IFERP Explore Your Research Journey...

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 12, Issue 01, January 2025

RoboFetch – An Automated Item Retrieval System

[1] Abdul Rasheed Roshan, [2] Qurram Ali Ibad, [3] Mohammed Ibrahim Nasir, [4] Syed Mubeen Ali

[1] Electronics and Communication Engineering Dept, Muffakham Jah College of Engineering and Technology, Telangana
[2] Mechanical Engineering Dept, Muffakham Jah College of Engineering and Technology, Telangana
[3] Computer Science Engineering Dept, Muffakham Jah College of Engineering and Technology, Telangana
[4] Electronics and Communication Engineering Dept, Muffakham Jah College of Engineering and Technology, Telangana
Corresponding Author Email: [1] abdulroshan60@gmail.com, [2] qurram10@gmail.com,
[3] mibrahimnasir.engineer@gmail.com, [4] syedmubeenali553@gmail.com

***Abstract*— The RoboFetch project presents an innovative approach to automated item retrieval using a web-based command system. The system integrates an ESP32 microcontroller with a CoreXY mechanism to achieve precise motion control for fetching items from designated racks. Stepper motors are employed to drive the CoreXY mechanism, enabling efficient and accurate two-dimensional movement. Users can send commands via a website interface, which communicates wirelessly with the ESP32, translating user inputs into mechanical actions. This project aims to reduce human effort in repetitive tasks such as retrieving objects from storage, with applications in warehouses, libraries, and home automation systems. The RoboFetch system emphasizes ease of use, scalability, and reliability, showcasing the potential for robotic systems in automating everyday tasks.**

Index Terms*— Automated Item Retrieval, CoreXY mechanism, Inventory Automation, Robotic Systems, and Real Time Control.

I. INTRODUCTION

As automation continues to reshape industries, the need for efficient and autonomous systems capable of performing repetitive tasks is growing. One such area is item retrieval, commonly found in settings like warehouses, libraries, and home automation systems. Traditional methods of manually fetching objects from storage are time-consuming and labour-intensive, prompting the exploration of robotic solutions to enhance efficiency and accuracy.

The RoboFetch project addresses this challenge by introducing an automated system for item retrieval using a web-based command interface. At the core of this system is the ESP32 microcontroller, which, due to its built-in Wi-Fi and robust processing capabilities, allows for seamless communication with a web interface. The ESP32 is coupled with a CoreXY mechanism, an efficient and accurate motion control system used to achieve two-dimensional movement. Stepper motors drive the CoreXY platform, providing the precision required to retrieve objects from specific locations on a rack.

The RoboFetch system leverages a web interface, enabling users to remotely send commands that are translated into mechanical actions by the ESP32. This not only reduces human effort but also increases the speed and reliability of item retrieval, particularly in environments where repetitive tasks are a key component. The use of WebSocket communication ensures real-time interaction between the web interface and the microcontroller, allowing for immediate response and inventory updates.

This paper explores the design, implementation, and potential applications of RoboFetch, highlighting its scalability, ease of use, and flexibility. By automating the retrieval of items, the system provides a foundation for smart, autonomous storage solutions, transforming the way objects are accessed and managed in various sectors.

II. LITERATURE SURVEY

The field of automated item retrieval and storage management has grown considerably, driven by the need to enhance the efficiency and reduce human intervention in tasks that involve repetitive and precise actions. RoboFetch, an autonomous retrieval system utilizing a CoreXY mechanism, an ESP32 microcontroller, and a real-time WebSocket communication framework, builds on various technologies and concepts previously developed in automated item retrieval systems. This literature survey examines related technologies, their applications, limitations, and the gap that RoboFetch aims to fill.

A. Existing Automated Retrieval Systems

Various technologies have been implemented to automate retrieval tasks, particularly in sectors like warehousing, manufacturing, and retail. The three primary types of systems are robotic arms, conveyor systems and autonomous mobile robots (AMRs).

