



Final Year Project Report

Autonomous Cubic Rail Robot for High-Density Storage

Ilyas Umurbekov, Madina Zhakhina, Temirlan
Kaiyrbay

A thesis submitted in part fulfilment of the degree of

BSc in Robotics and Mechatronics

Supervisor: Zhanat Kappassov

Department of Robotics Engineering
School of Engineering and Digital Sciences
Nazarbayev University

May 5, 2026

Abstract

The rapid expansion of e-commerce has precipitated a need for warehousing solutions that maximize volumetric efficiency. Robotic Compact Storage and Retrieval Systems (RCSRS) address this by utilizing a grid-based architecture where robots operate from the top to retrieve vertically stacked bins. This project, “MiniCube,” details the transition of a theoretical concept into a functional, autonomous cubic robot ($21 \times 21 \times 30$ cm) operating on a 3×3 grid.

To overcome the limitations of tethered power and mechanical instability, the prototypes were upgraded to fully autonomous systems powered by 2300 mAh LiPo batteries. Significant mechanical innovations include a “sliding wall” mechanism for reliable X-Y axis switching and a rigid tape-based lifting system equipped with a 12V solenoid to eliminate payload oscillation. The dual-robot system operates on a 3×3 grid using ESP32-S3 microcontrollers and NEMA 17 stepper motors communicating over UART, utilizing Hall sensors and intersection magnets for precise grid positioning. Furthermore, a Python-based control algorithm was developed to solve the “digging problem”—the retrieval of bottom-layer packages by intelligently relocating upper bins to adjacent free slots. This system validates theoretical models of high-density storage found in recent literature and establishes a robust hardware foundation for future multi-agent research.

Acknowledgments

We would like to express our profound gratitude to the Mignon Robotics team for their invaluable insights into how high-density storage challenges are addressed within the modern logistics industry. Furthermore, we are deeply appreciative of their expert supervision and technical guidance regarding the mechanical design and mechanism development of our robotic system. We are also sincerely thankful to the Tactile Lab for their generous support, providing crucial equipment components, laboratory resources, and the necessary funding that made the fabrication and testing of this project possible.

Contents

1	Introduction	5
1.1	Background and Motivation	5
1.2	Project Actuality	5
1.3	Project Aims	5
2	Related Works	7
2.1	The Evolution of Compact Storage Systems	7
2.2	The “Digging” Problem and Puzzle-Based Storage	8
2.3	Dynamic Storage Assignment	8
2.4	Autonomous Navigation and Path Planning	9
3	Methodology	10
3.1	Engineering Rationale for Mechanical Selection	10
3.1.1	Kinematic Rationale: The S-Path Sliding Wall	10
3.1.2	Dynamics Rationale: Semi-Rigid Tape Lift	10
3.2	The Digging Algorithm	11
4	System Design and Integration	12
4.1	Mechanical Design Evolution	12
4.1.1	Locomotion: The Sliding Wall Mechanism	12
4.1.2	Payload Retrieval: The Tape-Based Lift	13
4.2	Electrical Architecture and Autonomy	14
4.3	System Integration and Test Field	14
4.4	RViz-Based Kinematic Simulation	15
5	Experiments and Results	18
5.1	Performance Metrics	18
5.2	Reliability and Comparative Analysis	18
6	Discussion	19
6.1	Design Optimization Outcomes	19
6.2	Algorithmic Efficiency	19

6.3	Potential Applications and Scalability	20
7	Conclusions and Future Work	21
7.1	Conclusions	21
7.2	Future Work	22
7.2.1	Multi-Agent Scaling and Coordination	22
7.2.2	Mechanical and Interface Optimization	22
7.2.3	Digital Twin and Large-Scale Simulation	22
A	Appendix A: Learning Portfolio	23
A.1	Set of Skills	25
A.2	Future Learning Goals	26
A.3	Conclusion	26
B	Appendix B: Ethical and Legal Issues	27
B.1	Informed Judgments and Trade-offs	28
B.2	Sustainability and Social Responsibility	28
B.3	Examples of Ethical Reasoning in Practice	29
B.4	Conclusion	29

Introduction

1.1 Background and Motivation

Modern logistics centers face a critical spatial challenge. Traditional aisle-based storage systems suffer from low density due to the floor space required for human operators and forklifts. To mitigate this, the industry is shifting toward Robotic Compact Storage and Retrieval Systems (RCSRS). In these systems, bins are stacked vertically in a dense grid, and robots maneuver on overhead rails to retrieve items [Azadeh et al., 2019]. While theoretical models for such systems exist, there is a gap in accessible, scalable hardware implementations that can validate these theories in physical environments. For example, e-commerce giants and urban micro-fulfillment centers can deploy this research to maximize vertical volume in expensive warehouse locations, buffering and retrieving orders autonomously for rapid last-mile delivery.

1.2 Project Actuality

The MiniCube project addressed the immediate need for agile, autonomous hardware capable of navigating high-density grids. Recent studies by Chen et al. [2022a] and Zou et al. [2018] emphasize that while RCSRS architectures maximize energy efficiency, they introduce complex operational bottlenecks regarding retrieval times and robot congestion. Our project provided a physical platform to test solutions to these bottlenecks, specifically the mechanics of “digging” for buried bins, a problem identified as NP-hard by Yue and Smith [2020].

1.3 Project Aims

The primary objective was to evolve the initial static robot concept into a fully autonomous agent. This transformation focused on mechanical refinement to ensure stability and reliable axis switching, the achievement of energy autonomy by eliminating external tethers, and the development of algorithmic intelligence for solving bin

retrieval puzzles.

Related Works

This chapter situates the MiniCube project within the current landscape of robotics and logistics research. It analyzes the progression from traditional warehousing to high-density automated systems, defines the mathematical complexities of retrieving “buried” inventory, and examines the navigational strategies required for autonomous agents.

