ME470/ECE445
SENIOR DESIGN LABORATORY
DESIGN DOCUMENT

A Vision-Integrated Robot for Autonomous Book Classification in Library Environments

Team #34

Zhecheng Lou
Xinrui Xiong
Zhenxiong Tang
Zehao Bao

Advisor: Timothy Lee

April 6, 2026

Contents

Introduction

1.1 Problem Statement

In modern library operations, returned books must be identified, matched to catalog records, and manually reshelved. This process is labor-intensive, time-consuming, and difficult to scale, especially in collections with multiple languages or diverse classification systems. While robotic manipulation for pick-and-place tasks is well studied, the key challenge in this application lies not in executing predefined motions, but in enabling the system to perceive the current state of a bookshelf and make intelligent placement decisions in a dynamic and partially filled environment.

This project aims to develop an autonomous book-return and shelf-organization system in a fixed workstation setting. The system focuses on integrating vision-based perception, task planning, and robotic manipulation to automate the reshelving process. Instead of relying on pre-defined slot positions, the system analyzes real-time shelf occupancy, detects available gaps, evaluates whether a book can fit, and determines a stable and tidy placement strategy.

1.2 Solution Overview & Visual Aid

The proposed system is a fixed workstation comprising a collection bin, a robotic arm with gripper, and bookshelves within reach. The system operates in batch mode, where books are loaded into a vertical bin with their spines facing outward for barcode scanning. After a single user input, the system autonomously completes the following workflow:

- Scan barcodes to identify books and retrieve their designated shelf locations from a catalog database.
- Use real-time shelf occupancy perception to detect gaps, measure widths, and evaluate fit for each book.
- Apply intelligent slot selection policies to choose appropriate placement locations based on gap size, neighboring books, and tidiness criteria.
- Execute placement using visual-servo control, providing real-time feedback during insertion to ensure stability and correctness.

1.3 Demonstration Scenario

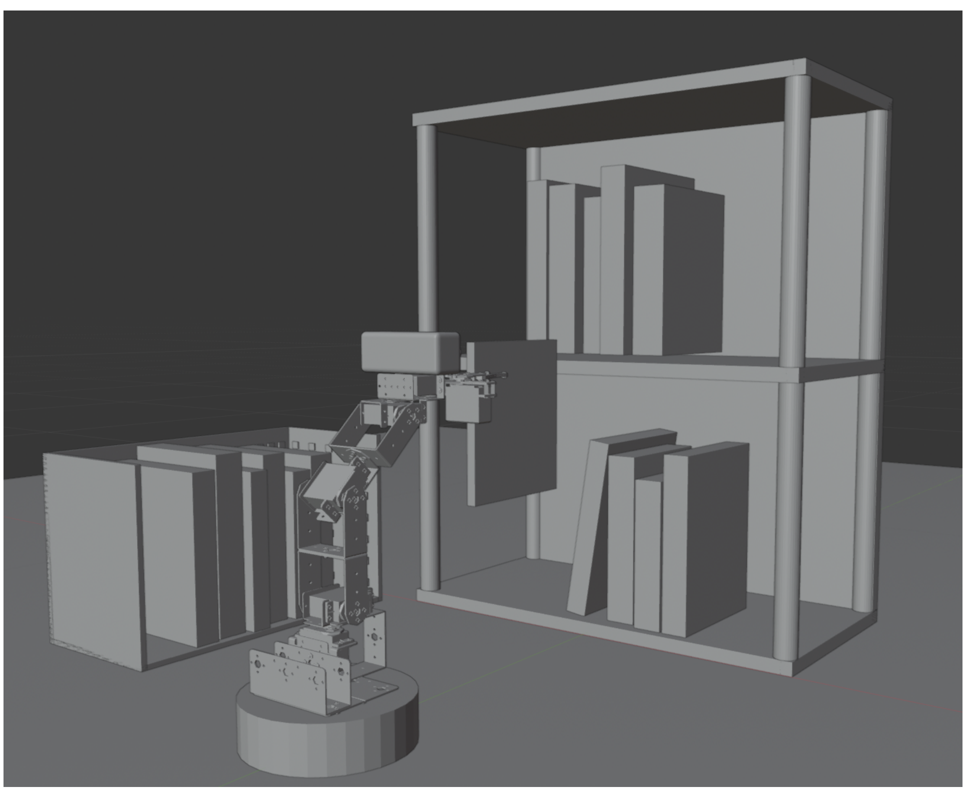
The system demonstration operates under the following setup:
Initial Conditions:
• Bookshelf is pre-populated with books (not empty) to simulate realistic library conditions
• Returned books are placed in a book tray positioned within the arm's reach
• The robotic arm is mounted on a movable platform allowing flexible positioning relative to the shelf

Demonstration Workflow (5 books, target ≤10 minutes total):
1. Robotic arm autonomously retrieves a book from the book tray

2. Camera scans the book spine to identify classification label (e.g., "A1", "B3" or color code)
3. Arm positions camera to scan bookshelf for available gaps using depth sensing
4. Planning module selects optimal gap based on priority algorithm (edge placement preferred over central gaps)
5. Visual servo guides precise insertion of book into selected gap
6. Arm returns to tray for next book
7. Process repeats for remaining books

Performance Targets:
• Total time for 5 books: ≤10 minutes (average 2 minutes per book)
• Book identification accuracy: ≥95%
• Gap detection accuracy: ≥90%
• Placement success rate: ≥90%

Testing Flexibility:
The bookshelf configuration (number and position of existing books) can be varied to create different environmental scenarios and challenge the system's gap detection and slot selection algorithms. Edge-case testing (densely packed shelves, minimal gaps) validates robustness.
Note: Complex insertion scenarios (inserting between tightly packed books in the middle of a row) are deprioritized in the current implementation. The intelligent slot selection policy avoids such cases by preferring edge placements and open gaps where available.