1. Conveyor Systems

Conveyor systems are widely used in large warehouses and manufacturing environments to move items along a fixed path. While they are efficient for transporting bulk materials, they lack the precision required for selective retrieval of specific items. Additionally, conveyor systems are generally limited to linear movement, which restricts their adaptability for complex storage layouts [11].

2. Robotic Arms

Robotic arms, such as those seen in automated storage systems, offer precise control and flexibility in retrieving specific items from storage. While effective, they are expensive, require significant space, and are complex to design and maintain. Robotic arms are also typically designed for industrial environments, making them impractical for smaller-scale applications due to their cost and complexity [12].

3. Autonomous Mobile Robots

AMRs, which include robots like Amazon's Kiva robots, navigate storage environments by using slam technology independently, retrieving items by moving entire racks or individual items to a pick-up point. AMRs have proven effective in high-throughput settings but have limitations in scalability, cost, and programming complexity. For smaller operations or personal use, AMRs remain cost-prohibitive and often lack the simple adaptability needed for non-industrial applications and small warehouses.

B. Challenges in Current Systems

1.Scalability Issues

Many solutions are designed for large-scale operations and are difficult to scale down for smaller environments like Pharmacies or small warehouses.

2.Complexity

These systems often require complex infrastructure and programming, making them hard to set up and maintain.

3.Lack of Real Time Control

Existing systems do not offer immediate control and feedback, limiting their responsiveness to user commands.

C. Gap Identification

Although automated retrieval technologies have significantly improved efficiency in large-scale industrial settings, several limitations prevent these systems from being widely accessible and adaptable, especially for smaller-scale applications. Key gaps in existing solutions include:

1. Scalability and Adaptability

Most current systems, such as robotic arms and autonomous mobile robots (AMRs), are designed for large-scale environments with substantial infrastructure and space requirements. These systems are designed for high-volume operations but lack the flexibility to be easily scaled down for smaller, constrained environments such as small warehouses, libraries, pharmacies or home settings.

RoboFetch is designed with scalability and modularity in mind. Its use of the compact CoreXY mechanism allows it to adapt easily to varying storage configurations and scales, from personal storage systems to medium-sized warehouses. This flexibility enables RoboFetch to function in environments where traditional systems may be too large or complex.

2. Cost and Complexity

Traditional automated systems, such as AMR and robotic arms, come with high costs for setup, maintenance, and programming, making them inaccessible to small businesses and individual users. Moreover, these systems often require advanced technical knowledge and infrastructure, which may be impractical for non-industrial users.

RoboFetch is cost-effective and user-friendly, built with readily available components like the ESP32 microcontroller and stepper motors, significantly lowering the entry barrier for automated retrieval. By integrating these affordable components, RoboFetch provides a solution that is both affordable and easy to maintain and access, making automation accessible to a broader range of users and applications.

3. Real Time Control and Responsiveness

Many existing systems rely on traditional HTTP communication protocols or batch processing, which lack the immediacy required for responsive, real-time control. For applications where users need instant feedback and control, such as in real-time inventory management or remote retrieval tasks, this delay can hinder efficiency and user experience.

RoboFetch uses WebSocket-based communication and a real time database to establish a persistent, low-latency connection between the web interface and the ESP32 microcontroller. This allows users to send commands and receive immediate feedback, enabling seamless real-time control and monitoring. Such responsiveness is critical for environments where efficiency and user interaction are prioritized.

4. Infrastructure and Complexity in Setup

Large retrieval systems typically require extensive infrastructure, such as complex conveyor networks, rigid shelving configurations, or floor navigation markers, which limit flexibility and increase setup time and costs. Adapting these systems to non-standard environments is challenging and often infeasible.

RoboFetch operates autonomously within a rack-based storage environment, requiring minimal setup. Its compact design allows it to be implemented in a range of storage layouts without the need for complex infrastructure, making it ideal for environments that require flexibility or where setup time and space are limited.

