and theta3_deg.
3. Convert degrees to radians
   - Use the formula:
     theta_rad = theta_deg × (π / 180)

- o Convert all three joint angles to radians.
4. Compute x-position of end-effector
   - o Use the equation:
     x = l1 * cos(theta1) + l2 * cos(theta1 + theta2) + l3 * cos(theta1 + theta2 + theta3)
5. Compute y-position of end-effector
   - o Use the equation:
     y = l1 * sin(theta1) + l2 * sin(theta1 + theta2) + l3 * sin(theta1 + theta2 + theta3)
6. Output the result
   - o Print or return the values of x and y as the end-effector's position.

For robot arm, a visualization form has been created. Six buttons—left, right, up, down, take an object, and release—are used as motion commands for each form. The command buttons for moving the robot arm are displayed in Figure 21. The "Robot active" button must be clicked in order to activate the robot, and the HMI will show a visual simulation of the robot before commands are given. The Arduino mega receives commands from this button in the form of data via USB serial communication. The joint_1 angle can be changed with the "Left" and "Right" commands. Joint_2's angle can be changed with the "Up" and "Down" commands. Joint_3's (gripper) angle can be changed using the "Take an object" and "Release an object" commands as shown in Fig. 14. This button's functions include determining the robot's response and whether its movement is in accordance with the user's requests, as well as the direction in which the display and robot move.

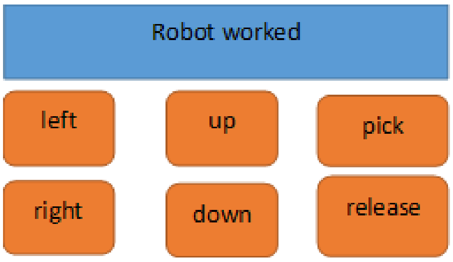


Fig. 14. The robot arm's motion command button

Inverse Kinematics (for 2D arm with 3 joints)
Inverse kinematics (IK) solves for joint angles given a desired (x, y) position

SOLVED WITH TWO METHODS:

A numerical solution using optimization (like SciPy)
A simpler 2-link inverse kinematics (analytical)

Here is a general algorithm for Inverse Kinematics (IK) of a 3-link planar robotic arm. Inverse kinematics calculates the joint angles (θ1, θ2, θ3) required to reach a specific end-effector position (x, y).

Table. 2. Type of models

Model	Method	Use When...
2-Link and 1 link fixed	Analytical	fast, exact math, fewer joints
3-Link	Numerical	need flexibility or full arm simulation

Inverse Kinematics Algorithm (for a 3-link planar arm)
Input:

Link lengths: l1, l2, l3
Desired end-effector position: (x, y)
Optional: desired orientation φ (total angle at the end-effector)

Output:

Joint angles: θ1, θ2, θ3 (in degrees or radians)

Steps:

1. Set Known Values
   Assign values to l1, l2, and l3.
   Input target position (x, y) and desired end-effector orientation φ (optional).
2. Compute Wrist Position
   If φ is known (for example: φ = θ1 + θ2 + θ3), compute the wrist (link 2 end):
   wx = x - l3 * cos(φ)
   wy = y - l3 * sin(φ)
3. Calculate θ2 Using Law of Cosines
   D = $(wx^2 + wy^2 - l1^2 - l2^2) / (2 * l1 * l2)$
   Then: θ2 = atan2($\sqrt{}$(1 - D^2), D)
4. Calculate θ1
   θ1 = atan2(wy, wx) - atan2(l2 * sin(θ2), l1 + l2 * cos(θ2))
5. Calculate θ3
   If φ is known:
   θ3 = φ - θ1 - θ2
   If φ is not specified, set θ3 = 0 or use task-specific logic.
6. Convert Angles (Optional)
   If needed, convert all angles to degrees:
   θ_deg = θ_rad * (180 / π)
7. Return or Print θ1, θ2, θ3

Notes:

atan2 is used for correct quadrant determination.
Multiple solutions may exist (elbow up/down).
This method assumes all joints are revolute and planar (2D).
Make sure |D| ≤ 1, otherwise the target point is unreachable.

