Lessons Learned from Large-Scale Mobile Robot Deployments in Warehouse Environments

Sarah Elliott, Charles Raimbert, Derek King, Steffen Fuchs and Brendan Clune
Fetch Robotics \ Zebra Technologies
Email: sarah.elliott@zebra.com

I. Deployment Summary

We deployed fleets of mobile robots to warehouse / fulfillment center environments. In one case, we deployed a fleet of more than 30 robots to a warehouse to service an area consisting of over 100,000 sq. feet (divided into 9 ft x 200 ft warehouse aisles).

The robots helped to fulfill e-commerce and wholesale orders. Human warehouse workers (pickers) picked items off warehouse shelves and loaded them into totes on the robots.

Our system would receive information about the orders, each order consisting of one or more items to be picked from known locations in the warehouse. The order information came in a "wave" of multiple orders from a Warehouse Management / Execution System (WMS or WES), for example Manhattan Scale. The wave would be in XML format and generally consisted of anywhere from approximately 100 orders to over 1000 orders. Multiple waves of orders would be fulfilled in a single day, with throughput ranging from about 3000-7000 total orders fulfilled per day. That equates to about 5000 - 14000 units (individual items) picked per day.

II. Order Fulfillment As An Optimization Problem

We were tasked with not only the usual robot navigation and path planning type operations, but also optimizing the order fulfillment process as a whole. Our task was to create a system that, given a wave of orders, would decide which items should be picked onto which robots and at what times. There were many parameters over which to optimize in this problem:

- Optimizing path length / travel time for humans and robots
- Minimizing idle time for the human picker and the robots
- Cubing: Ensuring the appropriate volume of items is assigned to a robot by:
  - Modeling the "foldability" of items
  - Considering how densely items can be packed into totes
- Managing orders with differing priorities / shipment times

This was a scheduling problem, a path optimization problem and a packing problem all in one.

III. Lessons Learned

When deploying our system (including the 30+ robots), we faced the usual robotics problems, for example localization in dynamic environments (with long aisles that all look the same) and navigation in narrow, high traffic areas. However, in this talk we wanted to focus on some of the other, more unexpected, challenges that we faced in our deployments.

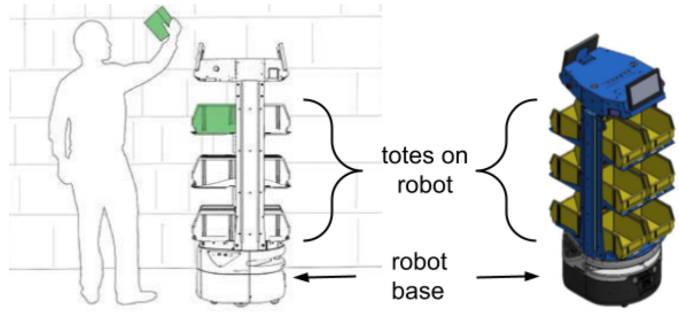


Fig. 1. Human picker picks items into totes on robot

A. Dealing with Bad Input Data

Like many real world applications, our system had to deal with a lot of missing, malformed or incorrect input data. An important lesson was figuring out which of these issues to address from within our system, and which to set aside and let the human operators solve manually.

The warehouse aisles were divided into bays, which were further divided vertically into shelves called "levels". Each level was further divided into slots. The location of an item in the warehouse was referred to by a series of numbers for each of its aisle, bay, level and slot. Often when we received the XML for a wave of orders, the location information for an order was formatted inconsistently. For example, the location data for an item at Aisle 100, Bay 2, Level 3, Slot 6 might be in any of the following formats (or others):

- 100-2-3-6
- 100236
- 100020306

Because this was such a common problem, but with a somewhat limited scope we decided to address most of the location inconsistencies within our system using complex logic for parsing the location field of the XML. This was possible because there are a finite number of possible locations in the warehouse and some logical assumptions could be used to disambiguate location information.

However, for non-location information that was missing or malformed, we quickly realized that it was not worth

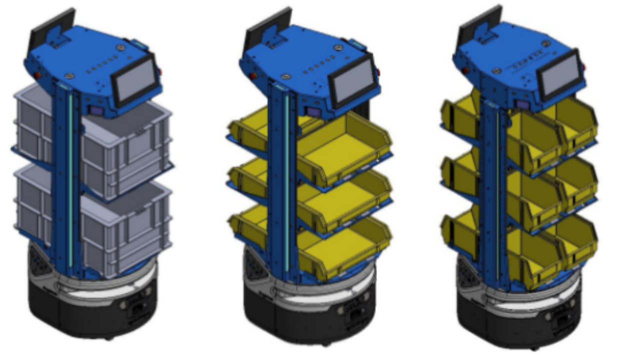
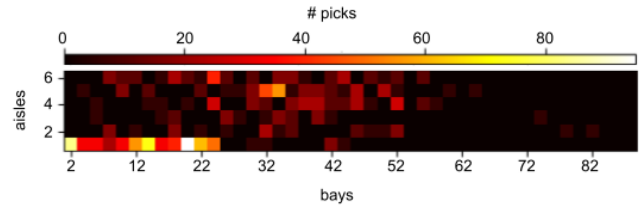
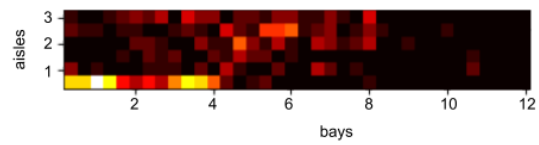