RoboFetch thus bridges the gap by providing an adaptable, affordable, and responsive retrieval solution suitable for various applications beyond industrial settings. This innovation not only makes automation more accessible but also addresses the need for versatile, real-time control in retrieval systems, demonstrating RoboFetch's unique contribution to the field of autonomous item retrieval.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 12, Issue 01, January 2025

III. SYSTEM DESIGN

The RoboFetch system is designed with a combination of hardware and software to enable efficient and precise item retrieval from a rack. The core components include the ESP32 micro controller, the CoreXY mechanism, stepper motors, and a web interface for real-time control.

A. ESP32 Micro Controller

The ESP32 was chosen as the brain of the system due to faster communication and its wireless capabilities. It features:

- **Built in Wifi and Bluetooth**: This provides us seamless wireless communication between the system and the user's device via a web interface.
- **Ease of Programming:** The ESP32 supports development platform like Arduino IDE, making it easier to program and integrate with other components.
- **I\O Flexibility:** It can handle multiple inputs and outputs, allowing control of both stepper motors and communicating with webpage simultaneously.[1]

Fig. 1. ESP32

B. Nema 17 Stepper Motors

Nema 17 Stepper motors are used to drive the CoreXY mechanism. They play a crucial role in ensuring accurate and controlled movement:

- **Precision Control:** Stepper motors move in discrete steps, allowing for precise positioning, which is crucial for retrieving specific items from the rack.
- **High Torque at Low Speeds:** Stepper motors provide the necessary force for precise control of the CoreXY mechanism, even at slow speeds.
- **Easy Integration:** These motors are easily controlled by the ESP32, ensuring smooth communication between the hardware and software.[3]

Fig. 2. Nema17 Stepper Motor

C. Web Interface

The web-based control system allows users to interact with RoboFetch in real-time. This interface operates as follows:

- **Commands:** Users can send commands through a simple web page to control the position of the CoreXY system and instruct RoboFetch to retrieve specific items.
- **Real Time Monitoring:** The web interface communicates with the ESP32 via WebSocket's, enabling real-time feedback and control with minimal delay.
- **Remote Access:** Since the system operates via Wi-Fi, users can control it from any location, making it versatile for various applications such as home automation, warehouses, or libraries.

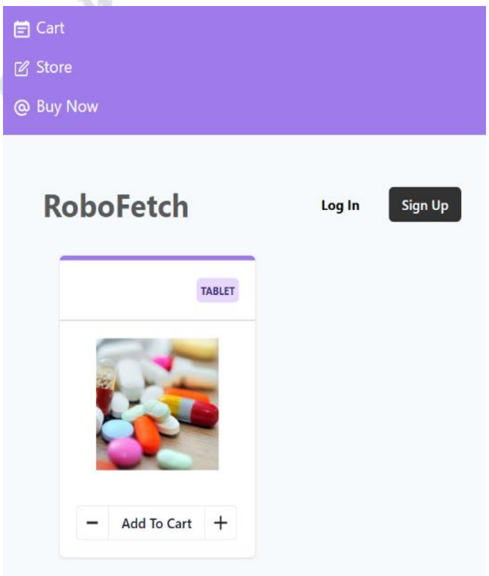

Fig. 3. Webpage

D. CoreXY Mechanism

The CoreXY mechanism is a key feature of RoboFetch that provides efficient and precise two-dimensional movement. It has several advantages:

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 12, Issue 01, January 2025

- **Lightweight and Efficient:** CoreXY reduces the complexity of the mechanical structure, using fewer moving parts compared to traditional Cartesian systems.
- **Compact Design:** Its design is well-suited for compact environments, such as fetching items from a rack, and can be scaled up or down easily.
- **Remote Access:** Since the system operates via Wi-Fi, users can control it from any location, making it versatile for various applications such as home automation, warehouses, or libraries.[2]

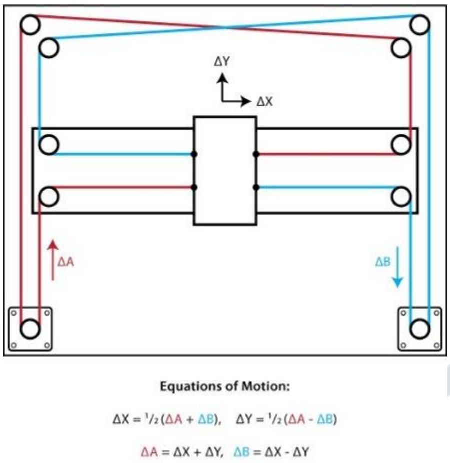

Fig. 4. CoreXY Mechanism

IV. METHODOLOGY

This section details the design and operational workflow of RoboFetch, an automated system for retrieving items from a storage rack. RoboFetch is structured to handle commands from users through a web interface and to operate autonomously using a defined retrieval process.