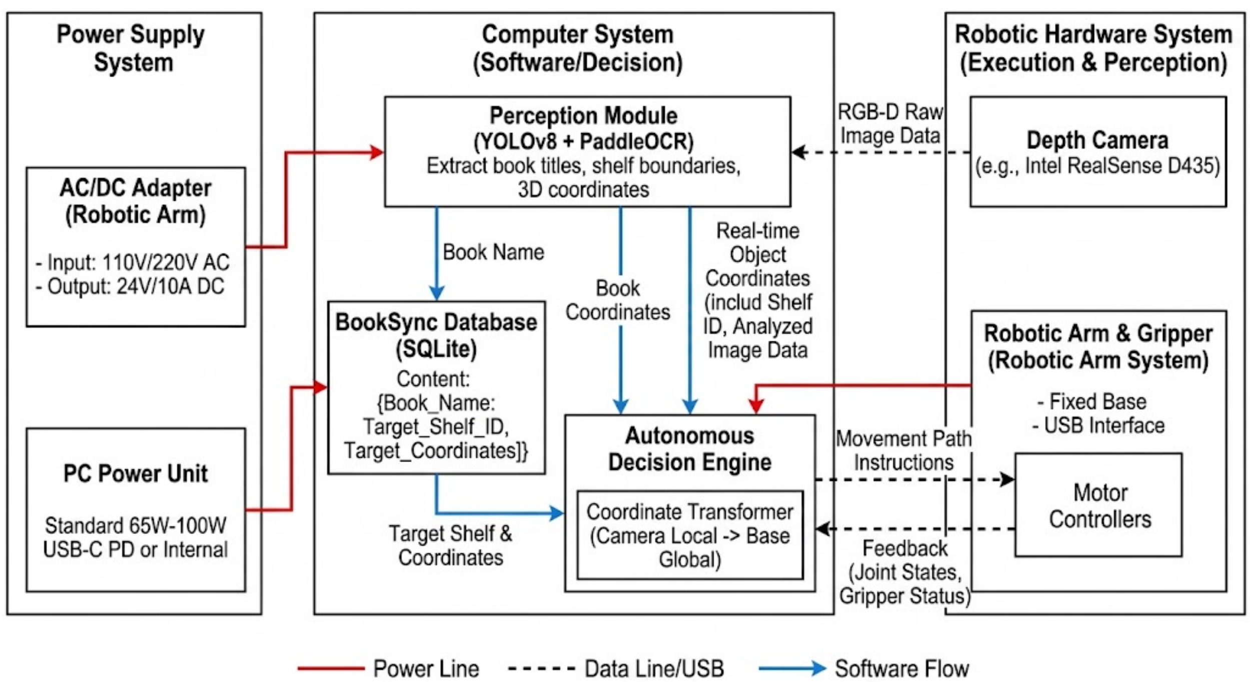
1.4 High-Level Requirements List

The system shall satisfy the following high-level requirements:

- The system shall autonomously complete a full reshelving cycle, including book identification, task planning, and placement execution, after a single user input, without requiring human intervention during operation.
- The system shall correctly identify at least 95% of books in a batch using barcode or vision-based recognition under normal operating conditions.
- The system shall achieve a placement success rate of at least 90%, where success is defined as placing each book within the correct shelf range and maintaining a stable upright or supported position.
- The system shall perform placement decisions based on real-time visual perception of shelf occupancy, including gap detection and fit evaluation, rather than relying solely on predefined positions.

2. Design

2.1 Block Diagram



2.2 Perception Subsystem (Subsystem I)

The Perception Subsystem is responsible for identifying books and analyzing shelf occupancy in real-time. It comprises two main functional components:

2.2.1 Book Identification Module

This module identifies returned books by processing visual data to extract unique identifiers, such as titles or barcodes.

- **Hardware Implementation:**

  The module utilizes an Orbbec Astra Pro RGB-D Camera mounted in an eye-in-hand configuration on the final link of the robotic arm, positioned away from the gripper to maintain an unobstructed view of the target area. This camera provides both RGB imagery at 1280×720 resolution and depth data at 640×480 resolution, both operating at 30 frames per second. The depth sensing capability offers ±3mm accuracy at 1 meter distance with a field of view of 58.4° horizontal by 45.5° vertical. The camera weighs approximately 310 grams and connects to the control computer via USB 2.0 interface, consuming less than 2.4 watts of power during operation.

- **Technical Solution (Algorithms):**

  The module employs a simplified classification approach suitable for controlled library environments:

  Book Identification Method:

  • Large labels or spine markings (e.g., "A1", "B3") are affixed to book spines

  • Color-based or template matching algorithms identify the classification label

  • Alternative: Simple color categorization (e.g., red books → Category A, blue books → Category B)

  Processing Pipeline:

  1. RGB image capture from Orbbec Astra Pro at 30fps

  2. ROI extraction focusing on book spine region

  3. Label/color detection using OpenCV pattern matching

  4. Classification result sent to database query module

  Processing Efficiency: The simplified pipeline completes identification within 100-150ms per

  frame, ensuring rapid feedback during autonomous operation.