Fig. 2. Examples of possible configurations of shelves and totes on robots



(a) Heatmap of Picks from Customer Warehouse



(b) Heatmap of Picks in Test Facility

Fig. 3. Fig. (a) shows the distribution of item locations for a real wave of approximately 1800 items picked. We learned the characteristics of the customer's waves and generated test data shown in Fig. (b) for our test facility. The test data was representative of the original wave even though our test facility was much smaller than the customer warehouse. Warehouses are divided into aisles, which are further sub-divided into bays. In the case of the test facility data, the left and right sides of each aisle are plotted separately.

trying to capture every edge case with our code. Problems with non-location information were much rarer and harder to disambiguate because they generally came in the form of missing data like a missing stock keeping unit (SKU). In that case, we flagged those orders for manual fulfillment by humans with carts instead of robots. Understanding the limitations of an automated system and the best use of the humans' and robots' time was key to our success.

B. System Introspection

It was important for us to be able to introspect our system in real time during operation. This helped us to establish the customer's trust in the new technology. Customers would make multiple inquiries per day as to the status of specific orders and we could tell them which robot was fulfilling that order and the location of that robot in the warehouse. When we experienced problems with the system, showing the customer that we still knew the state of the orders despite any issues helped us to maintain trust. The type of inquiries made by customers also helped inform the design of customer-facing tools such as system dashboards moving forward.

C. Hardware Flexibility & Inflexibility

To a certain degree in robotics, once the robots are in the field, it can be hard to make changes to hardware. However, we did strive to make the robot configurable in a few important ways. As mentioned previously, human workers fulfill orders by picking items off shelves into totes on the robot. Fig. 2 shows how the robots can support variable sizes, shapes and numbers of totes. Totes can be placed on adjustable shelves or the shelves can be removed and totes can be stacked on one another.

Different sizes of totes were important because if an order contains multiple items, for example someone orders a pair of shoes and a pair of socks from the same retailer, warehouses will often place all items for the same order in the same box or tote during picking. This is done rather than putting all items in the same large tote to simplify the packing step and prevent workers from having to "demultiplex" the items into their individual orders before packing. By contrast, all orders for single items, so every order for only one pair of socks, are often picked into one big tote since each of those items are their own individual order at packing anyway.

D. Unpredictable Human Pickers

Initially we did not explicitly direct the pickers to a specific location or robot. The robots would navigate to a location where an item was to be picked and indicate the item and its location via a screen. A picker would see the robot and pick the item. It was difficult to predict which robot a picker would see and choose to pick from next. We started with certain assumptions about picker behavior and discovered that small changes in that behavior greatly impacted system performance.

In more recent deployments, we can now equip the pickers with handheld or heads-up displays that direct them to specific locations. This makes the pickers more predictable and can increase efficiency but does come with the added cost of extra devices. In the next section, we detail how an inaccurate assumption led us to create an unrealistic simulation and how we resolved that issue.

E. Simulation & Testing

1) Simulator Design: One of the most important takeaways from the deployment was the role of robust simulation and testing in the development of real world robot solutions. We developed our own simulator and in the process learned a lot about the requirements and limitations of a simulator for such a project.

With our simulator, we simulated the movements of robots, pickers and inventory through the warehouse. It was important that the input and output to the simulator be identical to that from real world operations. This enabled running real world XML waves through the simulator and the output was logs detailing the operations of the system and XML feedback to the WMS / WES. By comparing the real and simulated output for the same input, we made the simulator more realistic.

2) Modeling Pickers in Simulation: It was particularly challenging to model the picker behavior in simulation, because of their unpredictable behavior as mentioned above. To improve our simulated pickers, we compared their behavior in simulation to the real world behavior for similar input, reviewed recordings of pickers working and interviewed them about their work. We then adjusted the simulation accordingly.

In one case, we had assumed that pickers would move to the next closest robot in their line of sight when deciding where to go next. However, in reality, pickers would gravitate towards clusters of multiple robots so that they could save time by achieving a high density of picks. They would even bypass single robots to reach denser clusters. Our incorrect assumptions about picker behavior caused us to initially program unrealistic simulated pickers. We realized our mistake when comparing the simulated output to the real world output and by observing pickers in the real warehouse.

As mentioned previously, small changes in picker behavior had a huge impact on the performance of our algorithms. It was important both to make simulated pickers as realistic as possible and to try not to overly tailor our algorithms for the picker behavior in simulation, since real world picker behavior is so variable beyond what we could capture in simulation.

3) Modeling Customer Data: Before deploying robots to the customer site, we also obtained multiple weeks of real XML waves from the customer and analyzed those and ran them through our simulator. This helped us test our interface for consuming the XML from the WMS / WES and also allowed us to build a profile of the characteristics of past waves. Fig. 3 shows how we used what we learned from the customer data to create synthetic data for our test facility that was representative of the distribution of items picked at a customer site. In addition to the locations of items picked, we also replicated the size and weight distribution of the orders. Our data generation pipeline also allowed us to generate atypical data to test corner cases.

4) The Test Suite: The simulator was used to iterate on and improve our order allocation algorithm. To maintain stability and prevent regressions, we also developed unit and integration tests for individual components and for the system as a whole. The simulator and automated tests were very useful for rapidly iterating on algorithmic changes. In addition, we created a scaled-down version of the customer warehouse at our own facility for further testing and validation. This scaled-down warehouse was extremely valuable in testing out the human-robot interaction components of the system and for doing end-to-end testing of the whole system.