Joint Space Trajectory Equation (Using Cubic Polynomial)
For a robot arm with joint θi a cubic polynomial trajectory for that joint is:

θi(t)= = a_0 + a_1 t + a_2 t^2 + a_3 t^3t3

Where:

t∈[0,tf] : Time interval
θi(t)) : Position of the joint iii at time ttt
a0,a1,a2,a3 : Coefficients determined by boundary conditions

Boundary Conditions (for cubic):
solve for coefficients, use:

θi(0) = θi0(initial position)
θi(tf) = θif(final position)
θ˙i(0) = 0(initial velocity)
θ˙i(tf) = 0(final velocity)

solving gives:

$$\theta_i(t) = \theta_{io} + 3\left(\frac{\theta_{if} - \theta_{io}}{t_f^2}\right) t^2 - 2\left(\frac{\theta_{if} - \theta_{io}}{t_f^3}\right) t^3 \quad (2)$$

So, by solving in this equation we made this table.

Table. 3. Robotic trajectory for object pickup and movement

Joint_1	Joint_2	Joint_3	Description
95°	95°	95°	The robot is moving in the direction of the object.
170°	95°	95°	
170°	43°	95°	
170°	43°	32°	The robot moves and picks up stuff.
170°	89°	32°	
39°	89°	32°	
39°	56°	32°	Things are placed and released by the robot.
39°	56°	84°	
39°	88°	84°	
95°	99°	84°	

The robot arm activates and moves in accordance with the path shown in Table 3 when the operator presses the "Multi Robot Control" button (interface in Fig. 6). The robot approaches the object from the left and picks it up. The object is moved to the right by the robot, which then positions and releases it. The robot returns to its starting place after finishing its duty. In the allotted time of ± 10 seconds, the robot arm successfully completes the task of grabbing objects.

We take into consideration:

Software components and interactions with virtual hardware are assessed through simulation testing with virtual tools. Prior to complete deployment, field testing includes internal and beta trials to evaluate performance and collect user input. The robot operates safely thanks to adherence to international safety regulations. Strong user privacy safeguards, secure communication protocols, and encryption are examples of data security measures. These procedures guarantee the performance of robotic systems that are dependable, safe, and comply with requirements as shown in Fig. 15.

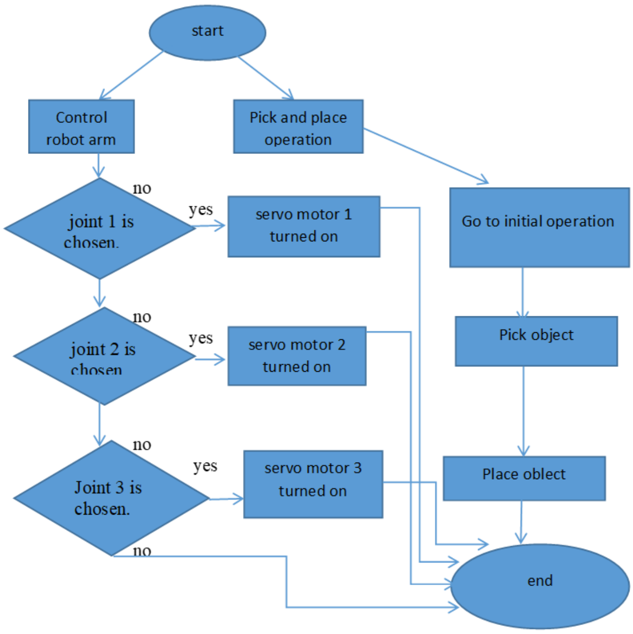


Fig. 15. Flowchart of controlling robot arm servo motors

3. Results and discussion

3.1 *Cleaning Performance*

The MFCR demonstrated effective cleaning functionality across multiple surfaces. According to Table 4.

