An Open-Source Framework for Last Mile Delivery with Heterogeneous Robots

Asha Kailin Jain,[1] Maxwell Svetlik,[2] Nicholas Machak,[2] Kavan Singh Sikand,[3] Cem Karamanli,[4] Kaiyu Zhou,[5] Justin Hart,[3] Joydeep Biswas,[3] Luis Sentis,[1] Junfeng Jiao[5]

[1]Dept. of Aerospace Eng., [2]Dept. of Mechanical Eng., [3]Dept. of Computer Science, [4]Dept. of Operations Research and Industrial Eng., [5]School of Architecture.
The University of Texas at Austin, Austin, Texas, USA
hart@cs.utexas.edu, joydeepb@cs.utexas.edu, lsentis@utexas.edu, jjiao@austin.utexas.edu

Abstract

The Short to Medium Range Autonomous Delivery System (SMADS) is a case study on developing a software stack that allows customers to request deliveries via autonomous robots on the University of Texas at Austin campus. The SMADS stack integrates several subsystems, including a robot autonomy module, a delivery scheduler, a robot interface, and a customer-facing app. In this article, we describe the steps taken to create the SMADS software stack and the field results of delivering bottled lemonade on outdoor routes to buildings on the UT Austin campus, representing advances in 1) integrating end-users and robot autonomy into an overall open-source architecture, and 2) minimizing the interface between the robot platform and the greater software stack. In total, the SMADS system completed 26 lemonade delivery trips to customers over 26 hours during the course of 5 days on two robot platforms. This paper analyzes the challenges of deploying autonomous robots in outdoor environments, lists our solutions to overcome some of these issues and propose future work.

1 Introduction

In light of the COVID-19 pandemic, there is considerable interest in reducing human contact in last-mile delivery problems (Pani et al. 2020), such as food and grocery deliveries. In this paper, we present the Short to Medium Range Autonomous Delivery System (SMADS) which employs a fleet of heterogeneous autonomous mobile robots to solve the last-mile delivery problem. SMADS provides a complete last-mile commerce solution, integrating a custom mobile application interface to take user orders, a task scheduler to assign deliveries to robots, an execution monitoring system to track and communicates delivery progress to the customer, and a robust exception handling interface to ensure that the system remains online despite unexpected delivery issues and intermittent internet connectivity. An overview of the SMADS components can be seen in Figure 1.

There are seven key challenges that robotic solutions to the last-mile commerce delivery problem must address: 1) sensitive information must be protected and the system should be restricted to authenticated users; 2) the system must efficiently distribute deliveries to robots; 3) the system must facilitate communication between the customer and robots; 4) the system must enable robots to autonomously navigate to outdoor destinations; 5) the system must support manual intervention to interrupt unsafe operation; 6) the system must handle unexpected delivery issues; 7) and finally, these tasks must be operable on a heterogeneous robot fleet.

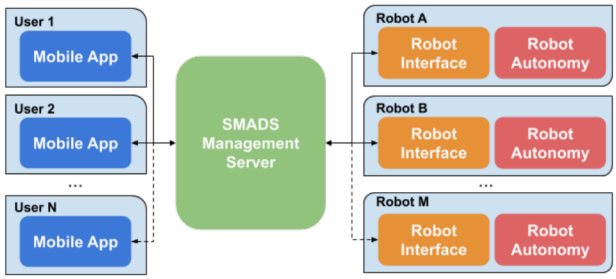

Figure 1: SMADS Software Stack

The architectural design of SMADS is intended to overcome all these challenges. While there exist open-source software solutions that address parts of the overall problem (e.g. Robot Web Tools for connecting robots to the web, and Google OAuth for user authentication), there is no open-source solution to a complete robot-based last-mile delivery problem. SMADS fills this void in the open software community, enabling future research on real-world robotics delivery solutions, and provides results from a case study in deploying SMADS on the campus of the University of Texas at Austin. Code developed for this system can be found on Github. [1]

We deployed our SMADS system for a week-long test from Novemeber 16 - 20th, 2020, totaling 26 lemonade delivery trips over 5 days in 26 active hours with 13 trips requiring human intervention. The challenges encountered during this deployment are reviewed in detail in the Field Results section, motivating future work in areas including localization, social navigation and domain adaptation.