2.1 The Evolution of Compact Storage Systems

The logistics industry is currently undergoing a fundamental shift driven by the demands of e-commerce, which requires warehouses to handle higher volumes of smaller orders with greater speed. Roodbergen and Vis [2009] established in early surveys that traditional aisle-based Automated Storage and Retrieval Systems (AS/RS) are inherently limited by the “waste” of space required for aisles and human accessibility. Consequently, the industry has pivoted toward Robotic Compact Storage and Retrieval Systems (RCSRS), where bins are stacked vertically in a grid, eliminating aisles entirely. Azadeh et al. [2019] classify this evolution as a move toward “grid-based” architectures, noting that while these systems maximize storage density, they introduce significant complexity in retrieval mechanics.

The operational viability of such systems has been further validated by Trost et al. [2023], whose simulation studies confirm that RCSRS architectures offer superior throughput compared to traditional methods, provided the robot-to-bin ratio is optimized. Furthermore, Chen et al. [2022b] and Carabin et al. [2017] highlight that these compact systems are not only spatially efficient but also energy-efficient, as they reduce the travel distance required per retrieval. The MiniCube project is directly motivated by these findings, aiming to provide a physical hardware platform to validate these theoretical efficiency gains in a scaled-down environment.

2.2 The “Digging” Problem and Puzzle-Based Storage

A unique and critical challenge introduced by RCSRS is the retrieval of items located at the bottom of a stack. In a Last-In-First-Out (LIFO) stack structure, accessing a lower bin requires the temporary relocation of all bins resting above it. Yue and Smith [2020] mathematically define this as the “digging problem,” proving that minimizing the time required to extract a buried bin is an NP-hard problem. This establishes that simple random movement of blocking bins is inefficient; intelligent relocation logic is required.

To address this, Gue and Kim [2007] introduced the concept of “Puzzle-Based Storage,” likening the movement of bins to a sliding-tile puzzle where items must maneuver into empty cells to clear a path. This theoretical framework forms the basis of our project’s algorithmic control. While Bozer and White [1984] provided foundational travel-time models for accessing bins in standard racks, these models fail to account for the vertical reshuffling penalties inherent in RCSRS. Therefore, our project adopted the operational policies analyzed by Zou et al. [2018], specifically the “immediate reshuffling” strategy, where blocking bins are moved to the nearest available neighbor slot rather than a remote storage location. This minimized the energy expenditure of the robot, a critical factor for the battery-powered prototype.

2.3 Dynamic Storage Assignment

To mitigate the frequency of digging, the arrangement of bins within the grid must be dynamic rather than static. Yuan et al. [2021] demonstrate that storage assignment—the decision of where to place a returned bin—significantly impacts future retrieval times. Their research suggests that high-correlation products should be grouped, a logic we integrated into our robot’s sorting algorithm.

Building on this, Chi et al. [2022] propose dynamic picking and storage optimization, arguing that the system should autonomously re-sort bins based on “turnover rates.” Bins that are requested frequently should be kept at the top surface of the grid, while low-demand items sink to the bottom. Zaerpour et al. [2017] describe this as a “live-cube” framework, where the storage configuration is fluid. Our robot implemented this logic by utilizing idle time to reorganize stacks, ensuring that the system state was always optimized for the next retrieval command. Cai et al. [2021] further support this by proving that collaborative optimization of storage location and path planning yields the highest throughput.

2.4 Autonomous Navigation and Path Planning

Transitioning the robot from a static mechanism to an autonomous vehicle required robust navigation logic. Patle et al. [2019] provide a comprehensive review of path planning strategies, distinguishing between classical approaches (like A*) and reactive approaches (like sensor-based navigation). Given the dynamic nature of our grid, where obstacles (bins) change positions, a static map is insufficient.

Zhong et al. [2020] propose a “Hybrid Path Planning” method combining Safe A* algorithms with adaptive window approaches to handle dynamic environments. This informed our decision to use Hall Effect sensors for real-time localization, allowing the robot to detect its position relative to the grid rails. Furthermore, Chen et al. [2022a] discuss blocking protocols such as “Wait-Outside-Aisle,” which prevents deadlocks in multi-robot systems. While the developed prototype involved a single agent, the navigation code was structured to accommodate the “Block-and-Recirculate” protocols described by Wang and Qiu [2011] and Lamballais et al. [2017], in preparation for the planned multi-agent expansion. Finally, Zhao et al. [2024] emphasize the importance of Human-Robot Collaboration in these systems; the control interface was designed to allow human operators to interrupt autonomous behaviors safely, ensuring the system remains practical for real-world warehouse deployments.

Methodology

This chapter describes the high-level strategy, mathematical rationale, and algorithmic framework developed to enable autonomous retrieval in high-density grids.

3.1 Engineering Rationale for Mechanical Selection

The design of the MiniCube robot was guided by the need to maximize operational reliability while minimizing spatial footprints. A primary focus was the reduction of payload instability and mechanical binding during axis switching.

3.1.1 Kinematic Rationale: The S-Path Sliding Wall

The transition to a synchronized S-path sliding wall mechanism addressed the kinematic instability of axis switching. Previous cam-shaft designs suffered from asymmetric engagement, leading to mechanical binding. The S-path geometry ensures that both sides of the chassis engage the rails simultaneously, maintaining parallelism between the robot and the grid—a fundamental requirement for maintaining the ± 3 mm positional accuracy required for multi-robot navigation.

3.1.2 Dynamics Rationale: Semi-Rigid Tape Lift

The selection of a semi-rigid tape-based lift over a traditional rope hoist was driven by the need to minimize payload oscillation. In a traditional hoist, the payload behaves as a pendulum with a period $T = 2\pi\sqrt{L/g}$. As the robot accelerates or decelerates, the suspended package oscillates, causing the latch to miss the target handle. By using a semi-rigid strip, we constrained the degrees of freedom of the payload strictly to the Z-axis, effectively eliminating the pendulum effect. This mechanical constraint ensures that the latch remains aligned with the package handle during high-speed vertical transitions, ensuring operational reliability during autonomous retrieval sequences.

3.2 The Digging Algorithm

We developed a Python-based control system to handle the “digging” scenario, which is critical for retrieving items not located on the top layer. The process begins with a state analysis where the robot queries the grid database for target package coordinates. If the package is buried, the algorithm identifies all blocking bins stacked above it and scans adjacent grid slots for free space. Finally, a relocation loop moves the blocking bins to the nearest free neighbors, allowing the robot to retrieve the target bin once the stack is cleared.

