Robotic Arm Box Sorter

Raunak Sood, Marcus Tita, Matt Rhee

Department of Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—A system capable of sorting items off of a moving conveyor belt .5 mph**. Automated sorting systems are essential in modern logistics and manufacturing, improving efficiency and reducing human labor. Current package sorting solutions rely on human labor, which is difficult and generally done in poor conditions. This project presents the design and implementation of a robotic arm box sorter that autonomously identifies and sorts packages using QR code detection. The system integrates a robotic manipulator with a vision-based scanning system to read QR codes on incoming boxes, determine their destination, and place them into the appropriate bins. The design includes a mechanical arm, computer vision algorithms for QR code recognition, and motion control for precise handling. Testing demonstrated the system's ability to accurately detect, pick, and sort boxes of varying sizes, achieving a high success rate in real-world scenarios. The results highlight the potential for scalable, cost-effective automation solutions in warehouse and distribution environments.

***Index Terms*—Robotics, Kinematics, Vision, Automation**

I. Introduction

In modern logistics and manufacturing industries, automation plays a crucial role in increasing efficiency, reducing human labor costs, and minimizing errors. One of the key challenges in automated material handling is accurately sorting and placing objects into designated locations. Box sorting is a common task in many warehouses where packages are placed on a conveyor belt and humans have to manually sort the boxes into their correct locations. This is a laborious, monotonous, and slightly dangerous task that many workers around the world have to do, so we would like to automate the process.

Traditional conveyor-based sorting systems, while effective, can be costly, inflexible, and difficult to reconfigure for different types of products. To address these challenges, this project presents the design of an intelligent robotic arm box sorter that utilizes QR code detection for autonomous sorting.

The robotic arm will be equipped with a vision system capable of scanning QR codes on incoming boxes to determine their designated bin. Using this information, the arm will pick up each box and place it in the correct location without human intervention. The system is designed to handle varying box sizes and weights within a predefined range, making it adaptable to different warehouse and packaging environments.

This design document outlines the key components of the robotic sorting system, including the mechanical structure of the robotic arm, the vision-based QR code detection system, motion planning algorithms, and the integration of hardware and software components. The primary goals of this project are to develop a system that is efficient, reliable, and scalable for real-world applications in warehousing, logistics, and automated manufacturing.

II. Use-Case Requirements

The robotic arm box sorter must meet a range of functional and performance requirements to ensure reliable, efficient, and accurate operation in an industrial setting. These requirements span mechanical precision, vision accuracy, system speed, operational efficiency, safety, and scalability. Below are the key use-case requirements for the system:

1. Mechanical and Motion Precision
   - The robotic arm must have a positional accuracy of ±5 mm to ensure precise box placement.
   - The arm must support at least four degrees of freedom (DOF) to allow flexible movement and efficient sorting.
   - The end effector (gripper) must be able to handle boxes of various weights (1–5 lbs) and sizes (minimum: 10×10×10 cm, maximum: 100×100×100 cm) without dropping or damaging them.

2. Vision System and QR Code Detection
   - The system must detect and correctly interpret QR codes with an accuracy rate of at least 90% under varying lighting conditions.
   - The camera resolution must be at least 1080p to ensure clear scanning of QR codes.
   - The vision system must be able to scan QR codes at a distance of 50-100 cm, with a scanning time of less than 0.1 seconds per box.
   - The QR detection software should support industry-standard QR code formats and be capable of integrating with warehouse databases for package tracking.

3. Sorting and Conveyor System Speed
   - The conveyor belt should operate at speeds of 0.5–1 mph, adjustable depending on throughput requirements.
   - The robotic arm must complete a full pick-and-place cycle in less than 3 seconds to maintain high sorting efficiency.
   - The system should support a throughput of at least 10 boxes per minute, ensuring high-speed processing in warehouse environments.
   - The robotic arm and conveyor system must be synchronized, preventing misplacements.

III. Architecture and/or Principle of Operation

The robotic arm box sorter system consists of several key components, including a robotic arm, a conveyor belt, a vision system, and multiple sorting bins. These elements work together to achieve efficient, automated sorting of boxes based on QR code detection. The system architecture is designed to maximize sorting accuracy, speed, and reliability while maintaining a scalable and modular framework.

