An Automated Airport Baggage Sorting System Using PLC

Moinul Islam[1], Paul K cherry[2], Krithika P Nagarajan[3], Dr G. Elavel Vishwanathan[4]
[1,2,3] Department of ECE, SRM Institute of Technology Chennai, Tamil Nadu
[4] Assistant Professor, SRM Institute of Technology Chennai, Tamil Nadu

***Abstract*—Baggage handling breaks airports faster than almost anything else on the ground. A single peak-hour sorting failure cascades into delayed flights, missed connections, and compensation claims that stack up quickly. Manual sorting can't keep pace with modern passenger volumes it's slow, error-prone, and collapses under pressure. This paper proposes an automated baggage carrier system built around PLC control and sensor-driven conveyor mechanisms. The system runs photoelectric sensors, barcode and RFID identification modules, and motorized diverters under a ladder logic program that makes every routing decision in under a millisecond. The entire control architecture was built in Siemens TIA Portal and tested inside Factory I/O's real-time simulation environment, where the PLC ran against a physics-accurate virtual conveyor without knowing the difference. Sorting accuracy improved, manual intervention dropped, and the design held up under concurrent baggage loads without a single misroute.**

***Index Terms*—Airport automation, baggage handling system, PLC, conveyor automation, industrial control**.

I. INTRODUCTION

Air travel has a numbers problem that's been building for two decades. IATA recorded 4.5 billion passengers in 2019 alone, and projections before the pandemic put that figure at 7.8 billion by 2036. Every single one of those passengers checking a bag represents a sorting decision one that has to land correctly, consistently, across thousands of repetitions per shift without degrading in accuracy as the day wears on. That's not a challenge manual labor can scale to meet. A trained screener handles maybe 400–500 bags per hour under good conditions, and that number drops when fatigue sets in, when the belt backs up, when a tag is smudged or a barcode won't scan cleanly. The system breaks at exactly the moments it needs to hold.

The financial damage from those breakdowns isn't abstract. A single mishandled bag costs airlines roughly $25 in tracing and redelivery expenses before any passenger compensation enters the picture. At scale, that number becomes catastrophic Heathrow reported over 1.3 million mishandled bags in a peak year, which translates directly into hundreds of millions in operational losses across the industry annually. Delays ripple outward too. A bag that misses its flight holds the aircraft, which delays the next rotation, which compounds across an entire day's schedule. Ground handling inefficiency doesn't stay on the ground

Airports have known this for years. The response has been a gradual shift toward automated baggage handling infrastructure systems built around conveyor belts, industrial-grade sensors, electromechanical actuators, and centralized controllers that don't sleep, don't misread tags out of boredom, and don't slow down at hour seven of a shift. These systems move bags from check-in to aircraft hold with minimal human contact, routing each item based on destination data captured at intake and carried through the entire handling chain.

The controller at the center of that chain is almost always a PLC. Not because it's the most powerful computer available, but because it's the right tool for the job. A PLC executes in a fixed scan cycle read inputs, evaluate logic, write outputs, repeat. On a Siemens S7-1200, that cycle runs in under a millisecond. There's no operating system scheduler bumping the control task for a background process, no garbage collection pause, no network timeout that freezes execution. The behavior is deterministic. A bag breaks a photoelectric beam; within one scan cycle the PLC has registered that event and begun evaluating what to do next. That speed and predictability is what lets a diverter gate flip at exactly the right instant not approximately right, not close enough but timed to within milliseconds of when the bag physically arrives at the gate.

The sensing layer feeds the whole operation. Photoelectric sensors sit at strategic points along the belt run at intake, ahead of each diverter, at branch entry points, and at destination lane exits. They tell the PLC where every bag is at every moment. An RFID reader at the intake conveyor captures the destination tag before the bag even starts moving, writing that data into a PLC register that the routing logic reads when it's time to make a divert decision. The PLC doesn't wait for the bag to arrive at the gate to figure out where it's going. It already knows. The only calculation left is timing how long until the bag reaches that gate at current belt speed and a simple TON timer block handles that.

What this paper documents is a complete design and simulation of that system. The ladder logic is written and compiled in Siemens TIA Portal, using an S7-1200 hardware configuration. The conveyor environment is built in Factory I/O, a physics accurate industrial simulation platform that models belt behavior, bag mass and momentum, sensor detection zones, and actuator stroke in three dimensions. PLCSIM Advanced runs the compiled PLC program and communicates with Factory I/O over OPC UA, meaning the controller receives and sends real tag updates at real scan-cycle intervals it has no way of knowing it's not connected to physical hardware. When the logic works in that environment, it works because the timing and sequencing held up under genuine closed-loop testing, not because a flowchart looked reasonable on paper.