This implementation provides a practical solution to the theoretical storage constraints analyzed by Lamballais et al. [2017] and Trost et al. [2023].

System Design and Integration

This chapter details the mechanical and electrical implementation of the MiniCube robot, along with the integration of sensors and simulation environments.

4.1 Mechanical Design Evolution

The physical development of the MiniCube followed a rigorous iterative process, transitioning from a basic non-functional frame to a fully integrated autonomous unit that prioritizes both mechanical precision and payload stability. The robot’s body compactly integrates distinct mechanical sub-systems within a single enclosure. We employ four NEMA 17 stepper motors equipped with MKS servo drivers to handle all major movements: one motor is dedicated to X-direction movement, one motor handles Y-direction movement, one motor is used for the lift mechanism, and the fourth motor is used for moving the sliding wall through an S-shaped path. This S-shaped path mechanism is responsible for changing travel from the X to Y direction (and vice versa) by lifting the respective side of the side wheels.

Accommodating all four motors within a single chassis required careful internal space planning. A key outcome of this compact integration is the support for a standardized package box size of $15 \times 15 \times 15$ cm—a symmetric cubic form factor that aligns with small-parcel dimensions common in micro-fulfillment and last-mile delivery workflows. The cubic package footprint simplifies grid cell design, as each storage cell can be dimensioned identically to accommodate both the robot and its payload, reducing manufacturing complexity and improving scalability for multi-cell grids.

4.1.1 Locomotion: The Sliding Wall Mechanism

The most significant mechanical challenge was the reliable switching between X and Y travel axes. Initial designs relied on passive friction or fixed wheel assemblies, which frequently resulted in jamming at rail intersections due to minor grid misalignments. To resolve this, we engineered an active “sliding wall” mechanism. In the previous version, the sliding wall was actuated using a cam shaft, which resulted in inconsistent

downward movement—the robot could fall to one side before the other. In the second version, we replaced this with a mechanism featuring two S-shaped control paths. When the wall slides from left to right, it pushes the lower wall down evenly via the control paths and bearings, ensuring a perfectly parallel engagement of the perpendicular rail set.

A key design improvement in this iteration is the dedicated allocation of the four NEMA 17 stepper motors. The sliding wall is driven by one motor that actuates the S-shaped path mechanism, allowing the robot to change directions by lifting the respective side wheels. Meanwhile, the X and Y direction movements are decoupled and driven by their own individual NEMA 17 motors. We transitioned from expensive Dynamixel motors to NEMA 17 steppers paired with MKS servo drivers because they offer a more cost-effective solution while providing reliable UART-based control and simplified integration within the ESP32-S3 ecosystem.

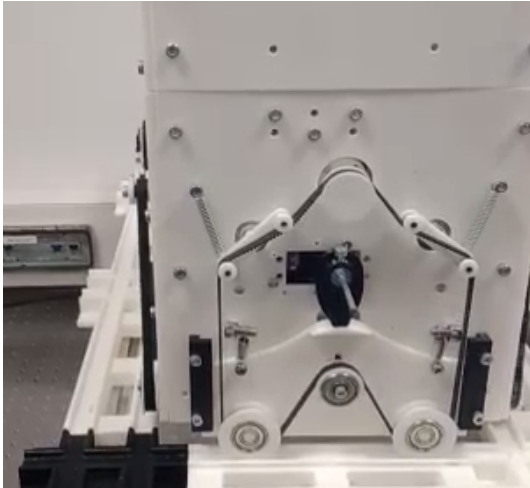


Figure 4.1: Previous MiniCube Version with Cam Shaft Mechanism

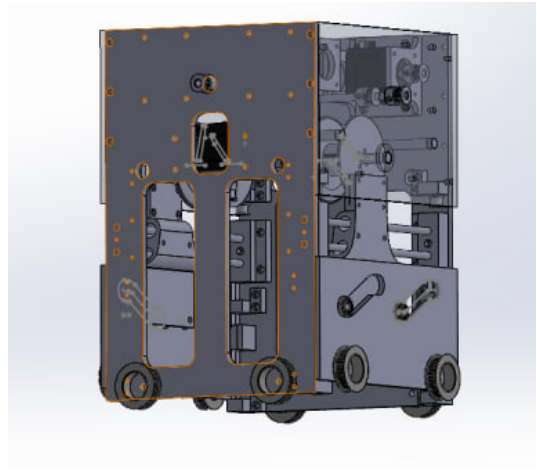


Figure 4.2: New MiniCube Version (V2) with S-shaped Control Paths

4.1.2 Payload Retrieval: The Tape-Based Lift

A critical flaw in the early prototype was the instability of the Z-axis lifting mechanism. The original design utilized a flexible rope, which behaved as a pendulum during operation. As the robot accelerated or decelerated, the suspended package would oscillate, causing the latch to miss the target handle or drop the payload. To solve this, we replaced the rope with a semi-rigid retractable tape (strip) system, powered by the dedicated fourth NEMA 17 motor. Additionally, we integrated a 12V solenoid for the lift latching mechanism. Similar to a measuring tape, the strip becomes rigid when extended, resisting lateral forces. This engineering change constrained the payload movement strictly to the vertical axis, eliminating oscillation and ensuring a reliable latching success rate across 50 consecutive automated retrieval cycles.

To accommodate the motor layout and the sliding wall drive in the upper section of the chassis, the tape rollers and spool assembly were repositioned slightly downward within the robot body. This deliberate layout choice reserves the upper interior volume for the motor mounting brackets, while still maintaining an unobstructed vertical channel through which the tape extends and retracts. The repositioned roller configuration does not affect lift performance, as the tape travel distance and spool capacity were adjusted accordingly.

4.2 Electrical Architecture and Autonomy

Transitioning the robot from a tethered laboratory unit to a fully autonomous vehicle required a complete overhaul of the power and control architecture. We integrated a 2300 mAh LiPo battery to supply power to the ESP32 microcontroller, the four motors, the solenoid, and the Hall sensor. The battery output is managed to ensure the motors and solenoid receive sufficient current at 12V, providing approximately 45 minutes of continuous operation under full-load conditions.