1. Physical Layout and Component Placement

The system consists of the following physical components:

- Conveyor Belt: We will use a small treadmill as our conveyor belt, which will transport incoming boxes under the vision system. It extends past the robotic arm to an overflow bin for unrecognized or misclassified boxes.
- Robotic Arm: Positioned adjacent to the conveyor belt, the robotic arm is responsible for picking up and sorting boxes based on QR code data. We are building the robotic arm from scratch by 3D printing its parts and using servos and angle sensors to control movement.
- Vision System: A high-resolution camera is mounted overhead, covering the entire conveyor belt width to scan and decode QR codes on the boxes.
- Sorting Bins: Two primary sorting bins are placed beside the robotic arm, each designated for a specific category of boxes. An overflow bin is positioned at the end of the conveyor belt for unrecognized or rejected boxes.

2. System Workflow

The sorting process follows these steps:

- Step 1: Box Arrival on Conveyor Belt
  - Boxes arrive on the conveyor belt in a continuous flow.
  - The speed of the conveyor belt is adjustable, depending on sorting throughput requirements.
- Step 2: QR Code Scanning and Processing
  - The overhead camera captures images of the moving boxes and scans the QR codes.
  - The vision system uses the QR location to make a kinematic plan and sends it to the ESP 32
- Step 3: Robotic Arm Pick-and-Place Operation
  - The arm moves to grasp the box using a suction gripper mechanism.
  - Based on the sorting criteria, the arm places the box into Sorting Bin 1 or Sorting Bin 2.

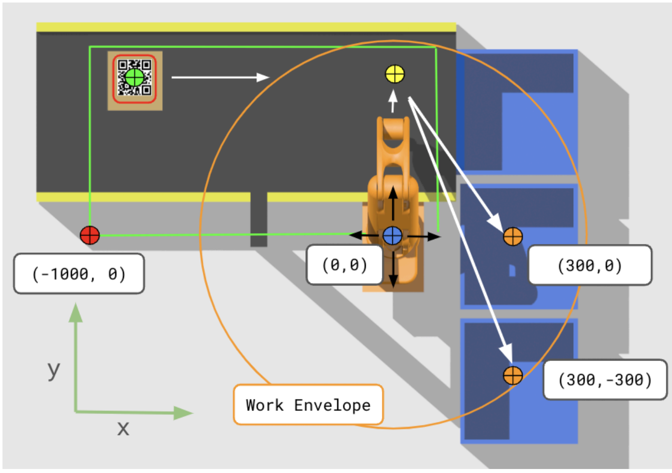

Figure 1: Diagram of our proposed layout. The boxes will move from left to right on the conveyor belt. The camera overlooks the entire system, tracking the boxes and scanning QR codes. The robotic arm then picks up the box at the specified position and places it into one of the two boxes next to it. If the classification fails, the box naturally falls into the overflow bin at the top.

3. Hardware Components

The system consists of the following key hardware components:

- Robotic Arm
  - Servos: 1
  - Angle sensors: 3
  - 3D printed components: 13
  - Motors 3
  - Vacuum pump/ gripper
- Conveyor Belt System
  - Motorized Conveyor: Small treadmill with adjustable speeds of 0.5 to 1 mph.
  - 3 Bins
- Vision System
  - Camera Type: High-resolution industrial camera (1080p or higher), capable of QR code detection.
  - Mounting Position: Fixed above the conveyor belt for top-down QR scanning.

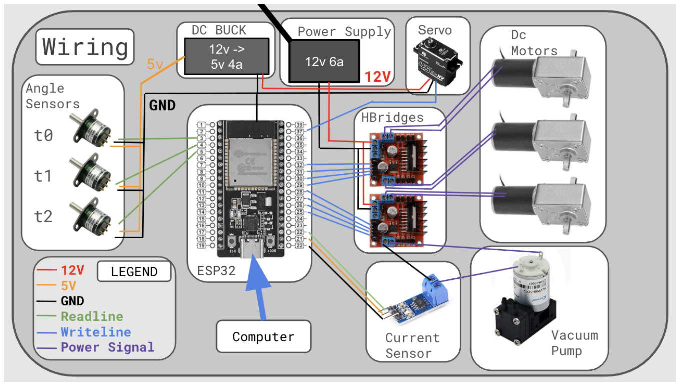

Figure 2: Block diagram

4. Software Requirements

The robotic sorter system is powered by an integrated control and decision-making software stack, which consists of the following key components:

- Vision Processing Module
  - Captures images of boxes on the conveyor belt.
  - Applies image processing techniques to locate and decode QR codes.
  - Sends decoded data to the sorting logic.
- Sorting Decision Engine
  - Receives QR code information from the vision module.
  - Sends commands to the ESP 32 for placement in the appropriate bin.
- Motion Control Module
  - Receives sorting commands from the decision engine.
  - Controls the robotic arm's motion via feedback, ensuring smooth pick-and-place operation.
  - Synchronizes with the conveyor belt system to align box positions.