In summary, this paper presents the development and field results of the SMADS software stack which enables multiple robots to autonomously deliver lemonades to customers on the University of Texas at Austin campus. Furthermore, we analyze the challenges involved in deploying autonomous

[1]https://github.com/UTSMADS/aaai_sss21

systems in outdoor environments with current state of the art technologies and identify areas of future work.

2 Related Work

The use of autonomous machines to solve logistics problems has been a focus in the commercial sector in recent years. Autonomous machines are now involved in nearly every stage of the supply chain: in coordinated inventory control in a warehouse (Enright and Wurman 2011), autonomous delivery of inventory via trucks (Boysen, Schwerdfeger, and Weidinger 2018) or tracking and restocking inventory inside a store (Ehrenberg, Floerkemeier, and Sarma 2007; Zhang et al. 2016). These ideas are now a commercial reality through companies like Amazon Robotics for warehouse logistics, Daimler, Waymo and Starsky for autonomous trucking and Bossa Nova Robotics for restocking inventory. While these primarily focus on business-to-business interactions, there is a growing push for autonomous robots in the space of business-to-customer interaction, particularly last-mile delivery which presents unique challenges; a robot navigating in a warehouse works in similar conditions and constraints at all times, while a robot performing last mile delivery will encounter new conditions each time a delivery is made.

While warehouses can be engineered to provide infrastructure to aid robots in localization and navigation (Chuan et al. 2007; Motroni et al. 2018; Wurman, D'Andrea, and Mountz 2008) here we're interested in less structured environments. There has been success in the area of indoor environments in deployments of the CoBot (Veloso et al. 2015a) and BWIbot (Khandelwal et al. 2017) which have been used to reliably localize, navigate and plan in office settings. RoboCup@Home (Wisspeintner et al. 2009; Shah et al. 2019; Jiang et al. 2018) pushes progress in this area by challenging teams to work on multifaceted problems that require a robot to use different approaches simultaneously to successfully complete a task in a domestic setting. We build on this work regarding indoor mobile robot autonomy and apply it to an outdoor setting for autonomous deliveries within short ranges. Though the robots in this study did not differentiate between humans and other worldly obstacles, there is a growing body of work that shows the need for social navigation (Ferrer, Garrell, and Sanfeliu 2013).

Robots with mobile platforms have been deployed to specifically address navigation in urban environments in crowded spaces (Bauer et al. 2009; Kümmerle et al. 2015). While these robots have been made to perform tasks using one particular platform and sensor suite, our work introduces a general framework that is compatible with heterogeneous robot platforms and provides a method for requesting the robot through a mobile application.

3 User Experience During Last Mile Delivery

Below, the SMADS user experience is described, including the procedure for placing an order, tracking a delivery and confirming the customer received the order.

Every order starts with the customer navigating through the ordering sequence in the Texas Botler app, as shown in Figure 2. This interface allows the customer to choose from the available delivery locations with current information on estimated delivery times and the ability to cancel. A customer can decide to cancel an order until the robot leaves the robot depot to service the delivery.

Once a robot begins moving to deliver an order, the robot's route is displayed on a map along with the robot's current position, and the pick-up and drop-off locations. The robot's location is updated every second. This feedback to the customer provides an interactive and informative user experience, reducing the uncertainty in the delivery process. When the robot determines it has arrived to the correct drop-off location, the Texas Botler app moves to the Confirm Pickup screen where the customer confirms they have retrieved their order. The Texas Botler app then displays a pop-up alert confirming the process is complete and initiates commands to send the robot back to the lemonade stand.

4 System Details

As seen in Figure 3, SMADS is comprised of two mobile applications (one for customers and another for the SMADS managers), the SMADS Management Server, the Robot Interface and the Robot Autonomy software. The mobile applications provide customers and SMADS managers a way to interact with the SMADS system, allowing customers to order and managers to supervise autonomous robots. The SMADS Management Server responds to customer and manager requests, handles communication with robots and manages the delivery schedule. On each robot, the Robot Interface translates SMADS Management Server requests to ROS-level commands and the Robot Autonomy stack handles localization and navigation tasks. Below, we discuss each module and its specific responsibilities.

