Robotically Enhanced Snack Dispenser and Retrieval System

JosieKate Cooley, Jacqueline Gibson, and Ashvin Govil

Abstract

The goal of this project was to implement a snack delivery system, in which a robot would receive a snack order from a human, send the order to a vending machine, and retrieve the item, ultimately delivering it to the requesting human. This task was divided into two subparts: the node operating the vending machine and the node that controls the robot's operations. The remainder of the paper details the technical approach to these two tasks and discusses the difficulties encountered throughout the project. Within the paper, the designs for the nodes and the GUI are detailed, the process for the simulation is described, and several ideas regarding future applications as well as expounding upon the initial work are discussed.

Introduction

Our end goal was to create a simple, yet functional vending machine delivery system. We created a node that the vending machine could use to dispense the correct item and let the robot know the item is ready for pickup. A related node allows the robot to communicate with the vending machine via a ROS service and client pair, so the information about the snack's location can be relayed quickly and easily. The simulator and its GUI show how the robot would receive a request from a human, save its location (to which it will return with the snack), send the snack order to the vending machine; next, it travels to the vending machine, where it remains until a human indicates that the correct snack is on the robot. It then returns to the saved location to deliver the item to the human who made the initial request. More testing is required to see if this is actually more convenient than humans going to the vending machine themselves, and whether improvements on the project can make it a viable source of future research.

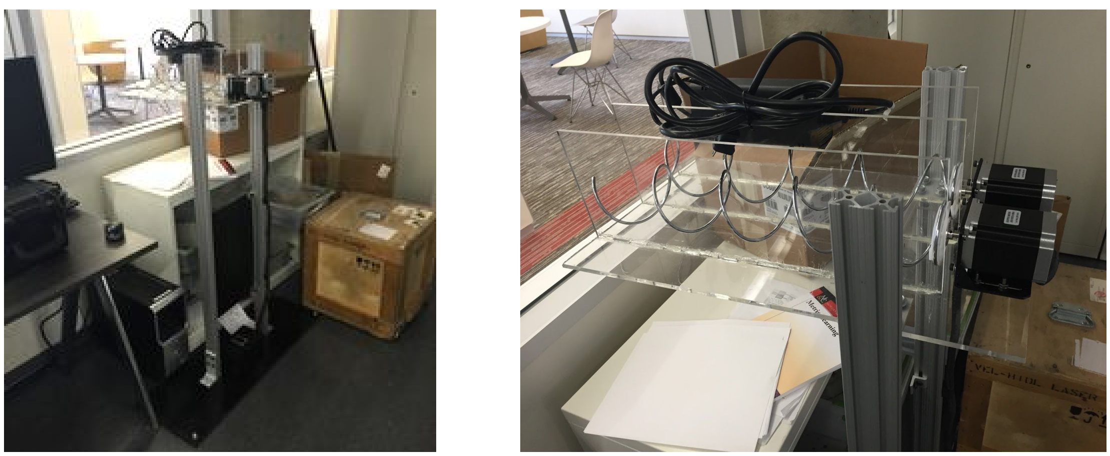
Background and Related Work

The idea of robots delivering snacks is not a new one, but it is still not often implemented. This poses an issue in relation to testing the usefulness of such systems: it can be difficult to determine whether people are using it for efficiency or novelty (Lee). Service robots have proven to be helpful and entertaining (at least in the short term), and give employees time to focus on more important tasks than running errands. However, programming these robots has proven to be something of a challenge - in order for them to be truly useful, non-technical employees need the ability to make simple modifications without breaking the robot or otherwise taking it out of commission for too long (Huang). Currently, robots are used in the service industry on small scales for simples, singular tasks. For example, there are vending machines that put down a cup, brew coffee, then pour said coffee into a cup for the person ordering (Coffee). But there are no robots that could, for example, take an order from someone sitting at a table in a coffee shop made on a smartphone, make them coffee, figure out where that person is sitting, and bring the coffee to that person without running into anyone in the way. Doing this entire problem with one robot would be especially difficult, considering how different each task is. Solving this problem requires a good understanding of cooperative robotics, which splits up larger problems like the coffee problem into smaller, simpler problems that are easier and perhaps more cost-effective to solve (Cao). A cooperative system of robots has some underlying system controlling all of them. For us, this is easy -- ROS is this underlying system. Using nodes and service clients in ROS makes communicating between robots while operating them under one system relatively simple. The tricky part is making all parts of the system operate correctly regardless of varying circumstances, which is harder with each component added to the system. When designing such complex systems, it is useful to take into account what the system will be going through on a daily basis, and months or years down the line. Ideally, a system for delivering snacks would be able to adjust to the changes in snack consumption over the course of time, and might even be able to nudge people towards healthier options as it learned their snacking requirements and habits (Lee, Vezzoli).