IV. Design Requirements

Now that we have a robot arm and a conveyor belt, we need to get the two in sync. The conveyor belt is going to be moving at .5mp at the slowest so we are going to rely on a guitar hero type plan. The goal is to have the robot arm hover over where the package is going to be and pick it up right when it passes beneath. The qr location is going to provide all the info we need to do this and the arm is going to be precise and quick.

V. Design Trade Studies

If we say that the robot can sort a number of specific items X in a minute. If we put more than X items on the conveyor belt, the robot won't sort all of them. This means that we need an overflow bin for unsorted packages. If the robot arm sorts at max X packages, we can make an equation using Y the number of packages that we put on the conveyor belt such that Y - X = O where O is overflow. This equation will have some variance and if we ever put a rate higher than X/min for any interval the robot could fail to sort X. Our goal is to have the robot be fast and efficient but not get swamped.

Another consideration would be adding wheels to our bins of sorted items. This would be most useful in a factory setting where multiple bins of packages would be sorted per hour. However in our capstone setting, we won't ever sort a high enough number of packages to meet this requirement. Additionally connecting other conveyor belts or sorting systems might be useful but this is outside of our scope.

A. *Design Specification 1*

Our first design specification is that the arm can move at the appropriate speed to sort 10 items per minute. This means that each join needs to move at approximately 15rpm to account for stoppage time. We will need to test that the main horizontal rotate axis is able to rotate at that speed or at least capable of doing 10 semi circle moves per minute.

B. *Design Specification 2*

Our Second design specification is that the suction gripper has the required suction force to lift an item off of the conveyor belt. The robot arm will move the suction gripper to the top of the box then lift up with the box attached. The current sensor should give a reading of if the suction gripper is engaged.

VI. System Implementation

Our system implementation uses 4 parts. The Computer connects to ESP 32 and Camera Via usb and uses the camera as data source and ESP 32 as movement actuator. The ESP 32 controls the robot arm via feedback, while the conveyor belt is running continuously in the background.

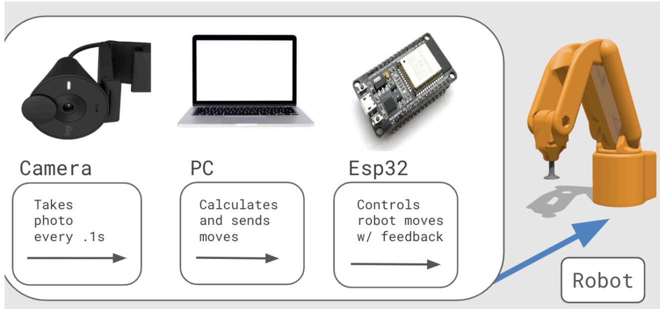

Figure 3: Diagram of our system specification

A. *Subsystem A Macbook*

The PC calculates the inverse kinematics for the Arduino to use to set the robot actuators. The Camera sends its qr data to the computer via Usb and the computer continuously refreshes this using getQR() returning the qr object and location. Then the Computer translates geometry frames to the robot arm and runs inverse kinematics taking into account package speed.

B. *Subsystem B Esp 32*

The Esp 32 gets geometry data from the computer in the form of three theta values. The esp32 sets these values to the robot arm actuator and the robot moves until the new position is achieved. Then the robot has a vacuum pump it can toggle to pick up items by the gripper.

VII. Test, Verification and Validation

We are going to test that the angle of each joint is correct and is updated within an appropriate window for the arduino to stop the motor at a desired position without over shooting. Additionally the robot should

A. *Tests for Design Specification 1*

For each actuator starting with the horizontal actuator we are going to connect it to the motor controller and angle sensor, and then run it on a test bench to see if the motor can achieve the desired speeds. If the motors cannot reach approx 10-15rpm we will need to find a different motor power control system, or lower our speed requirements.