4.1 Mobile App

The Texas Botler app, responsible for allowing customers to place orders, cancel deliveries, and track order progress, was developed for iOS devices. The Texas Botler app communicates with the SMADS Management Server to retrieve data and send customer requests. Details on this communication are discussed below.

App to SMADS Management Server Communication The Texas Botler app runs locally on a customer's phone. When connected to the Internet, the app communicates securely with the SMADS Management Server over HTTPS using REST API calls. Several endpoints exist to communicate information regarding the customer's order history, account information and customer feedback.

For each non-login request, the Texas Botler app sends its authentication token in the JSON payload header. If no token exists or if the token is invalid, the Texas Botler app prompts the user to log in. User authentication is achieved through Google's OAuth framework and Json Web Tokens.

Communication During the Delivery Process Once the customer's order is en-route, the Texas Botler app polls the server to obtain status updates on the robot's current location to update the displayed map. We choose to implement

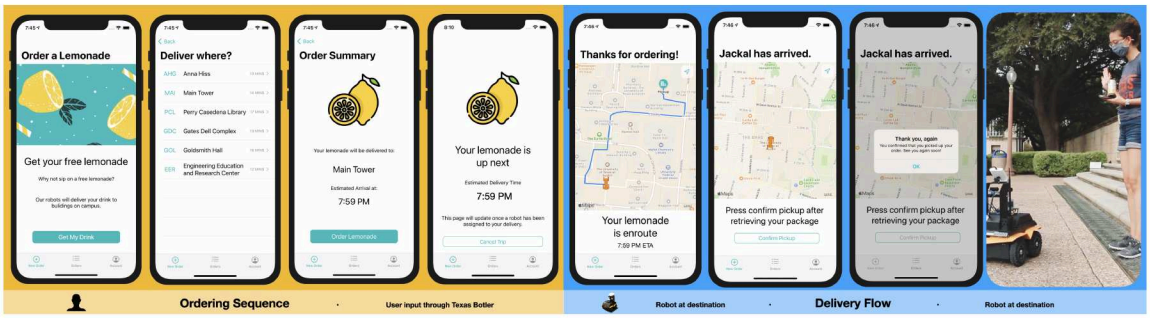

Figure 2: SMADS User Experience in the Texas Botler App. From left to right: Ordering Lemonade, Deliver Where, Order Summary, Queued Trip, Active Trip, Confirm Pickup, and Confirmation Alert screens.

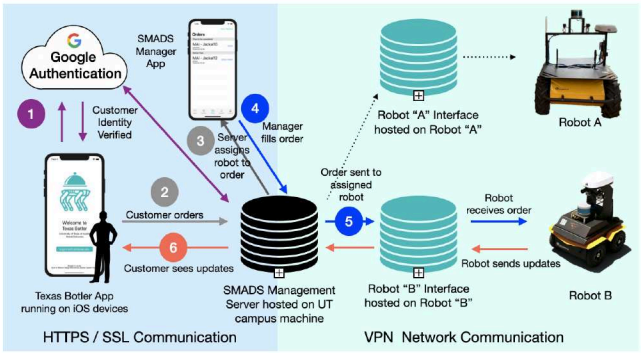

Figure 3: SMADS communication architecture

server polling to improve the robustness of the app-server communication as testing an earlier implementation with WebSocket communication proved to be unreliable. Often, WebSocket messages were lost as the robot traversed outdoor Wifi networks with known dead zones.

4.2 SMADS Management Server

The SMADS Management Server, responsible for handling customer orders and relaying robot information to clients, is hosted on a UT campus-based machine which has an open port to the wider Internet and is trusted on the private VPN robot network.

Handling Order Requests Upon receiving an order request, the SMADS Management Server determines whether there is an available robot that can immediately be assigned to the order or if the order will need to wait in a queue. Trips leave the queue once a robot becomes available, which occurs after a trip is complete or if a previously disconnected robot becomes online. With an available robot assigned to the order, the trip status is updated to "processing" and the robot managers are prompted to fulfill the order.