  Baseline Reference:

  • Standard test book: 25cm × 16cm × 4cm (thickness), weight ≤300g

  • Actual books may vary slightly in size within ±10% range

- **Database Interface:**
  Once an identifier (e.g., Book Title) is extracted, the module initiates a query to the **BookSync Database**.

  Implementation: The database is built on **SQLite** for localized, low-latency access.

  Data Mapping: The module sends a standard SQL query (e.g., SELECT target_shelf, target_x, target_y FROM books WHERE title = ?).

  Latency Requirement: The database must return the target shelf ID and physical coordinates within **50 ms**.

- **Error Handling and Robustness:** To ensure the project's goal of a **95% identification rate** , the following error-handling strategies are implemented:

  Retry Logic: If the confidence score from PaddleOCR is below **0.85**, the system will automatically trigger the robotic arm to adjust the camera angle or distance and re-scan the book.

  Invalid Perception Handling: If identification fails after three attempts, the system flags the item as "Unrecognized" in the log and proceeds to the next item in the batch to maintain autonomous flow.

  Validation Checks: Before passing data to the decision engine, the module cross-references results with the database to prevent "hallucinated" titles from triggering unsafe motions.

Requirements	Verification
The book identification module shall achieve a >= 95% recognition accuracy for titles and barcodes under standard indoor lighting (300-500 lux).	Test with a batch of 20 books from the catalog. Record the number of successful identifications and calculate the percentage.
The system shall retrieve shelf locations and target coordinates from the SQLite database within 50ms per query.	Use software profiling tools to measure the timestamp from the moment a query is sent until the coordinate data is received.
The Perception Module shall process and extract identifiers from a single 1080p RGB frame in less than 200ms using YOLOv8 and PaddleOCR.	Run the identification pipeline on 50 test frames and calculate the average execution time using system clock logs.

2.2.2 Shelf Occupancy Perception Module

Hardware: The shelf occupancy perception module uses the same Orbbec Astra Pro RGB-D camera mounted on the robotic arm that is used for book identification. With specifications of 1280×720 RGB resolution and 640×480 depth resolution both at 30 frames per second, depth accuracy of ±3mm at 1 meter, and a field of view of 58.4° horizontal by 45.5° vertical, the camera provides comprehensive spatial awareness. The eye-in-hand configuration allows the robotic arm to actively position the camera for optimal shelf scanning angles. The camera-to-shelf distance is dynamically adjusted during operation to ensure complete coverage of each shelf layer, enabling real-time gap detection without requiring multiple camera installations or view stitching procedures.

Software — Gap Detection Algorithm: The pipeline operates as follows on each depth frame:

ROI Cropping: A predefined region of interest (ROI) corresponding to the target shelf layer is cropped from the full depth frame, eliminating the shelf boards and background.

Column-wise Depth Projection: For each horizontal pixel column within the ROI, the minimum depth value across all rows is computed, yielding a 1D depth profile $D[x]$ where x is the horizontal pixel index.

Threshold Segmentation: A threshold d_{thresh} is set as the shelf back-panel depth minus a margin (typically 50 mm). Columns with $D[x] > d_{thresh}$ are classified as empty (gap); columns with $D[x] \leq d_{thresh}$ are classified as occupied (book present).

Gap Localization and Width Measurement: Contiguous empty segments are identified. Each gap's pixel span is converted to physical width using the following scale factor:

$$w_{physical} = n_{pixels} \times (d \times \tan(\theta_{HFOV} / 2)) / (W_{pixels} / 2)$$

where d is the measured depth to the shelf back panel, $\theta_{HFOV} = 58.4°$, and $W_{pixels} = 640$.

Fit Evaluation: A gap is deemed fit for the incoming book if:

$$w_{gap} \geq w_{book} + \Delta_{margin}$$

where w_{book} is the book thickness retrieved from the database and $\Delta_{margin} = 10$ mm is a safety clearance.

Output Format: The module outputs a structured list of detected gaps per layer:

[{gap_id, center_x_mm, width_mm, fit: bool}, ...]

This is passed directly to the Intelligent Slot Selection module.

Requirements	Verification
The shelf occupancy module shall detect gaps with width ≥ 30mm at a camera-to-shelf distance of 60–70cm.	Place calibration blocks of known widths (30mm, 40mm, 50mm) on a test shelf. Run the detection pipeline and verify that all gaps ≥ 30mm are correctly identified. Repeat 20 trials.
Gap width measurement error shall be within ±10mm compared to physical ground truth.	Measure 10 gaps of known widths using a ruler; compare against algorithm output. Record mean absolute error and verify it is ≤ 5mm.
The module shall complete one full shelf-layer scan and gap output within 2000ms.	Log timestamps at frame acquisition and gap-list output. Compute mean processing time across 50 consecutive frames.

2.3 Task Planning and Scheduling Subsystem (Subsystem II)

The Task Planning and Scheduling Subsystem receives book identities, shelf locations, and real-time occupancy data from the Perception Subsystem, then determines the order and specific placement positions for each book.

2.3.1 Batch Ordering Algorithm

Algorithm Selection:
The module implements a **Nearest Neighbor (Greedy) Heuristic**.