B. *Tests for Use-Case Specification 2*

We are going to connect the vacuum pump to a motor controller through the current sensor and then run the vacuum pump with a tube to the suction gripper. We could use the arm or just hold the suction gripper while lifting the cardboard boxes we plan to use to simulate packages on the conveyor belt. The suction gripper should be able to lift 3-5lbs.

VIII. Project Management

For our project management we plan to keep the same breakdown and schedule as previous.

A. Schedule

We plan to adhere to our previous schedule and continue to work on setting up our hardware and software algos. We have gotten in many hardware components including the camera and we are testing more as we get them in. We are still waiting on robot motors and wiring so we are planning on testing robot capabilities as soon as possible

B. Team Member Responsibilities

Marcus Tita - Robot arm Conveyor Belt setup
Matt Rhee - Qr Location Detection
Raunak Sood - Inverse Kinematics

C. Bill of Materials and Budget

This section is at the back of our Design Report

D. TechSpark Use Plan

We do not plan on using TechSpark for our project.

E. Risk Mitigation Plans

The two main risks we think we might face are difficulty positioning the gripper to pick up packages and having the robot arm pick up a moving package. If we are unable to get an accurate box height reading by using the qr height scaling equation, then we will introduce another sideways camera to estimate the box height using an algorithm. If we have trouble picking up the boxes at the aforementioned speed we will introduce a belt stopping mechanism.

IX. Related Work

Some projects we have seen are a laundry sorting robot. Many robots are capable of simpler taksing using GPT like brains. We have seen some trash sorting robots that rely on suction end effectors.

X. Summary

In summary we are making a scalable autonomous item sorting system. We are using a camera for QR, a robot arm, and a treadmill to act as a conveyor belt for cheap. We are going to plan synchronous actions such that our system can run continuously and have a high item throughput per hour.

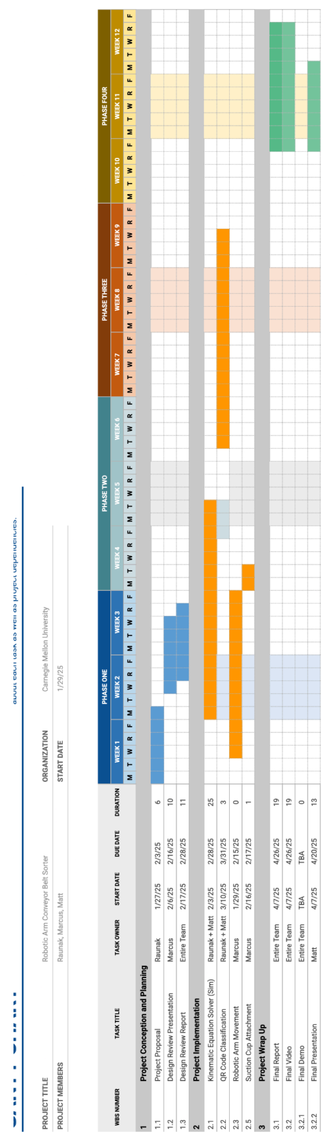

Figure 4: Gantt chart

References

[1] Laundry sorting robot digit web summit AI future," The Guardian, Nov. 15, 2024, [Online]. Available: https://www.theguardian.com/technology/2024/nov/15/laundry-sorting-robot-digit-web-summit-ai-future

TABLE I. BILL OF MATERIALS

What	Model	# if more than 1	Total price
Esp32	ESP32 Dev Kit		10
Camera	Logitech Brio 301		50
Converyer belt	Elseluck Walking Pad		100
Angle Sensor	Yuecoom Hall Angle Sensor,	3	90
Motor	uxcell Worm Gear Motor	3	90
Limit switches	Szliyands V-154-1C25	3-6	5
Wire	WYEUTO 18 awg 4core	10ft	10
Wire	WYEUTO 22 awg 3core	6ft	8
Current Sensor	ATNSINC 30A Range Current Sensor		3
Vac Gripper	NAFUFA Automatic Suction Cup		15
vacuum pump	AYNEFY Mini Vacuum Pump		20
Servo	Muzei		30
		Total	431