Manager Fulfillment The SMADS Management Server uses secure WebSocket communication to notify the manager in the iOS Manager app that a trip has been assigned and is awaiting fulfilment. In our lemonade stand scenario, this communication prompts a SMADS manager to load a lemonade can onto the assigned robot and confirm fulfilment. Once the manager fulfills the order and presses a "send" button in the manager iOS app, the SMADS Management Server changes the status of the trip to "en-route" prompting the Texas Botler app to display an interactive map view that shows the customer the robot's delivery route and progress. The SMADS Management Server communicates with the robot via its robot server to obtain the robot's planned path.

4.3 SMADS Management Server to Robot Communication

To stabilize each robot's IP address, we use a private VPN which has similar infrastructure requirements as a Dynamic DNS server but gives the added security of encrypted traffic. Static IP addresses allow the SMADS Management Server to route requests to the appropriate robot. This method is reliable as long as the robot is connected to the internet.

Robot State Error Handling Online robots send updated status requests to the SMADS Management Server every second. However, these updates can fail when the robot loses its internet connection or unexpectedly shuts down. If the SMADS Management Server does not receive a status update at least every three seconds, the corresponding robot is placed in a "reconnecting" state. The SMADS Management Server waits a minute to see if the robot successfully reconnects. If the robot reconnects, the robot server sends a new status update that overrides the "reconnecting" state. Otherwise, the server changes the robot's state to "offline" and notifies all robot managers. Managers are then tasked to intervene and recover the robot. The SMADS Management Server will not assign a delivery to an offline robot until it is back online.

4.4 Robot Interface

Figure 4 shows how the SMADS Management Server interacts with each robot through the Robot Interface. Robot navigation requests are received by the Robot Server, which makes the commands available over ROS. These commands are then interpreted by the Message Translator, converting it to the particular message type expected by the robot's autonomy systems. Finally, this converted message is given to the navigation system by the Autonomy Interface. The Robot Interface imposes no restrictions on the implementation details of the Robot Autonomy module employed by a robot, allowing the SMADS framework to be incorporated onto heterogeneous robot systems.

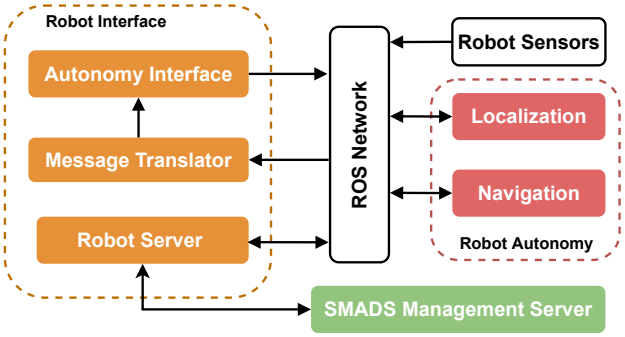

Figure 4: The components of the Robot Interface and how they interact with the existing systems of the robot.

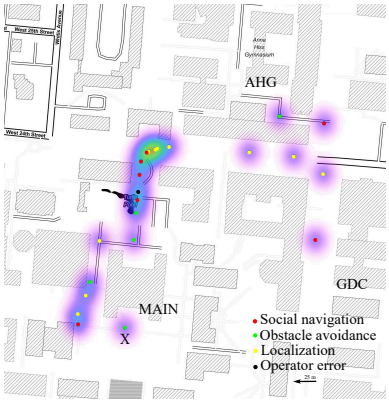

Figure 5: Manual Intervention Locations

Autonomy Interface The Autonomy Interface interacts with the robot's autonomy and is responsible for: 1) forwarding navigation goals from the system to the robot's navigation module, and 2) reading pose estimates from the robot's localization and relaying that to the Robot Server. This interaction with Robot Autonomy is facilitated by the ROS topics that the autonomy modules have exposed. We do not enforce a global coordinate frame for the robot to use during localization or navigation tasks. Instead, we convert between the robot's local frame and a common GPS frame in this Autonomy Interface.

Robot Server The Robot Server is a stateless communication manager that uses the `ROSNodeJS` library to interact with the robot's local ROS core while securely communicating with the SMADS Management Server via HTTPS. It receives and sends JSON to the SMADS Management Server and uses ROS messages to communicate to other components of the Robot Interface.

Message Translator Designing a system to control heterogeneous ROS-enabled robots has a fundamental issue: each robot may employ unique message types to interact with autonomy modules. If a new robot needs to be supported by the system, accounting for any unique messages used requires knowing their structure at compile-time, and will impose these new messages as a dependency for all robots that are already supported.