Input and Output:

- **Input:** An unsorted list of 3D target coordinates [x, y, z] and Book IDs retrieved from the **BookSync Database**.
- **Output:** A prioritized **Execution Queue** of coordinates sent to the Slot Selector and Kinematic Engine.

Complexity and Performance:
The algorithm operates at **O(n^2)** time complexity. Given the batch size constraint of n=5 books, the computational overhead is negligible (less than 10ms), ensuring the system meets the high-level timing requirements.

Error Handling:
If a target coordinate provided by the database is flagged as "unreachable" by the Coordinate Transformer, this module removes that item from the current batch queue, logs a "Kinematic Path Error," and re-calculates the sequence for the remaining items to prevent system stalls.

2.3.2 Intelligent Slot Selection Policy

This module determines the optimal insertion point for a book within a targeted shelf by analyzing real-time occupancy data and geometric constraints.

- **Selection Logic:**
  The module receives a list of available gaps from the **Perception Subsystem (Shelf Occupancy Analyzer)**. It applies a **Density-Maximization Strategy**:
  1. **Candidate Filtering:** It filters out any gaps where the measured width W_gap < W_book + 15mm. The 15mm buffer accounts for the combined mechanical tolerance of the arm and the visual measurement error.
  2. **Proximity Ranking:** To maintain shelf organization, **the algorithm prioritizes slots immediately adjacent to existing books rather than isolated gaps in the middle of the shelf.**
  3. **Centroid Calculation:** Once a slot is selected, the module calculates the precise center coordinate [x, y, z]base to ensure the book is inserted without colliding with neighboring book spines.
- **Integration with Workflow:**
  This policy acts as the bridge between "what the shelf looks like" and "where the arm should go." It converts the abstract "Target Shelf ID" from the **BookSync Database** into a concrete 3D coordinate for the **Trajectory Planner**.
- **Error Handling:**
  If the target shelf is detected as 100% occupied or no gap meets the minimum width requirement, the module sends a "Shelf Full" interrupt to the UI. The task is then appended to a "Pending" list, and the system moves to the next book in the batch to avoid halting the entire operation.

Requirements	Verification
The slot selection algorithm shall prioritize gaps that are at least 20% larger than book width	Conduct 50 pick-and-place trials and count successful insertions without collisions.
Path generation time shall be <=300ms	Record the timestamp for trajectory calculation in the system log.

2.4 Manipulation Subsystem (Subsystem III)

The Manipulation Subsystem executes the physical placement of books using a robotic arm with a gripper, guided by visual feedback for precise and stable insertion.

2.4.1 Robotic Arm Hardware

Mechanical Configuration and Specifications

The robotic arm is designed as a 6-DOF articulated manipulator to provide sufficient flexibility for positioning and orientation within the constrained geometry of a bookshelf. The system utilizes a purchased aluminum frame that has been retrofitted with a custom servo control system. Key specifications include a working radius of 36 centimeters and a vertical reach of 46 centimeters, with the complete arm assembly including servos weighing approximately 700 grams. The system has been verified to support payload capacities of at least 500 grams through reference product testing. Mechanical positioning accuracy of ±5 millimeters is achieved through the rigid aluminum structure, with visual servo compensation further refining the final positioning accuracy to the target specification of ±2 millimeters. The arm's workspace is sufficient to cover bookshelf dimensions of 60 centimeters width, 30 centimeters depth, and 80 centimeters height when positioned appropriately. The system is designed for flexible base positioning rather than fixed installation, allowing adjustment based on task requirements and operational constraints.

Controller Type and Communication Protocol

The robotic arm is driven by a hybrid servo system combining stepper and serial bus servo motors across its six degrees of freedom. Joint 1, responsible for base rotation, utilizes a NEMA 17 stepper motor paired with a TB6600 driver, providing 52 Newton-centimeters of torque at 12 volts and controlled through step and direction signals from the Arduino controller. Joints 2 through 6, encompassing the shoulder, elbow, wrist, and gripper functions, employ LX-224HV serial bus servos. These servos deliver 20 kilogram-centimeters of torque at 7.4 volts with an operating voltage range of 6 to 8.4 volts. Each LX-224HV unit provides 0.24-degree resolution across a 0-1000 position range with ±1-degree accuracy, weighing approximately 60 grams per unit. The serial bus architecture enables all five LX servos to be controlled via a single data line using half-duplex UART communication at 115200 baud rate, simplifying wiring complexity while enabling efficient real-time position feedback for closed-loop visual servoing. The control architecture consists of an upper computer running Ubuntu 22.04 with ROS2 Humble for vision processing and path planning, and a lower computer using Arduino Mega 2560 for the motion control layer. Communication between these layers is established through a rosserial bridge via USB serial connection, with the control loop operating at 30 Hertz synchronized with the camera frame rate.

Workspace and Limitations

The operational workspace of the robotic arm covers the bookshelf region with dimensions of 60 centimeters width, 30 centimeters depth, and 80 centimeters height. The base positioning is flexible and adjustable rather than fixed, allowing the system to be repositioned as needed for optimal task execution. The 36-centimeter working radius provides adequate horizontal reach for the 60-centimeter shelf width when positioned centrally, while the 46-centimeter vertical reach allows access to both shelf layers with their 40-centimeter spacing when the base is appropriately elevated. The system is capable of 25-centimeter insertion depth into the 30-centimeter-deep shelves. Trajectory planning must account for existing books, shelf edges, and camera and gripper geometry to ensure collision avoidance. Motion planning implements joint limit checking and singularity detection to maintain safe operation throughout the workspace. The system prioritizes gap selection strategies that minimize insertion into densely packed regions, favoring edge positions and open spaces to reduce collision risk during demonstration operations.