A significant change in the last design iteration was the complete replacement of all Dynamixel MX-28 servo motors with four NEMA 17 stepper motors paired with MKS servo drivers. The MX-28 servos, while precise, introduced considerable cost, weight, and control overhead. The NEMA 17 steppers, paired with MKS servo drivers, talk with the microcontroller using UART mode, drastically simplifying the control stack. This substitution reduced the total actuator cost by a factor of several times, lowered the overall chassis weight, and eliminated the need for a separate serial bus.

To manage high-level decision-making and low-level motor control, we utilized an ESP32-S3 N16R8 microcontroller. For navigation, a Hall sensor was embedded into the chassis. This sensor detects magnetic markers within the rail grid, allowing the robot to track its position during transit. Furthermore, a 3.3V logic relay was added to easily control the on/off state of the 12V solenoid.

4.3 System Integration and Test Field

To validate the operations of the developed autonomous robots, a comprehensive physical test field was constructed. The environment consists of a 3×3 grid equipped with two railways, meaning two or more robots can move simultaneously within the same grid. We have successfully assembled two such robots for this environment.

For reliable localization, 9 magnets were located exactly at the intersections of the grid. The onboard Hall sensor of each robot detects these magnets to determine its position. The ESP32-S3 N16R8 microcontroller coordinates all tasks by communicating

with the four MKS servo-driven NEMA 17 motors over UART, and utilizing the relay to actuate the 12V lifting solenoid.

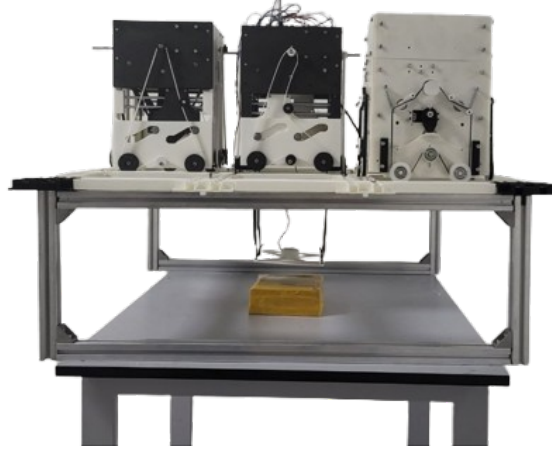


Figure 4.3: Experimental setup featuring three robotic units on the 3×3 grid: two new identical robots and the original prototype unit.

4.4 RViz-Based Kinematic Simulation

In addition to the physical prototype, a simulation environment was developed using ROS 2 and RViz2 to visually demonstrate the system’s behavior and, in particular, the “digging” process. The primary goal of this component was not to build a physically accurate simulator, but rather to create a clear, controlled, and reproducible visualization that could effectively communicate the algorithmic workflow of the system.

The simulation represents a simplified 3×3 grid-based storage environment, consistent with the physical setup of the MiniCube prototype. Two robots are modeled, along with a pair of stacked packages located in a single grid cell. The robots are defined using the Unified Robot Description Format (URDF), where their motion is represented through a series of prismatic joints corresponding to movement along the X and Y axes, as well as a vertical lifting mechanism along the Z axis. This kinematic structure mirrors the real robot’s movement capabilities while remaining simple enough for reliable simulation.

For visualization purposes, a hybrid modeling approach was used. The grid and robot base were represented using STL mesh files to preserve their geometric appearance, while simpler elements such as the lifting mechanism and packages were modeled using basic box geometries. This combination allowed for both visual clarity and computational simplicity, which proved important when debugging alignment and motion issues.

A custom Python node was developed in ROS 2 to control the simulation. This

node publishes joint states to the `/joint_states` topic, which are then interpreted by RViz2 through the `robot_state_publisher`. Rather than implementing a fully dynamic or sensor-driven control system, the digging behavior was encoded as a pre-defined sequence of actions. This decision was made deliberately to ensure predictable and stable execution, particularly given the time constraints and the focus on demonstration.

The simulated sequence closely follows the logical steps of the digging algorithm. First, the initial robot navigates to the stack and removes the top package. It then transports this package to a neighboring free cell, effectively clearing access to the lower package. Once the path is cleared, the second robot moves to the same location, retrieves the previously blocked package, and delivers it to a designated output cell. Finally, the first robot returns the temporarily relocated package back to its original position, restoring the stack configuration.

One of the more challenging aspects of the simulation was representing the interaction between the robots and the packages. Since URDF does not support dynamic attachment or detachment of objects, a different approach was required. The packages were modeled as independent kinematic entities with their own joints, separate from the robot structure. During specific phases of the sequence, their positions were programmatically synchronized with the robot motion, creating the visual effect of grasping and transporting. When the robot “released” the package, its position was fixed again relative to the grid. Although this approach is not physically realistic, it proved to be an effective and lightweight solution for visualizing manipulation tasks within RViz2.

Another significant part of the development process involved fine-tuning spatial alignment. Ensuring that the robots moved accurately between grid cells required careful adjustment of parameters such as cell size, joint limits, and origin offsets. Small inconsistencies between the URDF coordinate frames and the control logic could lead to noticeable visual errors, such as robots stopping between cells or packages appearing off-center. These issues were resolved iteratively through testing and calibration, highlighting the importance of consistent coordinate systems in robotic simulation.

Despite its simplicity, the final simulation successfully demonstrates the key concept of the digging process in a clear and intuitive manner. It provides a valuable visual complement to both the physical prototype and the theoretical algorithm, making it easier to understand the sequence of operations involved in retrieving buried items. While the current implementation is primarily intended for visualization, it also serves as a foundation for future development. With further enhancements—such as integration of real sensor data, collision handling, and scalable multi-agent coordination—the model could evolve into a more advanced simulation framework or a digital twin of the physical system.