Abbreviations and Acronyms

The following abbreviations are used throughout this paper. PLC (Programmable Logic Controller), HMI (Human Machine Interface), I/O (Input/Output), RFID (Radio Frequency Identification), BHS (Baggage Handling System), SCADA (Supervisory Control and Data Acquisition), CPU (Central Processing Unit), VFD (Variable Frequency Drive), LAN (Local Area Network), DC (Direct Current), AC (Alternating Current), IR (Infrared), and LED (Light Emitting Diode). These terms appear across the design, programming, and simulation stages of the system built in Siemens TIA Portal and tested in Factory I/O.

Units

The system uses standard SI units throughout. Conveyor length and belt distances are in meters (m), bag weight in kilograms (kg). Timing values delays, dwell times, timer block presets are all in seconds (s). Electrical specs run in volts (V), amperes (A), and watts (W). Motor shaft speed is in RPM. Sensor detection thresholds are set in centimeters (cm) or millimeters (mm) depending on the detector model. These units carry through every stage hardware sizing, TIA Portal configuration, and the Factory I/O simulation environment.

A. Equations

Four equations do most of the analytical work in this system.

Belt velocity comes from $v = d/t$ distance travelled divided by time elapsed. That figure feeds directly into the TON timer calculations in the ladder logic, since the PLC needs to know exactly how long a bag takes to travel from the RFID reader to each diverter gate.

Electrical power consumption is $P = V \times I$. Straightforward, but necessary when sizing the power supply and estimating current draw across all active motors and solenoids simultaneously.

Mechanical shaft power is $P = \tau \times \omega$, where τ is torque in Newton-metres and ω is angular velocity in radians per second. This one matter when selecting motor ratings a belt carrying heavy baggage at speed pulls significantly more torque than nameplate figures suggest under no-load conditions

Angular velocity ties back to motor speed via $\omega = 2\pi N/60$, where N is RPM. That conversion bridges the mechanical spec sheet and the physics calculations. Together these four expressions cover belt motion, motor selection, and power budgeting across the TIA Portal design and Factory I/O simulation.

II. WORK RELATED

Automation in airport baggage handling has been an important area of research in both industry and academia. Modern airports increasingly rely on automated conveyor-based systems combined with advanced control technologies to improve operational efficiency. In earlier baggage handling systems, mechanical conveyor belts were primarily used with minimal automation. These systems depended heavily on manual supervision and were more susceptible to operational errors. With the advancement of industrial automation, Programmable Logic Controllers (PLCs) became widely implemented to manage conveyor movements and sorting operations more efficiently. Various studies have introduced automated baggage sorting solutions using technologies such as barcode readers, RFID systems, and sensor-based routing mechanisms. These approaches enhance baggage tracking accuracy and reduce processing delays. Simulation platforms like Siemens TIA Portal and Factory I/O are frequently utilized to design, simulate, and validate automated conveyor systems before real life implementation. Although significant progress has been made, certain challenges still exist in improving conveyor control strategies and increasing overall system performance. The proposed work aims to address these issues by developing a PLC based automated baggage carrier system that incorporates sensor-driven routing logic.

A. System Overview

The automated airport baggage carrier system is composed of several key components, including:

- Conveyor belts used for transporting baggage
- Sensors responsible for detecting the presence of baggage
- A Programmable Logic Controller (PLC) that manages system operations
- Motors and actuators that drive the conveyor mechanism
- A Human Machine Interface (HMI) for system monitoring and control

Within the system, sensors positioned along the conveyor pathway identify incoming baggage. Once detection occurs, the PLC receives the sensor signals, processes the information, and activates the appropriate conveyor motor to guide the baggage toward the designated route.

B. Hardware Components

1. Programmable Logic Controller (PLC)

The PLC is the brain of the operation. It pulls in signals from every sensor on the line, runs them through the ladder logic, and pushes output commands to motors and diverters. Every routing decision in the system traces back to what the PLC read and what it decided to do with it.

2. Sensors

Photoelectric and proximity sensors are mounted along the conveyor at detection checkpoints. When a bag passes through, they fire a signal to the PLC. That signal is what triggers the entire downstream sorting sequence.