A. Software Implementation

The software on the ESP32 coordinates command processing, motor control, and communication. Key software functionalities include:

- **Command Parsing:**

The ESP32 interprets instructions received from the web interface, identifying the target coordinates for item retrieval.

- **Motor Control Logic:**

Libraries like accelstepper are used for stepper motor management enable the ESP32 to execute smooth and precise movement on the CoreXY mechanism.

- **Web Socket:**

WebSocket provides a persistent, real-time communication channel, allowing instant transmission of commands and feedback between the ESP32 and the web interface.

B. Workflow of RoboFetch

Based on the flowchart, the RoboFetch workflow proceeds as follows:

- **Item Request Input:**

The user initiates a request to retrieve a specific item via the web interface. This input acts as a trigger for the retrieval process.

- **Inventory Verification:**

Upon receiving the request, RoboFetch checks the database to verify the item's availability. If the item is in stock, the system proceeds with retrieval; if not, it searches for possible alternatives.

- **Alternative Item Search:**

If the requested item is unavailable, RoboFetch scans for alternative options. If an alternative item is available, the user is prompted to confirm retrieval. If the user declines or no alternatives are found, the process terminates here.

- **Coordinate Calculation and Transmission:**

Once an item is confirmed, the system calculates its coordinates within the storage rack. These coordinates are then sent to the ESP32, which translates them into motor instructions for the CoreXY mechanism.

- **Positioning:**

Using the coordinates, the ESP32 directs the stepper motors to move the CoreXY mechanism, positioning the end effector precisely at the target location.

- **Item Retrieval:**

The end effector retrieves the item from the specified location in the rack using a gripping or picking mechanism suited for the item's dimensions and weight.

- **Item Transfer to Collection Bin:**

After retrieval, the system transfers the item to a designated collection bin where the user can access it. This organized placement is particularly useful in applications that require multiple sequential retrievals.

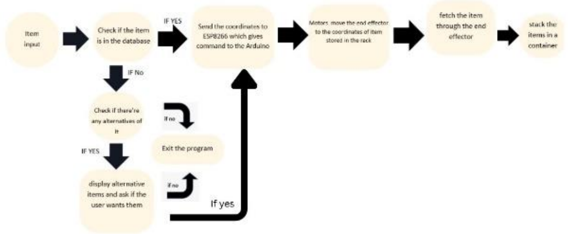

Fig. 5. Flow Chart

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 12, Issue 01, January 2025

C. Control Logic

The control logic on the ESP32 integrates the hardware and software components to manage the retrieval process. From command input to item placement, each stage is executed through programmed logic on the ESP32, ensuring that RoboFetch operates accurately and efficiently.

This methodology showcases the structured approach and system logic that enable RoboFetch to perform automated item retrieval tasks, leveraging real-time control and user-friendly operation to enhance efficiency in storage management.

V. RESULTS AND ANALYSIS

A. Performance Metrics

- **Precision and Accuracy:**

The CoreXY mechanism in RoboFetch demonstrates high precision and accuracy in item retrieval due to the use of stepper motors. During testing, the system achieved a positioning accuracy of approximately 0.1 mm, ensuring that items are fetched from the correct location on the rack. The CoreXY's design minimizes backlash, providing consistent and repeatable movements.

- **Response Time:**

RoboFetch's response time is largely dependent on the efficiency of the WebSocket communication protocol. On average, the system responds to user commands within 5-6 seconds, ensuring instant feedback for real time control. The minimal delay ensures a smooth experience, with rapid system actions following command input. This fast response time makes the system highly suitable for applications requiring quick item retrieval, such as in warehouses or Pharmacies.