2.4.2 Gripper Design

Gripper Type

The gripper adopts a three-finger configuration specifically designed for handling books in densely packed shelves, where direct side grasping is often infeasible. It consists of one upper finger and two lower fingers with distinct functional roles, enabling both extraction and stable grasping within confined spaces.

Grasping Control Method

The gripping process follows a two-stage strategy in which the upper finger first inserts into the gap between books and applies a controlled pulling force to partially extract the target book, thereby creating sufficient clearance. Subsequently, the two lower fingers close from both sides to securely clamp the book, while the upper finger assists in maintaining alignment during lifting and transport. The control scheme combines position control for coordinated motion and force-limited control to prevent excessive pressure on book surfaces.

Adaptability to Different Book Sizes

The gripper is designed to handle books with thicknesses ranging from 10 millimeters to 50 millimeters, with primary focus on the baseline specification of 40 millimeters thickness. Rather than implementing complex adjustable mechanisms, the design prioritizes sufficient clamping force combined with high-friction contact surfaces to securely grasp books within this range. Actuation is provided by a single LX-224HV servo motor, with gripper jaw spacing mechanically limited to prevent over-compression of books. This simplified approach reduces mechanical complexity while ensuring reliable performance across the target size range for the demonstration system.

2.4.3 Visual-Servo Placement Control

Camera Configuration:

The system utilizes an Eye-in-Hand configuration with the Orbbec Astra Pro depth camera mounted on the final link of the robotic arm. The camera is positioned away from the gripper to avoid occlusion while maintaining a clear line of sight to the target insertion region. This configuration allows the camera to move with the arm while providing an unobstructed view of the gap and surrounding books throughout the manipulation sequence. The depth sensing capability provides real-time three-dimensional spatial information for precise position-based visual servoing control.

Visual Servoing Strategy:
We implement **Position-Based Visual Servoing (PBVS)**. Since the depth camera provides real-time 3D coordinates (x, y, z) of the gap's centroid relative to the camera, the controller calculates the 3D pose error between the current gripper position and the target slot. This error is then translated into velocity commands for the robot joints to close the gap.

Feedback Loop & Frequency:

The control loop operates at 30 Hertz, synchronized with the depth camera's RGB-D frame rate which provides both RGB and depth streams at 30 frames per second. The autonomous decision engine processes the visual feedback and updates joint angle commands at this rate via the rosserial communication bridge to the Arduino Mega 2560, which in turn commands the LX-224HV servos through the serial bus protocol. The communication pipeline flows from camera capture at 30 frames per second through ROS2 node processing at 30 Hertz, then via rosserial at 30 Hertz to the Arduino, and finally to servo updates at the same 30 Hertz rate. The LX-224HV servos provide real-time position feedback at this same frequency, enabling closed-loop verification of commanded versus actual joint positions throughout the visual servoing process.

Trajectory Planning:
The module uses a **Trapezoidal Velocity Profile** for the approach trajectory. This ensures smooth acceleration and deceleration, preventing oscillations or "overshooting" as the book enters the narrow shelf gap.

Stability Verification:
Stability is verified through **Error Convergence Analysis**. The system monitors the Euclidean distance between the gripper and the target; the mission is aborted if the error does not decrease monotonically or if the calculated torque exceeds safety limits, preventing collisions with the shelf.

Requirements	Verification
The gripper shall securely hold books ranging from 10 mm to 50 mm in thickness without causing damage.	Test grasping on books of varying thickness. Inspect for slippage and physical damage after repeated trials.

The gripper shall achieve a grasp success rate of at least 95%.	Perform repeated grasp trials across different book sizes and record success/failure rates.
The robotic arm shall reach any point within a 50 cm radius workspace with ±2 mm positional accuracy.	Perform workspace calibration using a grid of target points. Measure end-effector position using a vision system or ruler-based reference. Compute positioning error and verify it is within ±2 mm.

2.5 Control System

Main Controller Selection:

The system uses a two-tier control architecture for the development and demonstration configuration. The upper computer consists of a laptop running Ubuntu 22.04 with ROS2 Humble, handling high-level processing tasks including vision processing, path planning, and decision-making, at no additional cost as it utilizes the existing development laptop. The lower computer employs an Arduino Mega 2560 for the motion control layer, responsible for servo commanding, position feedback, and trajectory execution. Communication between these tiers is established through a rosserial bridge via USB serial connection to the laptop, with updates occurring at 30 Hertz synchronized with the visual servo loop. This architecture separates computational tasks from real-time motor control, allowing the laptop to handle vision processing while the Arduino ensures deterministic servo updates. The design includes a hardware abstraction layer to facilitate future migration to embedded systems such as Orange Pi 5 or Jetson Nano if autonomous deployment is required beyond the demonstration phase.