We solve this problem by eliminating the need to compile against message types explicitly. Instead, the `ros_type_introspection`[2] library is used to reason about the message type at run-time, and then partially recreate the message structure. This task is the function of the Message Translator.

For each robot deployed, the names of the ROS messages used by its autonomy stack are specified to the Message Translator as strings. Since the robot is already using these messages for autonomy, it removes any need to resolve dependencies locally. When the Message Translator receives messages from the robot's autonomy stack, it processes them as a data buffer which is used to populate the required message locally. When messages are received by the Robot Server, e.g. to send the robot to a new location, then this process of message conversion happens in reverse.

[2]http://wiki.ros.org/ros_type_introspection

4.5 Robot Autonomy

The Robot Autonomy module is responsible for two primary tasks. First, it must maintain an estimate of the robot platform's current location, a task referred to as localization. Given the high-traffic, constantly evolving nature of the college campus environment in the SMADS project, we chose to use Episodic Non-Markov Localization which reasons about observations of permanent, temporary, or moving objects (Biswas and Veloso 2017). This localization information is used in the robot's navigation algorithm and in status updates sent to the Texas Botler app. Secondly, the Robot Autonomy module is responsible for issuing low-level commands to the robot platform in order to navigate to a given destination, a task known as navigation. To accomplish navigation, the Robot Autonomy module uses a technique called graph navigation, in which the robot first constructs global navigation graph, and then performs A-Star search over the graph to find the path of minimal distance to reach the destination. This global planner then sends intermediate waypoints to a local planner, which performs obstacle avoidance based on the robot's local observations while navigating towards these waypoints. This navigation stack has been previously shown to perform well in long-term deployments of autonomous robots (Veloso et al. 2015b).

5 Field Results

In order to test and validate the efficacy of the SMADS stack, the system was deployed for use by external customers for five days from November 16-20th, 2020. This deployment was necessary for discerning failure modes in the architecture that were otherwise undetected in development and internal testing of the system.

Over the course of these five days, the SMADS completed 26 serviced trips on UT Campus. With the Anna Hiss Gymnasium (AHG) serving as the home location for each order, the robots autonomously navigated to either the Gates Dell Complex (GDC) or the UT Main Tower (MAIN) as the target destination. Figure 5 displays the map with the home location (AHG) and target locations (GDC, MAIN) labelled.

Figure 6: SMADS Robot Platforms

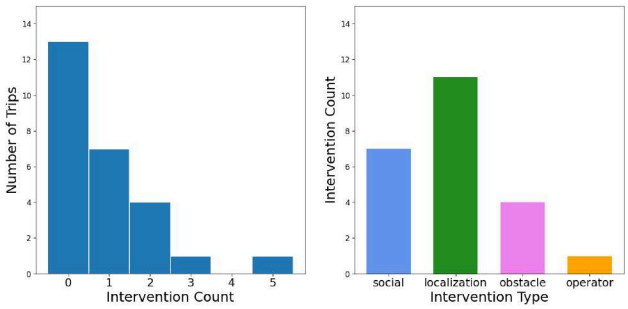

Figure 7: Manual Interventions

During each of these trips, the robot was accompanied by a robot safety operator to ensure safety of the robot and environment.

This deployment was carried out using a Clearpath Jackal and a Clearpath Husky (see Figure 6). Both are four wheeled, differential drive mobile platforms, equipped with a Velodyne VLP-16 3D Lidar and stereo FLIR RGB cameras. Over the deployment period, Jackal traversed a total distance of 9.67 km, and Husky traversed a total distance of 4.53 km. The SMADS Management Server retained a log of how many orders were placed, as well as which location had been serviced. Table 1 summarizes the distribution of these trips.

While all the trips in the deployment reached destination, some trips encountered a variety of issues along the way. Throughout the deployment, the robot faced issues with intermittent Wifi connection. The robot dropped its Wifi connection during a total of 6 runs. This connection loss caused display errors with the Texas Botler and iOS Manager apps.