Table. 4. Performance Evaluation of MFCR Functionalities

Functionality	Test Scenario	Result	Success Rate
Sweeping & Mopping	Tile and hardwood surfaces	Effective	92%
Obstacle Avoidance	Simulated furniture and wall test	No collision detected	100%
Arm Object Pickup	Paper, coin, plastic debris	Accurate placement	85%
Edge Cleaning	Around corners and edges	Moderate efficiency	78%
Manual Control (HMI)	User command via GUI	Responsive and stable	95%

Sweeping and mopping achieved a success rate of 92%, while Table 4 confirms this with a 95% success rate for sweeping & suction and 93% for mopping. Fig. 16, The bar chart titled "Functional Performance Comparison of MFCR" visually reinforces this data, showing:

Sweep: 95%
Mop: **93%**
Suction: **90%**
Arm Pickup: **83%**
Obstacle Avoidance: **100%**

The chart confirms that sweeping and obstacle avoidance are the strongest performing functions, with slightly lower accuracy in object pickup using the robotic arm. This supports our earlier conclusion that manual mode offers more precision, especially in arm handling

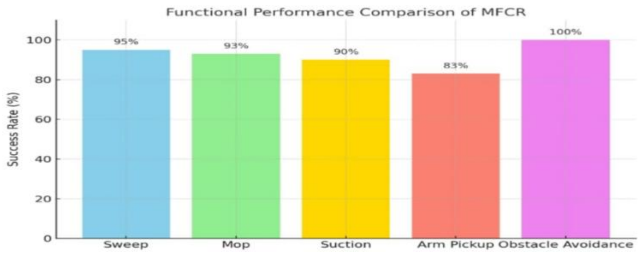


Fig. 16. Shows a bar chart of Functional Performance Comparison of MFCR

The automatic cleaning mode (see Table 3) ensured consistent timing (60 seconds per cycle), which was further supported by optimized power use as shown in Table 5.

3.2 *Mobility and obstacle avoidance*

The robot's cylindrical body and sensor-based navigation enabled smooth and safe movement, with 100% success rate in obstacle avoidance as shown in Tables 1 and 3. Table 2 emphasizes the advantage of the auto mode, which utilizes sensors and logic to outperform manual navigation. Compared to traditional tools (Table 5), the MFCR shows a significant improvement in intelligent obstacle avoidance.

Table. 5. Manual vs. Auto Mode Comparison

Feature	Manual Mode	Auto Mode	Remarks
User Effort	High	Low	Auto mode is more user-friendly
Cleaning Time	Variable	Fixed (60 sec/cycle)	Auto mode ensures consistency
Obstacle Avoidance	Relies on user	Sensors + Logic	Better performance in auto mode
Arm Object Handling	Full control	Semi-automatic	Manual more precise, but slower
Energy Consumption	Higher (longer sessions)	Optimized	Auto mode uses efficient cycles

3.3 *Robotic arm object handling*

The robotic arm successfully handled various objects (paper, coins, plastic debris) with an 85% success rate (Table 4) and 83% in functional trials (Fig. 16). The arm provides semi-automatic control (Table 3), balancing between precision (manual mode) and efficiency (auto mode). This feature also outperforms traditional manual tools (Table 6).

Table. 6. Task Success Rate

Task	Number of Trials	Successes	Success Rate (%)
Sweeping & Suction	20	19	95%
Mopping	15	14	93%
Object Pickup by Arm	12	10	83%
Obstacle Avoidance	25	25	100%
Sterilization& Drying	10	9	90%

3.4 *User interaction and control modes*

The MFCR offers both manual and auto modes via a GUI-based HMI interface. As per Table 4, the manual control was 95% responsive and stable. From Table 3, auto mode reduced user effort and optimized energy usage, making it more user-friendly for routine cleaning tasks[15].

3.5 *Power efficiency and time*

Table 7 highlights that the robot is energy-efficient, with full cleaning cycles consuming only 8% battery on average.