Communication Protocols:

The camera-to-laptop connection utilizes USB 2.0 for the Orbbec Astra Pro, handling the RGB stream at 1280×720 resolution and depth stream at 640×480 resolution, both at 30 frames per second. The ROS2 interface employs either the openni2_camera or astra_camera driver package for seamless integration. Laptop-to-Arduino communication is established through USB serial using the rosserial bridge protocol, operating at 115200 baud rate with update rates of 30 Hertz for transmitting joint angle targets across all six degrees of freedom and receiving position readback from the servos. The Arduino-to-servo layer includes step and direction signals via TB6600 driver for the NEMA 17 stepper motor, and serial bus communication at 115200 baud using half-duplex UART for the five LX-224HV servos. The serial bus architecture enables a single data line to control all five servos through daisy-chained addressing, with bi-directional communication supporting both command transmission and real-time joint state telemetry for closed-loop control.

User Interface Design (One-button Operation): The system features a streamlined **"One-Button" GUI** developed using **PyQt5**.

- **Start Logic:** Upon pressing the physical or virtual "START" button, the system initiates a self-test (homing motors, checking camera feed). Once "Ready" is displayed, a second click triggers the full batch return cycle for 20 books.
- **Status Monitoring:** The UI provides a real-time progress bar, identified book titles, and an "Emergency Stop" dashboard widget.

Emergency Stop and Error Handling:

- **E-Stop Mechanism:** A dual-layer approach is implemented. A **physical latching kill-switch** cuts the **24V/10A DC power** to the motors immediately. Simultaneously, a software interrupt sends a **"NULL" command** to the motor controllers to lock the current position.
- **Error Handling:** If the vision module fails to identify a book with > 0.85 confidence, the system enters a "Retry" state, adjusting the arm angle by 5 degrees to re-scan. If 3 retries fail, it logs the error and moves to the next book.

Requirements	**Verification**
The control system shall respond to a physical or software emergency stop signal by cutting motor power within **100 ms (+/- 10 ms)**.	Trigger the E-stop while the arm is in motion. Use a digital oscilloscope to measure the latency from the switch signal to the drop in voltage on the 24V motor line.
The system shall complete one full cycle of returning 20 books (pick-and-place) within 10 minutes under autonomous mode.	Load 20 books into the bin and initiate the cycle. Use a stopwatch to record total time from "Start" button press to the final book being placed and the arm returning to home position.
The control heartbeat (feedback loop) between the PC and motor controllers shall maintain a frequency of 20 Hz (+/- 2 Hz).	Monitor the serial data packets using a logic analyzer or serial terminal. Record the timestamp of 100 consecutive

	"Feedback" packets and calculate the average frequency.

2.6 Power System

Total Power Requirement Estimation

The total power consumption of the system is calculated based on measured and specified values for all components. The NEMA 17 stepper motor consumes an average of 10 watts with peaks reaching 18 watts. The five LX-224HV servos collectively draw 18.5 watts on average with peak consumption of 92.5 watts, noting that not all servos reach maximum load simultaneously during typical operation. The Orbbec Astra Pro camera requires less than 2.4 watts, while the Arduino Mega 2560 consumes less than 1 watt as it is USB powered from the laptop. The laptop power supply is self-contained and not included in the system power budget. Total system power requirements amount to 31 to 35 watts for average continuous operation, with peak instantaneous demand reaching 113 watts when all servos and the stepper motor operate at maximum load. Typical operating power during active manipulation is conservatively estimated at 40 to 50 watts. The power system is designed with sufficient margin to handle peak loads while maintaining voltage stability throughout the operational cycle.