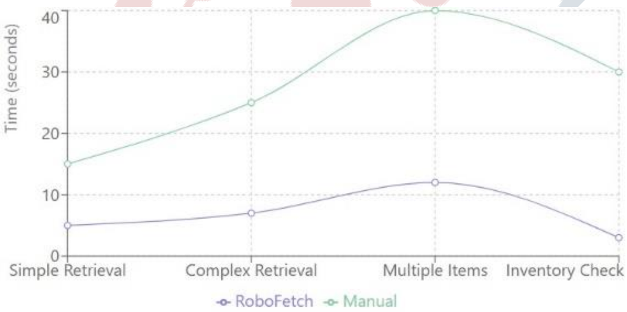

Fig. 6. Time taken to retrieve

- **Scalability:**

The scalability of RoboFetch was evaluated by testing the system on racks of varying sizes and with different types of items. The CoreXY mechanism's flexibility allows for easy adaptation to larger or more complex rack systems without significant changes to the hardware. The system can handle multiple item types and sizes with no loss in performance, but further adjustments may be required for very large racks to ensure motor strength and speed remain consistent. Additionally, RoboFetch can be expanded to control multiple units simultaneously with minimal changes in the control logic.

Fig. 7.

B. Testing

Several test cases and trials were conducted to assess RoboFetch's performance:

- **Time taken to retrieve items:** On an average, RoboFetch takes approximately 8-10 seconds to retrieve an item from a small-to-medium rack. For larger racks, the time may vary due to the additional distance.
- **Error Rates:** During testing, the error rate (mispositioning or failing to retrieve an item) was very low. From Fig.5 we can say that the error rate is very less.

C. Inventory Tracking

RoboFetch integrates real-time inventory tracking using WebSocket communication. This feature allows users to monitor the status of the items on the rack. The system updates the inventory whenever an item is fetched or placed back, providing accurate, up-to-date information. The effectiveness of this feature was demonstrated during testing, where all item retrievals and placements were correctly reflected on the web interface in real-time. This tracking capability is particularly valuable for inventory management in warehouses, ensuring accurate stock levels and minimizing the risk of misplaced items.

VI. APPLICATIONS

A. Use Cases

- **Warehouses:** RoboFetch can be effectively used in warehouses to automate repetitive item retrieval tasks. By reducing the need for human workers to manually fetch items from storage racks, the system reduces labour costs, and minimizes the risk of human error.
- **Pharmacies:** In Pharmacies, RoboFetch can be employed to automate the retrieval of medicines from the racks. By integrating the system with the software, it can locate and retrieve items from designated racks.

IFERP Explore Your Research Journey...

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 12, Issue 01, January 2025

B. Potential Extensions

- **Multi Rack Management:** A potential extension of the system is multi-rack management, where RoboFetch could operate across multiple storage racks simultaneously. This could be achieved by implementing multiple CoreXY systems, each managed by a central controller, allowing for parallel item retrieval. This feature would be highly advantageous in settings with high retrieval demand, such as distribution centres or retail stockrooms.

VII. APPLICATIONS

A. System Advantages

- **Ease of use:** RoboFetch is designed with a user-friendly web interface, allowing users to control the system without any technical expertise. The real-time WebSocket communication ensures immediate responses to commands, making the system responsive for users.
- **Cost-Effectiveness:** Compared to other automated retrieval systems, RoboFetch is built using affordable components such as the ESP32 microcontroller and stepper motors. This low-cost design makes it accessible for smaller operations, unlike more expensive robotic systems used in large warehouses.

B. Limitations

- **Item Size and Weight:** The current design is limited to handling small and lightweight items. Larger or heavier objects may require more robust motors and a reinforced CoreXY mechanism, which could increase the system's complexity and cost.
- **Power Consumption:** Although the system is relatively energy-efficient, continuous operation of the stepper motors may lead to higher power consumption in larger setups or environments requiring constant retrieval.
- **Potential Mechanical Failures:** Like any mechanical system, RoboFetch is susceptible to wear and tear, mainly in the moving parts of the CoreXY mechanism. Regular maintenance and calibration may be required to ensure smooth operation over time, and mechanical failures could disrupt item retrieval.