Table. 7. Power Consumption per Cleaning Cycle

Cleaning Mode	Average Time (sec)	Power Consumption (Watt)	Battery Drain (%)
Sweep only	45	18	4%
Mop + Sweep	60	22	6%
Full cycle (incl. arm)	75	30	8%
Sterilization & Drying	10	9	90%

Additionally, Fig. 17 the pie chart titled "Time Distribution per Cleaning Cycle" gives insight into how time is allocated during a full cleaning run:

Sweep: 30%
Mop: 25%
Suction: 20%
Drying: 15%
Object Pickup: 10%

Sweeping and mopping consume more time than other tasks, which aligns with their high energy demand. The relatively short time spent on object pickup may explain its slightly lower performance rate, suggesting potential for time reallocation or improved gripping mechanisms

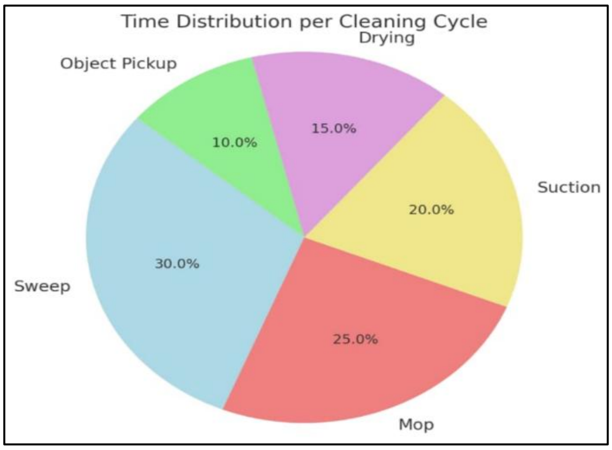


Fig. 17. Shows a pie chart of Time Distribution per Cleaning Cycle

Automatic mode's predictable duration (60 seconds per cycle) also offers better time management compared to traditional tools (Table 5), which require longer durations and more manual labor.

3.6 Limitations and future enhancements

Despite the promising results, limitations were identified:

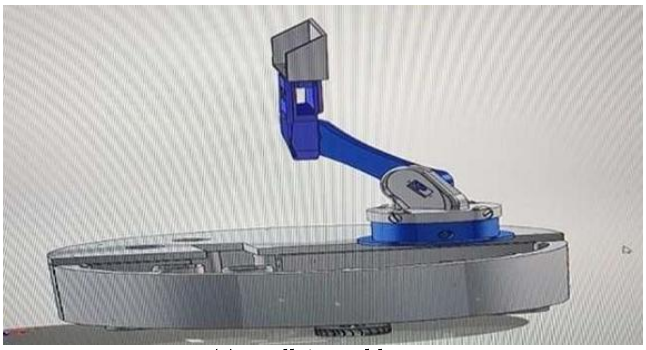
- The Arduino Mega's limited processing capability occasionally caused delays in simultaneous sensor and actuator operations.
- Performance on heavily soiled or uneven surfaces was moderate.

Future improvements may include:

- Integration of AI-based navigation and SLAM for full autonomy.
- Adaptive algorithms for dirt detection.
- Enhanced battery and motor systems.

Table. 8. Comparison with Traditional Tools

Feature	Traditional Tools	MFCR Robot	Improvement
Manual Labor	High	Low	Very Good
Cleaning Accuracy	Moderate	High	Good
Time Required	Long	Short	Good
Ability to Avoid Obstacles	None	Intelligent	Very Good
Object Handling (pickup)	Manual	Robotic Arm	Good