Power Supply Selection

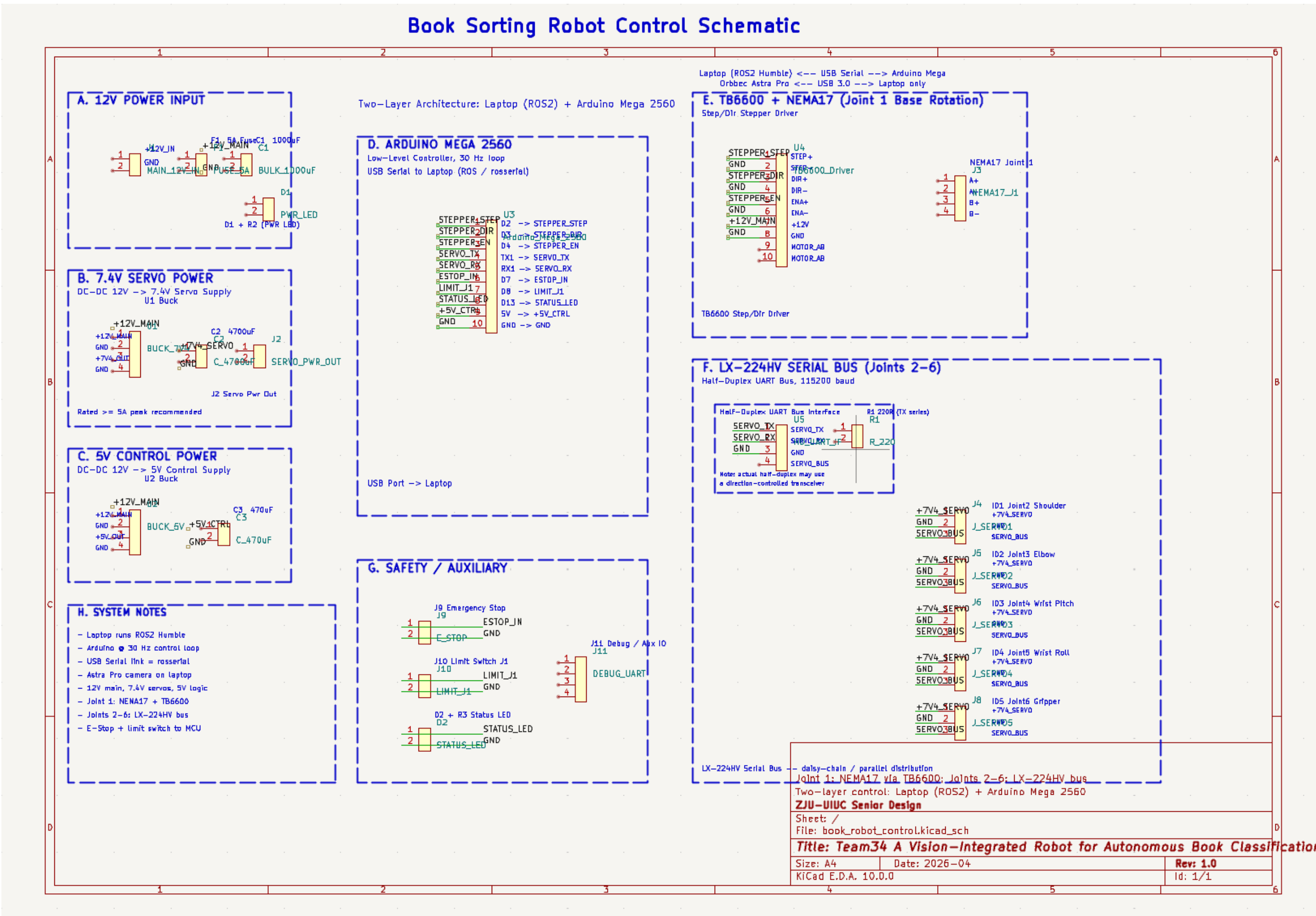
The system operates on an AC adapter power solution suitable for the demonstration configuration. The primary power supply consists of a 12-volt 10-ampere switching power adapter with 120-watt rated capacity, supplying the NEMA 17 stepper motor through direct 12-volt connection. Voltage regulation is achieved through a DC-DC buck converter stepping down from 12 volts to 7.4 volts with 15-ampere capacity, supplying all five LX-224HV servos via a common bus. Additional components include power cables, XT60 connectors, and a distribution board. Power budget verification confirms that the 120-watt supply capacity exceeds the 113-watt peak demand with a minimal but acceptable 6-percent margin for demonstration purposes. During normal operation, the 120-watt supply provides 243-percent margin over the 35-watt average consumption. The voltage distribution provides a 12-volt rail for the NEMA 17 stepper motor and a 7.4-volt rail for the LX-224HV servo bus, while the Arduino Mega receives 5-volt power via USB from the laptop and is therefore not part of the main power system. An alternative battery-powered configuration using a 3S 11.1-volt lithium polymer battery with 8000 milliampere-hour capacity could provide 2 to 2.5 hours of runtime, though this is not required for the demonstration phase.

Voltage Regulation and Power Management

To ensure stable operation, the system incorporates voltage regulators and power management circuits that maintain consistent output levels despite load variations. DC-DC converters are used to step down voltage for low-power electronics, while filtering and protection mechanisms such as capacitors, fuses, and overcurrent protection are implemented to prevent noise, voltage spikes, and potential damage to components. Power distribution is carefully designed to isolate sensitive electronics from high-current motor noise, improving overall system reliability.

Requirements	Verification
The power system shall provide stable 12 V ± 0.5 V output under maximum load conditions.	This requirement will be verified by operating the system at peak load and measuring the output voltage using a calibrated multimeter or oscilloscope to ensure that voltage variations remain within the specified tolerance range.
The power system shall supply sufficient current to all components without voltage drop or instability.	This will be tested by running all subsystems simultaneously under worst-case conditions and monitoring voltage levels and system performance for signs of instability or power-related failures.

2.7 Schematics



2.8 Tolerance Analysis

To maintain a high placement success rate (target >= 90%), the system accounts for uncertainty in perception, calibration, actuation, and book size variation by using conservative decision margins.

Perception tolerance:

Gap width from depth sensing contains measurement error e_depth. To prevent false-fit decisions, a gap is accepted only if:

w_gap >= w_book + Delta_safe

where Delta_safe = 10 mm provides clearance for sensing noise and shelf irregularities.

Calibration and actuation tolerance:

Hand-eye calibration error (e_cal) and servo/mechanical error (e_act) both affect insertion accuracy. The effective lateral placement error is considered as:

$$e_{total} = e_{cal} + e_{act} + e_{depth,proj}$$

Visual-servo correction is used near the final insertion stage to reduce this accumulated error.

Book/shelf variation tolerance:

Because book thickness and shelf geometry vary, the planner uses conservative thickness estimates and only selects gaps that satisfy the safety-margin criterion.

Engineering implication:

If no candidate gap meets the tolerance condition, the planner skips that gap and chooses another valid location. This slightly reduces space efficiency but significantly improves insertion reliability and operational safety.