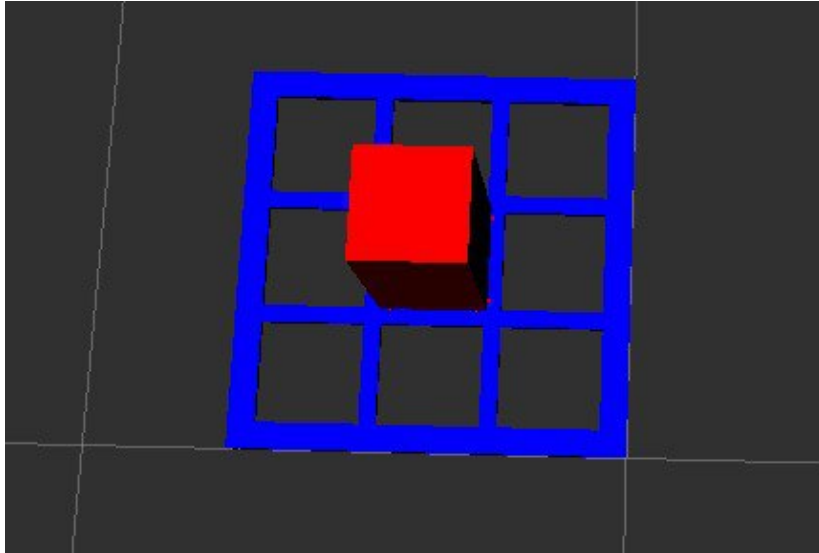


Figure 4.4: Kinematic Digital Twin of the MiniCube Platform in RViz2

Experiments and Results

This chapter presents the quantitative evaluation of the MiniCube prototype based on experimental trials conducted on the 3×3 grid.

5.1 Performance Metrics

The mechanical and electrical refinements resulted in a robust operational profile. The final prototype achieved a total system mass of 4.0 kg while supporting a maximum payload of 1.0 kg. This represents a significant improvement over the previous iteration, which weighed 5.5 kg and was limited to a 0.5 kg payload. Key performance indicators included a maximum linear velocity of 0.5 m/s and a navigation precision of ± 3 mm relative to the rail center, enabled by Hall Effect sensors. The 2300 mAh LiPo battery proved sufficient for approximately 45 minutes of continuous operation under full-load conditions.

5.2 Reliability and Comparative Analysis

To evaluate the operational stability of the S-path sliding wall mechanism, a series of experimental trials were conducted consisting of 10 continuous retrieval cycles per robot. Each cycle involved multiple axis-switches and package retrievals to simulate a high-traffic warehouse environment.

During these trials, the final MiniCube prototype exhibited a rail misalignment error rate of 15%, where the robot required a minor software-driven adjustment to center itself at an intersection. In contrast, the previous prototype iteration exhibited a significantly higher error rate of 45% under identical conditions. The 30% reduction in alignment errors is directly attributed to the synchronized engagement of the S-path mechanism, which prevents the chassis from tilting or binding when transitioning between orthogonal rails.

Discussion

This chapter provides a critical interpretation of the results and discusses the broader implications of the design choices made during the project.

6.1 Design Optimization Outcomes

The iterative refinement of the 3D-printed chassis and actuator selection yielded three measurable improvements over the initial prototype. First, the optimized internal layout increased package capacity to support standardized $15 \times 15 \times 15$ cm boxes. Second, replacing proprietary Dynamixel MX-28 servos with NEMA 17 stepper motors and MKS drivers significantly reduced the per-unit actuator cost. Finally, the transition to a belt-driven S-path design for the sliding walls ensured smooth and synchronized lateral engagement, eliminating asymmetric binding and ensuring consistent axis-switching cycles.

6.2 Algorithmic Efficiency

Testing the “digging” logic revealed that the relocation of a single blocking bin took an average of 30 seconds, including axis-switching and lifting cycles. The rigid tape-based lift contributed significantly to this efficiency by reducing the time spent waiting for payload oscillations to dampen.

While the heuristic approach is not globally optimal compared to more complex NP-hard solutions, it prioritized stability and real-time predictability, which are essential for physical hardware deployments. However, it is important to concede that while this local heuristic is stable for a 3×3 grid, it may lose energy efficiency as the grid scales. Specifically, moving a bin to the nearest neighbor without global optimization can lead to “shuffling traps,” where the same bin must be relocated multiple times to clear paths for future retrievals in a larger, more complex system.

6.3 Potential Applications and Scalability

While the current dimensions of the MiniCube ($21 \times 21 \times 30$ cm) and the small-scale grid are not directly suitable for large-scale industrial warehouses, the system demonstrates high potential for niche high-density applications. One particularly promising field is the automation of microgreen farms. Microgreens require high-density vertical stacking and frequent, precise handling of small trays—requirements that align perfectly with the MiniCube’s current form factor, payload capacity, and autonomous “digging” logic. In such an environment, the robots could autonomously retrieve and relocate growing trays for harvesting or inspection, maximizing space utilization in urban farming contexts where floor area is at a premium.

Conclusions and Future Work

This chapter summarizes the outcomes of the completed prototype development and outlines directions for future research. The progression from a static concept to a fully autonomous robotic agent provided critical insights into the mechanical and algorithmic requirements of high-density storage.

7.1 Conclusions

The primary objective of this project was to validate the feasibility of a miniature, autonomous cubic robot capable of navigating a three-dimensional storage grid. The successful fabrication and testing of the “MiniCube” prototype have demonstrated that the theoretical constraints of Robotic Compact Storage and Retrieval Systems (RCSRS) can be overcome through specific mechanical and control innovations.

Mechanically, the project resolved the critical issue of locomotion stability. Early iterations suffered from friction binding during axis switching, a common failure mode in rail-based robots. By engineering an active “sliding wall” mechanism, we decoupled the X and Y drive trains, ensuring that the robot can transition between orthogonal rails without mechanical interference. Furthermore, the instability of the vertical retrieval process—identified by Azadeh et al. [2019] as a bottleneck in system throughput—was mitigated by replacing the traditional rope hoist with a rigid tape-based lift. This modification eliminated payload oscillation, allowing the robot to latch onto packages reliably even at higher operating speeds.