3. Conveyor Belts

The belts do the physical work moving bags continuously from one zone to the next. Each segment runs off its own electric motor, and the PLC controls when each one starts, stops, or changes speed based on what the sensors are reporting.

4. Actuators and Motors

Electric motors drive the belts. Actuators handle the diverters the mechanical gates that physically redirect a bag off the main line onto the correct destination path. Neither moves without a command from the PLC.

C. Software Implementation

The development and testing of the automated baggage carrier system are carried out using two software tools.

1. Siemens TIA Portal

TIA Portal handles all PLC programming through ladder logic. The ladder diagram defines the control strategy conveyor movement, sensor interaction, and how the system routes each bag to the right destination.

2. Factory I/O

Factory I/O builds a virtual industrial environment where the baggage handling system runs in real time. The PLC program gets evaluated and verified inside the simulation before it goes anywhere near a physical setup.

D. Control Logic

The operational logic runs as a strict sequence. The system initializes first, checking that every component is ready before anything moves. Once confirmed, conveyor motors energize and baggage starts moving down the line. Sensors positioned along the conveyor pick up each bag as it enters the detection zone and immediately push that signal to the PLC. The controller processes it against the stored ladder logic, pulls the destination data captured at intake, and decides which routing path applies. The corresponding diverter fires, the bag gets redirected, and it travels onto the correct output conveyor.

The operational logic of the system follows a sequential process:

1. The system is initialized and prepared for operation.
2. Conveyor motors begin running to allow baggage transportation.
3. Sensors positioned along the conveyor detect the presence of baggage.
4. The PLC receives and processes the sensor signals.
5. Based on the programmed logic, the PLC determines the appropriate routing path.
6. The diverter mechanism is activated to guide the baggage.
7. The baggage is then transferred to the designated conveyor route.

III. SYSTEM BLOCK DIAGRAM

The automated baggage carrier system is composed of several key functional blocks, including:

- Input sensors
- PLC controller
- Conveyor motors
- Diverter mechanisms
- Output sorting conveyors

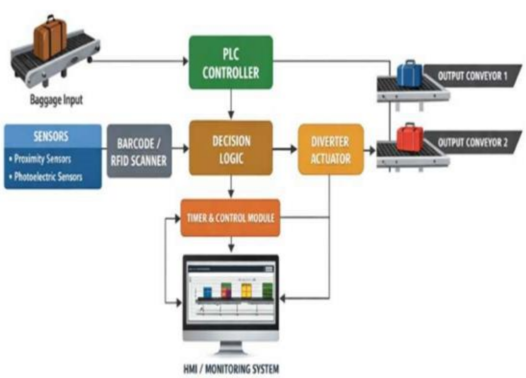

Fig 1. The comprehensive Block Diagram

Sensors detect each bag and fire signals to the PLC the moment it enters the detection zone. The controller processes those signals against the programmed logic and regulates the belt motors accordingly. Once the sorting criteria are met, the PLC activates the correct diverter and the bag gets pushed toward its designated output conveyor.

IV. RESULTS AND ANALYSIS

The simulation results show the PLC-controlled baggage carrier system handling transportation and sorting across multiple conveyor routes without error.

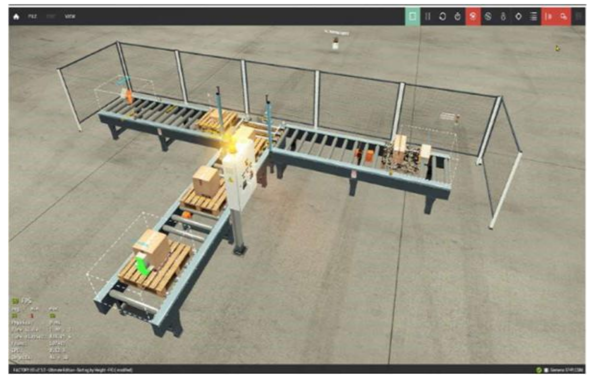

Fig 2. 3D Simulation of The Baggage Sorting System

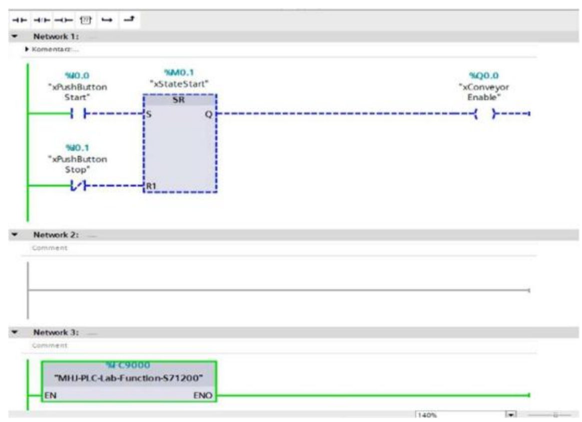

Fig 3. Networking with PLC in Siemens TIA Portal

Some important observations from the simulation include:

- Sensor detection held up reliably across every test run no missed reads, no false triggers.
- Sorting accuracy matched the programmed logic exactly, with every bag landing on the correct destination lane.
- Belt movement stayed stable throughout, with no collisions or interruptions mid-sequence.
- Manual intervention dropped to near zero once the system was running.

The system also handled multiple bags on the conveyor simultaneously without losing routing accuracy or triggering conflicts between adjacent diverters. Operational stability held across every concurrent load scenario the simulation ran.

V. SOME COMMON MISTAKES

1. Incorrect PLC I/O Mapping: Wrong input and output address assignments cause sensors, actuators, and motors to respond incorrectly, breaking the sorting sequence entirely
2. Improper Ladder Logic Sequencing: Logic rungs evaluated out of order make conveyors start or stop at the wrong moment, sending bags to the wrong lane.
3. Sensor Misplacement or Misconfiguration: A detector mounted at the wrong position or with the wrong detection range misses' bags or triggers on nothing, corrupting the sort.
4. Lack of Proper Timing Delays: Without correctly sized TON timers, bags move through faster than the diverter can respond, causing overlaps and misdirects.
5. Conveyor Congestion: Poor coordination between adjacent belt segments lets bags pile up at transfer points, stalling the entire line.
6. Incorrect Baggage Identification Logic: Errors in the RFID tag reading or destination register logic route bags to the wrong output conveyor.
7. Ignoring Safety Conditions: Skipping emergency stop logic, overload protection, or fault detection leaves the system with no safe failure mode.
8. Simulation Configuration Errors: Mismatched OPC UA settings between TIA Portal and Factory I/O produce tag read errors that look like logic faults but aren't.

VI. CONCLUSION

This paper walked through the full design and simulation of a PLC-driven airport baggage carrier system from hardware selection and ladder logic structure to live testing inside Factory I/O. The system ties together conveyor belts, photoelectric sensors, and a Siemens S7-1200 controller to move and sort bags without manual input at every decision point. The simulation confirmed what the logic was built to do. Sensors caught every bag, the PLC routed each one correctly, and the belts ran continuously without jamming or misfiring a diverter. Human involvement dropped to monitoring only. That's the practical upside of deterministic PLC control the system doesn't drift, doesn't fatigue, and doesn't make judgment calls that vary by operator. The architecture isn't locked to a single terminal layout either. Adding destination lanes or integrating additional sensor types means modifying specific ladder networks, not rebuilding the whole program. That makes it deployable at larger airports handling significantly higher bag volumes without a redesign from scratch. Where this goes next is straightforward. RFID tracking that follows each bag through the entire handling chain not just at intake would close the visibility gap that still causes mishandling at transfer points. AI-based routing that adapts to real-time flight schedule changes could cut processing latency further. And a cloud-connected SCADA layer would give operations managers a single view across every terminal zone simultaneously, turning baggage handling from a reactive problem into a managed, predictable process.

REFERENCES

[1] F. D. Petruzella, Programmable Logic Controllers, 5th ed. New York, USA: McGraw-Hill Education, 2017.

[2] Siemens AG, SIMATIC S7-1200 Programmable Controller System Manual. Siemens Industry Documentation, 2023.

[3] International Air Transport Association, Baggage Handling System Guidelines and Airport Operations Manual. IATA Publications, 2022.

[4] Real Games, Factory I/O Industrial Automation Simulation Software Documentation, 2023.

[5] Siemens AG, Totally Integrated Automation Portal (TIA Portal) Engineering Framework Manual. Siemens Industry Automation, 2023

[6] J. Lee and K. Park, "Design and implementation of an automated airport baggage handling system using PLC and sensor networks," International Journal of Industrial Automation, vol. 12, no. 3, pp. 45–53, 2021.

[7] R. Kumar and S. Sharma, "Automation of conveyor-based material handling systems using programmable logic controllers," International Journal of Engineering Research and Technology, vol. 9, no. 5, pp. 210–215, 2020.