Serviced trips to MAIN	8
Serviced trips to GDC	6
Serviced trips to AHG	12
Total serviced trips	**26**

Table 1: Summary of Trips Serviced

Other environmental issues affected the performance of the SMADS deliveries. These issues included physical barriers in the nominal robot path that prevented the robot from arriving to its destination, causing the customer to never see the Confirm Pickup screen (see Figure 2). Additionally, customers frequently failed to press the "confirm pickup" button once they retrieved the package. This behaviour indicates that it may not be natural for customers to interact with the Texas Bolter app user interface after receiving their order.

The robot safety operator accompanying the robot for these trips was responsible for intervening, or manually taking control of the robot, when the safety of the robot or humans in the environment was perceived to be at risk, or the robot seemed incapable of completing its current task autonomously. These manual interventions were tracked over the course of this deployment to help evaluate the robustness of SMADS.

Manual interventions largely represent issues with Robot Autonomy. Three issues that are of paramount interest include errors with localization, obstacle avoidance, and social navigation. Localization errors occurred when the robot's localization estimates differed greatly from its true position. Obstacle avoidance errors occurred when the robot's trajectory was highly unlikely to bypass static obstacles. Finally, social navigation errors occurred when the robot's trajectory was highly unlikely to bypass dynamic obstacles, such as moving cars, bicyclists, or crowds. Figure 7 summarizes the manual inventions performed throughout this deployment.

Figure 5 shows the distribution of locations where manual intervention were required for the Jackal and Husky (Babicki et al. 2016). The location marked "X" on the map denotes an error in obstacle avoidance caused by a university booth placed at the designated dropoff location for the MAIN building, thus requiring manual intervention.

Overall, the deployment on UT campus was a success, with 26 serviced trips throughout campus, half of which required no intervention on the part of the safety operators. This field deployment validated the feasibility of SMADS, and revealed a number of areas for improvement, informing the direction of future work on this project.

6 Conclusion and Future Work

In this work, we have introduced the SMADS framework as a means to enable autonomous deliveries on 2 different robots. We show successful use of the SMADS framework during a 5 day deployment on a college campus, performing 26 successful lemonade deliveries through the Texas Botler iOS app.

Though the deployment was overall a success, a number of issues arose that highlighted key areas for future improvement:

- **Robot Autonomy** From a technical perspective, Robot Autonomy was the largest source of failure, requiring manual intervention on about about half of the trips, despite using a state of the art autonomy stack. To address the failures we saw in these deployments, further research in the areas of social navigation and semantic-aware path planning, as well as more robust localization and obstacle avoidance, would significantly improve the performance of the system. Additionally, current methods of robot autonomy require a significant amount of human annotation and interaction for tasks such as initial localization and map construction. Future research that addresses these issues would also be of value to smooth deployment of the

SMADS architecture.

- **Connectivity** This real-world deployment highlighted the presence of connectivity issues, and the need to be robust against intermittent connection. During testing we found that using the IWD (iNet Wireless Daemon) as the networking back-end of the robots helped increase reliability during wireless hand-offs. Still, there remained problems with dropped network connection. To address these issues, the SMADS team plans to upgrade the robots' network connection to LTE service, rather than outdoor campus WiFi. Furthermore, predictive robot movement in the Texas Botler app could smooth the movement of the robot's location from the user's perspective over regions with intermittent connection loss.
- **User Experience** User interaction issues regarding confirming pick-up will be addressed in future implementations. The robot could implement sensors to detect if the delivered good was removed, notifying the SMADS Management Server. Then, the customer could receive a push notification that the order was delivered.

The SMADS team intends to collect further field data on new robotics platforms in the near future, and will be working on improvements to the areas enumerated above.

7 Acknowledgments

Thank you to the numerous student collaborators who worked to make this project possible. Thank you to Daksh Dua, Geethika Hemkumar, Marika Murphy, and Shikhar Gupta for your work to develop a simulation. Thank you to Anurag Patil and William Kwon for your iOS app contributions. Thank you to Kent Hansen for your work on the robot server. Finally, thank you to Ryan Gupta and Minkyu Kim for your guidance and efforts during testing.

References

Babicki, S.; Arndt, D.; Marcu, A.; Liang, Y.; Grant, J. R.; Maciejewski, A.; and Wishart, D. S. 2016. Heatmapper: web-enabled heat mapping for all. *Nucleic acids research* 44: W147–W153.