Algorithmically, the project successfully transitioned “Puzzle-Based Storage” theory into practice. While Yue and Smith [2020] define the minimization of digging times as an NP-hard problem, the Python-based heuristic algorithm demonstrated that a simplified logic was sufficient for real-time control in a 3×3 grid. By scanning for empty neighboring cells and executing a relocation loop, the robot autonomously retrieved “buried” inventory without human intervention. The successful testing of the 0.5 m/s travel speed and ± 3 mm accuracy validated the hardware platform for future multi-agent expansion and confirmed the operational viability of dynamic storage assignment strategies proposed by Yuan et al. [2021] on physical hardware.

7.2 Future Work

With the single-agent hardware validated and the core algorithmic pipeline demonstrated, future development is directed toward scaling, optimization, and multi-agent simulation. These next steps are motivated by the need to test the system under the high-traffic conditions typical of real-world logistics centers.

7.2.1 Multi-Agent Scaling and Coordination

The efficiency of an RCSRS relies on parallelism; a single robot cannot match the throughput of traditional systems. As part of this work, we successfully fabricated a second identical robot to introduce multi-agent dynamics to the grid. This addition necessitates the implementation of complex traffic management protocols. As noted by Chen et al. [2022a], multi-robot systems are prone to gridlock. The “Wait-Outside-Aisle” and “Block-and-Recirculate” collision avoidance protocols would be implemented to manage right-of-way rules, ensuring that two robots can operate on the same grid without occupying the same cell or deadlocking at intersections.

7.2.2 Mechanical and Interface Optimization

Although the sliding wall mechanism functioned correctly throughout testing, high-speed operation generated vibration that could interfere with the Hall Effect sensors used for localization. A further design iteration would refine the wheel-to-rail interface using tighter tolerances and vibration-dampening materials to smooth the ride. This optimization would be critical for increasing the travel velocity v_{max} toward the theoretical limits described by Bozer and White [1984], thereby reducing the total cycle time per retrieval.

7.2.3 Digital Twin and Large-Scale Simulation

Physical testing was constrained by the size of the prototype grid. To validate the digging algorithm against large-scale inventory scenarios, a discrete event simulation (DES) approach, as recommended by Trost et al. [2023], is planned as a next step. Using kinematic data harvested from the physical robot (acceleration, lift time, switching time), a larger grid populated with thousands of bins would be simulated. This would allow stress-testing the storage assignment logic against Pareto-distributed order profiles (where 20% of items account for 80% of orders), verifying whether the dynamic reassignment strategy effectively reduces retrieval times at scale.

Appendix A: Learning Portfolio

Our capstone project started as a compact robotics idea: design a cubic robot that could move across a rail grid, retrieve bins from stacked storage, and work autonomously inside a high-density warehouse system. On paper, the concept looked manageable. Once development began, the project became much more complex. Mechanical design, electrical architecture, embedded programming, and path-planning logic were tightly connected. A change in one subsystem often created new problems somewhere else. Because of that, the most important learning did not happen from theory alone, but from repeated cycles of testing, failure, redesign, and gradual improvement.

At the beginning, we did not fully understand how difficult system integration would be. We had worked with motors, sensors, CAD models, and software separately, but combining them into one reliable machine was a different challenge. For example, we assumed that if each component functioned individually, then the assembled robot would also function correctly. This proved false almost immediately. Motors that worked smoothly on a bench produced vibration once mounted inside the chassis. Sensors that gave stable readings in isolation became noisy when placed near moving mechanical parts. Power wiring that seemed sufficient caused resets when multiple actuators drew current at the same time. These experiences showed us that robotics problems are rarely isolated; they emerge from interaction between components.

One of the largest technical challenges was motion across two rail axes. The robot had to switch between X and Y directions while staying stable on the grid. Our first design used simpler switching methods, including a cam-shaft based mechanism, but the movement was inconsistent. In several tests, one side of the robot engaged before the other, causing the chassis to tilt and jam at intersections. At that stage, we did not know how sensitive alignment tolerances could be, especially when several 3D-printed parts were assembled together. After repeated failures, we redesigned the system into a synchronized sliding-wall mechanism. Instead of relying on passive contact, the robot actively engaged the required axis. This made movement smoother and reduced asymmetry. Through this process, we learned practical mechanical reasoning: when failure repeats in the same location, geometry and tolerances usually need attention.

Another key challenge was the lifting system used to retrieve storage bins. Our

early prototype used a rope-based mechanism. In theory, it could raise loads vertically. In practice, the payload swung like a pendulum whenever the robot accelerated or stopped. This caused alignment errors and unreliable latching. At first, we tried solving the issue through slower motion commands, treating it as a control problem. Later we understood that the real cause was structural flexibility. We replaced the rope with a semi-rigid tape-based lifting system. Once extended, the tape resisted sideways motion and constrained the payload to a mostly vertical path. After this redesign, lifting became much more reliable. This taught us an important engineering lesson: not every unstable system needs better code; sometimes it needs a better mechanism.

The electrical system introduced another set of learning experiences. We moved from a tethered laboratory prototype to a fully autonomous robot powered by a LiPo battery with onboard voltage regulation. Before this project, we had limited practical experience with mobile robot power systems. During testing, the ESP32 microcontroller occasionally reset while motors were under heavy load. Initially, the problem seemed random. After measuring the system more carefully, we traced the issue to voltage drops caused by shared current demand. We solved it by separating sensitive electronics from motor power lines and improving regulation. This experience taught us to investigate hidden physical causes instead of assuming faults are random.

Software development also became a major learning area, especially through ROS 2 and simulation tools. At the start, we did not have a strong practical understanding of nodes, topics, transforms, and publishers. We knew the definitions, but applying them during debugging was another matter. While building the robot model in RViz2, we encountered frame alignment issues, incorrect mesh orientation, and motion that looked numerically correct but visually wrong. Through ROS documentation, GitHub examples, and many trial-and-error sessions, we learned how to inspect topic communication, verify joint states, and debug TF trees step by step. ROS 2 gradually changed from an abstract framework into a practical engineering environment.