3. Cost

Part	Cost (RMB)
Servo motor —20kg LX-224HV x5	391.5
Stepping motor — 42-40stepping motor 0.4N	58
Orbbec Astra Pro RGB-D Camera	199
Aluminium frame	100
Carbon plate	80
Arduino Mega 2560	307
Bookshelf	54.3
Books	36.98
Total Cost	**1226.78**

4. Schedule

Week	Xinrui Xiong	Zhecheng Lou	Zehao Bao	Zhenxiong Tang
Weeks 1-2	Robotic Arm Interface Redesign, Feasibility Analysis and Design	Robotic Arm Interface Redesign, Feasibility Analysis and Design	Researched CV approaches for shelf perception and technical stack	Model state diagram based on Block Diagram, clarify decision subsystem design
Weeks 3-4	Design new three-claw gripper, design and integrate fixed camera mount, system assembly	Design new three-claw gripper, design and integrate fixed camera mount, system assembly	Complete Orbbec Astra Pro depth stream verification, implement barcode recognition	Write main program code and develop Book Database
Weeks 5-6	Hardware-software integration, prototype construction & testing	Hardware-software integration, prototype construction & testing	Implement shelf occupancy perception and slot selection policy	Integrate vision, planning nodes into runnable baseline
Weeks 7-8	Calibrate gripper, debugging and testing	Verify camera positioning, debugging and testing	Implement visual-servo placement control	Refine overall performance, end-to-end validation

5. Ethics and Safety

5.1 Ethics

The design and implementation of an autonomous robotic system for library environments raise several ethical considerations related to reliability, user trust, and responsible system behavior.

First, system accuracy and reliability are critical. The system performs automated book classification and placement based on barcode recognition and perception algorithms. Incorrect identification or placement may lead to misclassification, which can negatively impact users' ability to locate materials. Therefore, the system must be designed to minimize recognition errors and clearly define its operational limits. Performance metrics such as recognition accuracy and placement success rate are explicitly defined, and the system should not be presented as fully reliable beyond these validated conditions.

Second, transparency and user awareness must be ensured. The system operates autonomously after a single user input, which may create an impression of complete independence. However, users should be informed that the system is a demonstrator and may require supervision in real-world deployment. The system should provide clear feedback on its operational status (e.g., running, completed, error state) to avoid confusion or misuse.

Third, responsible use of automation must be considered. While the system reduces manual labor, it should not be positioned as a complete replacement for human workers without considering broader social implications. In this project, the system is explicitly designed as a small-scale demonstrator and research platform, focusing on improving efficiency and supporting human operators rather than replacing them.

Fourth, data usage and privacy must be addressed. The system processes visual data from the environment and scans book identifiers. Although no personal user data is intentionally collected, images captured by the camera may unintentionally include individuals or sensitive information. Therefore, all image processing should be limited to real-time analysis without storing identifiable data unless necessary for debugging, and any stored data should be handled responsibly.

Finally, honest reporting of system capabilities is required. The project must not exaggerate performance, robustness, or generalizability. All reported results, including success rates and limitations, should accurately reflect experimental conditions and system constraints.

5.2 Safety

The system involves robotic manipulation, electrical components, and real-time operation, which introduce multiple safety risks that must be addressed through design and operational measures.

Mechanical Safety

The robotic arm and gripper introduce risks of collision, pinching, or unintended motion. These risks are mitigated by:

- Limiting the workspace to a fixed and controlled area, reducing interaction with users during operation.
- Using lightweight mock books and a scaled-down demonstrator environment to minimize potential damage or injury.
- Implementing motion constraints and speed limits in control algorithms to prevent abrupt or high-force movements.
- Designing the gripper to apply limited force to avoid crushing or damaging objects.

Additionally, the system should include a clearly accessible emergency stop or software interrupt to immediately halt motion if abnormal behavior occurs.

Electrical Safety

The system includes a single-board computer, servo motors, and power supply components, which present risks such as overheating, short circuits, or electrical faults. These risks are addressed by:

- Operating all components within their rated voltage and current limits.
- Using regulated power supplies and proper wiring to prevent overcurrent conditions.
- Ensuring proper insulation and secure connections to avoid exposed conductors.
- Monitoring system temperature and avoiding prolonged operation under excessive load.

Software and Control Safety

Since the system relies heavily on perception and decision-making algorithms, software-related risks must also be addressed:

- Implementing error handling for failed barcode recognition, invalid perception results, or unreachable placements.
- Preventing unsafe actions (e.g., attempting to insert a book into an insufficient gap) through validation checks before execution.
- Using closed-loop visual feedback during placement to correct deviations and reduce the risk of unstable insertion.
- Logging system behavior for debugging and identifying failure modes.

References

[1] IEEE, "IEEE Code of Ethics." [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.

[2] Association for Computing Machinery, "ACM Code of Ethics and Professional Conduct." [Online]. Available: https://www.acm.org/code-of-ethics.

[3] National Society of Professional Engineers, "Code of Ethics for Engineers." [Online]. Available: https://www.nspe.org/resources/ethics/code-ethics.

[4] International Organization for Standardization, ISO 10218-1:2011, Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots, 2011.

[5] International Organization for Standardization, ISO 10218-2:2011, Robots and robotic devices — Safety requirements for industrial robots — Part 2: Robot systems and integration, 2011.

[6] International Organization for Standardization, ISO 12100:2010, Safety of machinery — General principles for design — Risk assessment and risk reduction, 2010.

[7] Occupational Safety and Health Administration, "Machine Guarding eTool." [Online]. Available: https://www.osha.gov/etools/machine-guarding.

[8] Occupational Safety and Health Administration, "Control of Hazardous Energy (Lockout/Tagout), 29 CFR 1910.147." [Online]. Available: https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910.147.