Technical Approach

Robot_Node

The main functionality of the node is moderating the process of dispensing and retrieving the snack. First, the node asks for user input, allowing the participant to enter whether they want “goldfish” or “izze,” or nothing. Based on the response, the robot sends the appropriate instructions to the vending machine, so the appropriate item is dispensed if an order (e.g. a request for the snack or the drink option) is placed. After receiving a notification of completion from the vending machine, the node sends the bot to the vending machine. A popup appears prompting nearby humans to retrieve the item and place it on the bot (it should be noted that items should be placed only on turtlebots, not the segbot we use for testing). The robot then returns to the original location and asks for feedback (e.g. was the correct item delivered? Was it delivered in a timely manner?).



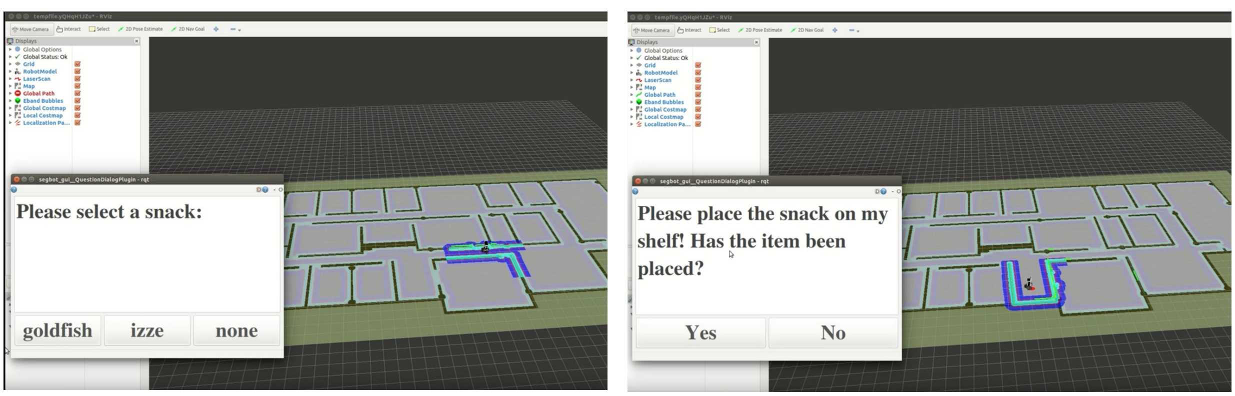
The vending machine, built by students at UT Austin - runs using stepper motors

Vending_Machine_Node

The main functionality of the node is dispensing the proper snack and serving as an intermediate stop for the robot. The node subscribes to the messages published by ***Robot_Node***. According to the received message, the node interfaces with the Raspberry Pi, prompting the correct slot to push out the snack. After this is complete, the node publishes a message that the action is complete.

Simulation and GUI

This project took advantage of the ***RViz*** virtual environment modified by UT's ***bwi_launch*** package. The simulator used was ***simulation.launch***, and the GUI a modified version of ***visit_door_list_gu***i from the ***bwi_tasks*** package. The robot goes door-to-door using a predetermined list of known door locations, and at each door enquires whether anyone in the room would like a snack. The user selects the desired snack from a list of options, and the robot is sent to the vending machine's location. The robot waits at this location until a different user has indicated via the GUI that the correct snack has been successfully placed on the robot's shelf, at which point it returns to the door of the snack requester and waits for the snack to be retrieved before resuming its door-to-door rounds.



A mockup of the final GUI, used to direct a simulated version of the robot

Evaluation

Testing during this phase of the project was completed via the simulator, as hardware and schedule complexity prevented testing on the planned robots. The simulator is more efficient than the actual robots, but prone to doing things like running through walls and moving at quicker speeds than those safely used by real-world robots. Despite these imperfections, the simulation yielded valuable testing data. It offered an environment to test the GUI, which led to improvement of the GUI and increased knowledge of how people would interact with the robot and the vending machine by proxy. The entire system proved to run somewhat slower than anticipated, and the testing highlighted that the robots are not often milling about the lab - the convenience of having a robot fetch a snack is greatly lessened by having to go into a different room, turn on said robot, run its startup sequence, then tell it to go to another room to take an order. The system itself worked well, although it relies somewhat heavily on human help and input and would likely not perform well in cases of heavy demand in its current form.