The algorithmic side of the project focused on the “digging problem,” where a target bin at the bottom of a stack must be retrieved by relocating the bins above it. At first, we underestimated the problem and thought simple conditional statements would be enough. Instead, the robot needed to identify the target location, detect blocking bins, search for empty neighboring spaces, move obstacles efficiently, and then return for the desired package. We implemented a sequential control logic for this process. Even on a small 3×3 grid, poor decisions created unnecessary movement and wasted time. Better planning noticeably improved retrieval efficiency. This made optimization more concrete for us, since we could directly measure the result of better logic.

A.1 Set of Skills

The project helped us build a broad set of technical and professional skills. In ROS 2, we reached a working to confident level in robot modeling, state publishing, node communication, and visualization. We applied these skills when building the digital twin of the MiniCube robot in RViz2 and validating its planar motion on the grid.

In CAD and mechanical design, we learned to redesign parts not only for appearance but for manufacturability and reliability. Early designs focused mainly on shape and fit. Later versions considered tolerance stacking, assembly order, motor placement, cable routing, and structural rigidity. This shift was important because functional prototypes depend on details that are easy to ignore in early design stages.

In hardware-software interfacing, we gained hands-on experience integrating stepper motors, drivers, Hall effect sensors, power converters, batteries, and onboard computing. We learned how electrical noise, current draw, and poor grounding can affect system stability. Our skill level in this area moved from basic familiarity to working competence through direct troubleshooting.

We also developed stronger debugging skills. Early in the project, debugging often meant changing several variables at once and hoping something improved. Later, our process became structured. We isolated one subsystem at a time, tested inputs and outputs, logged observations, then modified a single variable before retesting. This more disciplined approach reduced wasted time and made failures easier to trace.

System integration became one of the most realistic engineering skills we gained. A robot is not one problem; it is many smaller problems interacting simultaneously. Increasing motor torque could improve motion but create power instability. Reinforcing the chassis could reduce vibration but increase mass. Faster algorithms could demand more accurate localization. We learned to think in trade-offs rather than isolated improvements.

The project also strengthened our soft skills. Team coordination became essential when progress slowed. There were periods where mechanisms jammed, sensors drifted, or software crashed for several days. During those phases, communication mattered as much as technical ability. We learned to divide tasks clearly, share updates honestly, and adjust plans quickly when delays appeared. Time management also improved. We initially underestimated the time required for hardware iteration, printing, rewiring, and calibration. By the second half of the project, our schedules became more realistic because they were based on actual experience rather than optimistic assumptions.

A.2 Future Learning Goals

The project also clarified what we still need to learn. Our first future goal is real-time synchronization between the physical robot and its ROS 2 digital twin. In the current version, simulation and hardware are separate systems. A live telemetry pipeline would allow the virtual model to mirror the robot state in real time, which would improve debugging and monitoring.

Another major goal is advanced path planning for multi-robot operation. The current prototype demonstrates a single robot on the grid, but real warehouse systems depend on multiple agents moving simultaneously. This introduces traffic conflicts, congestion, and deadlock risks. We want to study cooperative planning algorithms and scheduling methods that allow several robots to share the same grid efficiently.

We also want to improve localization through sensor fusion. At present, Hall effect sensors provide position feedback, but relying on one source has limitations. Combining Hall sensors with wheel encoders, inertial sensing, and possibly camera-based detection would make the robot more robust when one sensor becomes unreliable.

Finally, we aim to improve design for maintenance and long-term reliability. Future versions should have cleaner internal wiring, modular actuator mounts, faster battery replacement, and easier access to repair points. These goals come directly from the practical delays we experienced during testing.

A.3 Conclusion

This capstone project changed how we understand engineering work. At the beginning, we viewed mechanics, electronics, and programming as separate tasks. By the end, we understood that successful robotics systems depend on all of them working together. The final robot is important, but the deeper result is the problem-solving approach we developed while building it. We learned to test assumptions early, trace failures systematically, and redesign without becoming attached to first ideas. We learned that progress in robotics is often slow and uneven, with many setbacks before visible results appear. Most importantly, we moved from theoretical understanding toward practical engineering competence. The lessons from this project will remain useful not only for future robotics work, but for any complex system that requires integration, patience, and disciplined problem solving.

Appendix B: Ethical and Legal Issues

Our project focuses on the development of an autonomous cubic rail robot (“MiniCube”) designed for high-density warehouse storage and retrieval systems. While the system is primarily technical, it raises several important ethical and legal considerations.

Operational safety is critical. The robot is designed to autonomously navigate a grid system, lift payloads, and interact with other potential agents. Malfunctions—such as navigation errors, lifting failures, or collisions—could damage goods or pose risks to nearby human workers in real warehouse environments. To mitigate this, we prioritize controlled testing, starting with a small-scale 3×3 grid prototype before scaling to more complex environments. We also incorporate safety features such as limit switches, sensor feedback, and the possibility of human override in the control system.

Liability and accountability are important legal concerns. In autonomous systems, responsibility for errors (e.g., damaged goods or system failures) can be unclear. As developers, we acknowledge the importance of transparent system design and clear documentation to ensure traceability of decisions and failures.

Intellectual property and licensing are relevant. Our project builds upon existing research in robotic storage systems and algorithms such as puzzle-based storage and path planning. We ensure proper citation of all referenced work and avoid unauthorized use of proprietary technologies. Any software or designs we develop are intended for academic use, with the possibility of open-source sharing to support further research.

Finally, data and system security must be considered. In real-world deployment, such systems may handle inventory data, operational metrics, or logistics information. Ensuring that this data is securely stored and accessed only by authorized users is essential to prevent misuse or commercial exploitation.

B.1 Informed Judgments and Trade-offs

The design of MiniCube required multiple engineering trade-offs, many of which have ethical implications.

A key trade-off was between performance and cost. High-precision components and industrial-grade actuators could improve reliability but would significantly increase system cost. We addressed this by using cost-effective components (e.g., NEMA 17 motors instead of more expensive alternatives) while improving reliability through mechanical innovation, such as the sliding wall mechanism for stable motion. This reflects a responsible balance between accessibility and safety.

Another trade-off involved system complexity vs. robustness. For example, solving the “digging problem” (retrieving bins buried under others) could involve highly complex optimization algorithms. Instead, we implemented a simplified heuristic approach that ensures reliable real-time operation, even if it is not globally optimal. This decision prioritizes stability and predictability over theoretical efficiency.