C. Comparison with existing systems

Compared to traditional conveyor systems or robotic arms, RoboFetch offers a simpler and more affordable solution for smaller environments. Conveyor systems are typically more efficient in handling bulk items but lack the precision and adaptability for specific item retrieval. Robotic arms provide high accuracy but are expensive and complex to implement.

In terms of cost and complexity, RoboFetch is significantly more accessible, particularly for small-to-medium scale applications. Unlike autonomous storage robots, which require complex navigation algorithms and high initial investment, RoboFetch relies on a simpler mechanism that can be easily deployed and maintained. Additionally, it scales better for small operations, whereas other systems are optimized for large warehouses.

D. Future Work

- **Three-Dimensional Retrieval:** Expanding the system to handle three-dimensional movement would allow RoboFetch to retrieve items from multiple levels of a rack, significantly increasing its versatility. This could be achieved by adding a third axis to the CoreXY mechanism or integrating a robotic arm for vertical motion.
- **Additional sensors for object recognition:** Integrating object recognition sensors, such as cameras or bar code readers, would enable RoboFetch to autonomously identify and retrieve specific items without requiring user input. This would be especially useful in environments where items are not stored in fixed locations.

These future developments could make RoboFetch even more powerful, expanding its applications and improving its performance across different industries.

VIII. CONCLUSION

The RoboFetch project introduces an efficient, cost-effective solution for automating item retrieval, significantly reducing the need for manual labor in repetitive tasks. By leveraging a simple yet precise CoreXY mechanism controlled by the ESP32 microcontroller, RoboFetch offers real-time, user-friendly control via a web interface. The project addresses key challenges in manual item retrieval, such as time consumption, human error, and labor intensity, making it an attractive option for small-scale operations like warehouses and Pharmacies.

RoboFetch's flexibility and scalability demonstrate its potential to be deployed across various sectors. From managing inventory in warehouses to Pharmacies, RoboFetch showcases the growing role of robotics in everyday automation. As robotics continues to evolve, systems like RoboFetch will become increasingly integral to improving efficiency, reducing costs, and transforming how we interact with technology in our daily lives.

REFERENCES

[1] Marek Babiuch, Petr Foltynek, Pavel Smutny, "Using the ESP32 Microcontroller for Data Processing" , [2019]

[2] Lucas B. Lange, Michael Knight, CoreXY Mechanism, ResearchGate [2020]

[3] B. Aranjo, P. K. Soori, P. Talukder, "Stepper motor

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 12, Issue 01, January 2025

drives for robotic applications" [2012]

[4] A. Choudhury, "Inventory Management Systems (IMS)", ResearchGate, [2023].

[5] Y. Zhang, J. Wang, "Research of Web Real-Time Communication Based on Web Socket", Research Gate [2015]

[6] Ramamritham, Krithivasan. (1993). Real-time databases. Distributed and Parallel Databases. 1. 199-226. 10.1007/BF01264051.

[7] Stephen, Namulengo. (2022). WAREHOUSING AND INVENTORY MANAGEMENT.

[8] Qigang, Liu, Sun, Xiangyang. (2012). Research of Web Real-Time Communication Based on Web Socket. International Journal of Communications, Network and System Sciences.

[9] Xuan, Cheng, Verbytskyi, Ievgen. (2022). REMOTE CONTROL SYSTEM OF STEPPER MOTOR BASED ON THE ESP32 DEVELOPER BOARD. Bulletin of the National Technical University «KhPI» Series New solutions in modern technologies.

[10] Hooper, Shane. (2020). CoreXY Mechanism.

[11] Mattone, R. and Campagiorni, G. and Galati, F. (2000). Sorting of items on a moving conveyor belt. Part 1: A technique for detecting and classifying objects. Robotics and Computer-Integrated Manufacturing

[12] Kruthika, K and B M, Kiran and Lakshminarayanan, Sanjay. (2016). Design and development of a robotic arm.