Bauer, A.; Klasing, K.; Lidoris, G.; Mühlbauer, Q.; Rohrmüller, F.; Sosnowski, S.; Xu, T.; Kühnlenz, K.; Wollherr, D.; and Buss, M. 2009. The autonomous city explorer: Towards natural human-robot interaction in urban environments. *International journal of social robotics* .

Biswas, J.; and Veloso, M. M. 2017. Episodic non-Markov localization. *Robotics and Autonomous Systems* 162 – 176.

Boysen, N.; Schwerdfeger, S.; and Weidinger, F. 2018. Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research* 271.

Chuan, L. P.; Johari, A.; Abd Wahab, M. H.; Nor, D. M.; Taujuddin, N. S. A. M.; and Ayob, M. E. 2007. An RFID warehouse robot. In *2007 International Conference on Intelligent and Advanced Systems*, 451–456.

Ehrenberg, I.; Floerkemeier, C.; and Sarma, S. 2007. Inventory management with an RFID-equipped mobile robot. In *2007 IEEE International Conference on Automation Science and Engineering*, 1020–1026.

Enright, J. J.; and Wurman, P. R. 2011. Optimization and coordinated autonomy in mobile fulfillment systems. In *Workshops at the twenty-fifth AAAI conference on artificial intelligence*.

Ferrer, G.; Garrell, A.; and Sanfeliu, A. 2013. Social-aware robot navigation in urban environments. In *2013 European Conference on Mobile Robots*, 331–336.

Jiang, Y.; Walker, N.; Kim, M.; Brissonneau, N.; Brown, D. S.; Hart, J. W.; Niekum, S.; Sentis, L.; and Stone, P. 2018. LAAIR: A layered architecture for autonomous interactive robots. *arXiv preprint arXiv:1811.03563* .

Khandelwal, P.; Zhang, S.; Sinapov, J.; Leonetti, M.; Thomason, J.; Yang, F.; Gori, I.; Svetlik, M.; Khante, P.; Lifschitz, V.; Aggarwal, J.; Mooney, R.; and Stone, P. 2017. BWIBots: A platform for bridging the gap between AI and human–robot interaction research. *The International Journal of Robotics Research* 36: 635 – 659.

Kümmerle, R.; Ruhnke, M.; Steder, B.; Stachniss, C.; and Burgard, W. 2015. Autonomous robot navigation in highly populated pedestrian zones. *Journal of Field Robotics* 32(4): 565–589.

Motroni, A.; Nepa, P.; Magnago, V.; Buffi, A.; Tellini, B.; Fontanelli, D.; and Macii, D. 2018. SAR-based indoor localization of UHF-RFID tags via mobile robot. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 1–8.

Pani, A.; Mishra, S.; Golias, M.; and Figliozzi, M. 2020. Evaluating public acceptance of autonomous delivery robots during COVID-19 pandemic. *Transportation Research Part D: Transport and Environment* 89: 102600.

Shah, R.; Jiang, Y.; Karnan, H.; Briscoe-Martinez, G.; Mulder, D.; Gupta, R.; Schlossman, R.; Murphy, M.; Hart, J. W.; Sentis, L.; et al. 2019. Solving Service Robot Tasks: UT Austin Villa@ Home 2019 Team Report. *arXiv preprint arXiv:1909.06529* .

Veloso, M.; Biswas, J.; Coltin, B.; and Rosenthal, S. 2015a. CoBots: Robust Symbiotic Autonomous Mobile Service Robots. In *IJCAI*.

Veloso, M.; Biswas, J.; Coltin, B.; and Rosenthal, S. 2015b. CoBots: Robust Symbiotic Autonomous Mobile Service Robots. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press.

Wisspeintner, T.; Van Der Zant, T.; Iocchi, L.; and Schiffer, S. 2009. RoboCup@ Home: Scientific competition and benchmarking for domestic service robots. *Interaction Studies* 10: 392–426.

Wurman, P. R.; D'Andrea, R.; and Mountz, M. 2008. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine* 29: 9–9.

Zhang, J.; Lyu, Y.; Roppel, T.; Patton, J.; and Senthilkumar, C. 2016. Mobile robot for retail inventory using RFID. In *2016 IEEE international conference on Industrial technology (ICIT)*, 101–106.