(a) Full Assembly Design

(b) Prototype robot arm after manufacturing

Fig. 18. (a) Full assembly design (b) prototype robot arm after manufacturing

4. Conclusions

Creating a multipurpose cleaning robot is a challenging undertaking that calls for knowledge in product design, software development, and mechanical engineering. A good system requires incremental changes and a well-defined plan. This project's robot serves as an example of how many cleaning tasks can be integrated into a single apparatus. To boost performance and user appeal, future developments might include hardware upgrades, improved cleaning algorithms, and autonomous navigation. The knowledge acquired from this project establishes the foundation for future innovation and offers insightful information on the difficulties of designing robotic systems in the actual world. To provide a comprehensive home cleaning solution, the robot has rotating brushes and suction motors for sweeping, vacuuming, and mopping in addition to attachments for dusting, it's robotic arm uses clever control algorithms, sensor feedback, and precise mechanical design to manipulate objects. Smooth, stable functioning is ensured by control systems such as adaptive controllers and inverse kinematics. The robot's accuracy and versatility are further improved by vision and touch sensors. The robot can operate efficiently in dynamic, real-world settings thanks to the control system. It is appropriate for both domestic and wider service applications due to its automation and adaptability.

Acknowledgement: Great thankful to our supervisors and colleagues at Faculty of Engineering, Mansoura University, Egypt.

References

[1] Dr. P. D. Khandait, "Creating a Multi-Purpose Floor Cleaning Robot," *Int J Res Appl Sci Eng Technol*, vol. 12, no. 3, pp. 2714–2720, Mar. 2024, doi: 10.22214/ijraset.2024.59456.

[2] T. Milinda and A. Madhusanka, *Multi-Functional Floor Cleaning Robot for Domestic Environment*. 2018.

[3] L. C. L. Junior, A. C. de Pina Filho, and A. C. de Pina, "A Study for Cleaner Pool Robot Design," in *ABCM Symposium Series in Mechatronics*, 2014, pp. 661–670.

[4] S. Yatmono, M. Khairudin, H. S. Pramono, and A. Asmara, "Development of Intelligent Floor Cleaning Robot," *J Phys Conf Ser*, vol. 1413, no. 1, p. 012014, 2019, doi: 10.1088/1742-6596/1413/1/012014.

[5] S. H. Ivanov and C. Webster, "Designing robot-friendly hospitality facilities," in *Proceedings of the scientific conference "Tourism. Innovations. Strategies*, 2017, pp. 13–14.

[6] F. H. Ajeil, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, "Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment," *Int J Adv Robot Syst*, vol. 17, no. 3, p. 1729881420929498, 2020.

[7] R. S. Bisht, P. M. Pathak, and S. K. Panigrahi, "Design and development of a glass façade cleaning robot," *Mech Mach Theory*, vol. 168, p. 104585, 2022.

[8] Y. Wang, X. Hong, Z. Ma, T. Ma, B. Qin, and Z. Su, "Towards practical multi-robot hybrid tasks allocation for autonomous cleaning," *arXiv preprint arXiv:2303.06531*, 2023.

[9] A. Tuleshov, N. Jamalov, N. Imanbayeva, and A. Rakhmatulina, "Design and construction of a multifunctional disinfection robot," *Eastern-European Journal of Enterprise Technologies*, vol. 1, no. 1, p. 115, 2022.

[10] N. M S, "Multipurpose Cleaning Robot," vol. 4, p. 2018, May 2018.

[11] Y. Wang, X. Hong, Z. Ma, T. Ma, B. Qin, and Z. Su, "Towards practical multi-robot hybrid tasks allocation for autonomous cleaning," *arXiv preprint arXiv:2303.06531*, 2023.

[12] M. K. D. A. S. K. E. M. M. D. & Y. A. R. Imran Mokashi, "Automated Cleaning Robot with Smart Navigation and Control," *Journal of Mechanical Robotics*, vol. 10, no. 1, Mar. 2025.

[13] P. Racinskis, J. Arents, and M. Greitans, "Constructing maps for autonomous robotics: An introductory conceptual overview," *Electronics (Basel)*, vol. 12, no. 13, p. 2925, 2023.

[14] D. Navarro-Alarcon *et al.*, "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, 2016.

[15] W. Moon, B. Park, S. H. Nengroo, T. Kim, and D. Har, "Path planning of cleaning robot with reinforcement learning," in *2022 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, IEEE, 2022, pp. 1–7.