We also considered energy autonomy vs. safety. Transitioning from a tethered system to a battery-powered robot improves flexibility but introduces risks such as power instability. To address this, we separated power supplies to maintain stable operation.

Finally, we maintained academic integrity throughout the project by properly referencing all literature and clearly distinguishing our original contributions from existing research.

B.2 Sustainability and Social Responsibility

The MiniCube project contributes to sustainability primarily through improving space and energy efficiency in logistics systems. High-density storage systems reduce the need for large warehouse footprints, which can lower land use and energy consumption.

Our design also emphasizes modularity and reusability. Components such as motors, structural elements, and lifting mechanisms can be replaced or upgraded without rebuilding the entire system. This reduces electronic and material waste over time.

From a societal perspective, automation in warehouses raises concerns about job displacement. However, systems like MiniCube are better understood as tools that augment human workers, handling repetitive or physically demanding tasks while allowing humans to focus on supervision, maintenance, and decision-making roles. Additionally, our system is being developed with future human-robot collaboration in mind. The design includes the possibility for human intervention and control, aligning with modern approaches to safe collaborative robotics.

B.3 Examples of Ethical Reasoning in Practice

One clear example of ethical reasoning occurred during the redesign of the lifting mechanism. The initial rope-based system caused payload oscillation, leading to instability and potential failure during operation. Instead of accepting this risk, we redesigned the system using a rigid tape-based lift, which eliminated oscillation and ensured reliable payload handling. This decision prioritized safety and reliability over design simplicity.

Another example is the development of the digging algorithm. While more complex optimization methods exist, we chose a simpler and more predictable relocation strategy that minimizes energy use and reduces the likelihood of unexpected behavior. This reflects a commitment to safe and understandable autonomous decision-making.

We also addressed fairness and teamwork ethics within the group. Responsibilities for mechanical design, programming, and documentation were distributed evenly, and all contributions were acknowledged. This ensured a collaborative and respectful working environment.

B.4 Conclusion

Ethical considerations have played a central role in the development of the MiniCube project. From ensuring safe autonomous operation to addressing sustainability, fairness, and legal responsibility, our decisions reflect the core principles of engineering ethics.

As the project evolves toward multi-agent systems and real-world applications, we plan to further formalize safety protocols, improve transparency in algorithmic decision-making, and continue prioritizing responsible innovation.

Our goal is not only to build a functional robotic system, but also to demonstrate that advanced engineering solutions can be developed with integrity, accountability, and respect for society.

References

- K. Azadeh, R. De Koster, and D. Roy. Robotized and automated warehouse systems: Review and recent developments. *Transportation Science*, 53(4):917–945, 2019.
- Y. A. Bozer and J. A. White. Travel-time models for automated storage/retrieval systems. *IIE Transactions*, 16(4):329–338, 1984.
- J. Cai, X. Li, Y. Liang, and S. Ouyang. Collaborative optimization of storage location assignment and path planning in robotic mobile fulfillment systems. *Sustainability*, 13(10):5644, 2021.
- G. Carabin, E. Wehrle, and R. Vidoni. A review on energy-saving optimization methods for robotic and automatic systems. *Robotics*, 6(4):39, 2017.
- W. Chen, R. De Koster, and Y. Gong. Analysis and design of rack-climbing robotic storage and retrieval systems. *Transportation Science*, 56(6):1658–1676, 2022a.
- X. Chen, P. Yang, and Z. Shao. Simulation-based time-efficient and energy-efficient performance analysis of an overhead robotic compact storage and retrieval system. *Simulation Modelling Practice and Theory*, 119:102560, 2022b.
- C. Chi, S. Wu, D. Xia, and Y. Wu. Dynamic picking and storage optimization of robotic picking systems. *Wireless Personal Communications*, 128:1–23, 2022.
- K. R. Gue and B. S. Kim. Puzzle-based storage systems. *Naval Research Logistics*, 54(5):556–567, 2007.
- T. Lamballais, D. Roy, and M. B. M. De Koster. Estimating performance in a robotic mobile fulfillment system. *European Journal of Operational Research*, 256(3):976–990, 2017.
- B. K. Patle, L. G. Babu, A. Pandey, D. R. K. Parhi, and A. Jagadeesh. A review: On path planning strategies for navigation of mobile robot. *Defence Technology*, 15(4):582–606, 2019.
- K. J. Roodbergen and I. F. A. Vis. A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research*, 194(2):343–362, 2009.

- P. Trost, G. Kartnig, and M. Eder. Simulation study of rcs/r-systems with several robots serving one picking station. *FME Transactions*, 51:201–210, 2023.
- C. Wang and Y. Qiu. Vehicle routing problem with stochastic demands and simultaneous delivery and pickup based on the cross-entropy method. In *Advances in Automation and Robotics, Vol. 2*, pages 55–60. Springer, 2011.
- R. Yuan, J. Li, W. Wang, J. Dou, and L. Pan. Storage assignment optimization in robotic mobile fulfillment systems. *Complexity*, 2021:4679739, 2021.
- A. Yue and S. L. Smith. Minimizing robot digging times to retrieve bins in robotic-based compact storage and retrieval systems. *Transportation Science*, 54(6), 2020.
- N. Zaerpour, Y. Yu, and R. De Koster. Small is beautiful: A framework for evaluating and optimizing live-cube compact storage systems. *Transportation Science*, 51(1): 34–51, 2017.
- Z. Zhao, J. Cheng, J. Liang, S. Liu, M. Zhou, and Y. Al-Turki. Order picking optimization in smart warehouses with human–robot collaboration. *IEEE Internet of Things Journal*, 11(9):16314–16325, 2024.
- X. Zhong, J. Tian, H. Hu, and X. Peng. Hybrid path planning based on safe a* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment. *Journal of Intelligent & Robotic Systems*, 99:65–81, 2020.
- B. Zou, R. De Koster, and X. Xu. Operating policies in robotic compact storage and retrieval systems. *Transportation Science*, 52(4):788–811, 2018.