Difficulties and Obstacles

The delay in arrival of hardware parts, stemming mainly from the growing list of accessories needed to make the Raspberry Pi more accessible (e.g. a wireless keyboard, a wifi Dongle, HDMI cord so the Raspberry Pi output could be projected onto a monitor for ease in coding and software installation, and general Raspberry Pi necessities such as the charger) proved to be the initial obstacle. Once all hardware pieces were received, the group faced difficulties with the actual Raspberry Pi computer. First, it became evident that the group needed a SD card on which the NOOBS operating system could be loaded; this had to be ordered and completely formatted (e.g. loading the necessary software) before work could resume. Second, despite using various HDMI cables and several monitors and laptops, the Raspberry Pi display could not be accessed. It seemed the Raspberry Pi was consistently failing to send output, and after discussion with Professor Sinapov, running the node in a virtual simulator was determined to be best suited for the initial stage of the project. Lastly, creating and maintaining a Git repository proved difficult, as the specific command-line prompts needed to be learned and cloning the repository on a combination of lab and personal machines proved problematic at times. In terms of actual code, determining how best to get information from humans, as well as what that input should be, proved to be a major obstacle. This was compounded by the vending machine itself not yet functioning, resulting in a moderate amount of guesswork involved in writing the GUI.

Conclusion and Future Work

This project could easily be selected by a different group and continued, taken in a different direction, or expanded upon. Beyond continued work on the efficiency and functionality of the two nodes presented here, the efficiency and functionality of the entire system could be improved. The GUI could be updated to include a more thorough list of snack options, and some sort of sensor or inventory could be added to the vending machine and its code to facilitate this. The robot could also potentially carry snacks - it has the space for a dedicated snack compartment, and the practice seems to have worked at other universities. The vending machine itself could probably be expanded to hold more snacks. If the snack delivery system works well enough, it could also be expanded to hold and deliver more than snacks. It would be ideal if an alternative to simply waiting for a helpful human to come along and move things onto the robot could be implemented, making the robot more autonomous as opposed to spreading work among humans. Text-to-speech could also be employed to make the process simpler for humans, using pre-existing ROS packages such as ***sound_play***. It could also be interesting to include speech recognition along with text-to-speech, to essentially eliminate the GUI and streamline the snack-ordering process.

As in the simulation, it would be useful to include a functionality that involved detecting people in the rooms visited, or perhaps when roaming the hallways. The robot is often moving, as are people, and it would be challenging but probably worthwhile to take orders while one or both parties are moving. Because this would present problems with getting a location to deliver the snack, it would also be convenient to get a delivery location from the human. This could also allow people to send snacks to one another, which would be an entertaining application of the snack delivery system. Additionally, taking instructions via email, text, or even a simple website would be another useful application, although user input is notoriously unreliable and could be difficult to account for. Decreasing unreliability in other areas, such as snack availability, would be easier. As mentioned before, sensors could be used to keep track of snacks ordered, or code alone could track what had been ordered, although this would again depend on human input of what the machine initially contained. The robot could also potentially keep track of multiple snack orders at once, then deliver all the snacks in a more efficient manner. There are a number of possible improvements to this system, of varying levels of difficulty and achievability. Now that the preliminary stages of this project are complete, the work this group has done can serve as a solid base for future snack-related endeavors.

Sources

Baker, Mary, and Justin Manweiler. "Notes from the Community: Drones, Robots, and Sushi!" IEEE CS(2016): 93-97. IEEEXplore. Web. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7389277>.

Cao, Y. Uny, Alex S. Fukunaga, and Andrew Khang. "Cooperative Mobile Robotics: Antecedents and Directions." Autonomous Robots 4.1 (1997): 7-27. Springer Link. Web. <http://te7fv6dm8k.scholar.serialssolutions.com/?sid=google&auinit=YU&aulast=Cao&atitle=Cooperative+mobile+robotics:+Antecedents+and+directions&id=doi:10.1023/A:1008855018923&title=Autonomous+robots&volume=4&issue=1&date=1997&spage=7&issn=0929-5593>.

"Coffee Vending MachineGrande Milano." Coffee Vending Machine. N.p., n.d. Web. 11 May 2016.

Huang, Justin, Tessa Lau, and Maya Cakmak. 2016. Design and Evaluation of a Rapid Programming System for Service Robots. In The Eleventh ACM/IEEE International Conference on Human Robot Interation (HRI '16). IEEE Press, Piscataway, NJ, USA, 295-302. <http://delivery.acm.org/10.1145/2910000/2906883/p295-huang.pdf?ip=128.62.55.213&id=2906883&acc=ACTIVE%20SERVICE&key=603D2E7028CD4EF5%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=779835498&CFTOKEN=28613124&__acm__=1462233540_e8d226c62ead7fdf16aca6a0efd4fe26>.

Lee, Min Kyung, Jodi Forlizzi, Paul E. Rybski, Frederick Crabbe, Wayne Chung, Josh Finkle, Eric Glaser, and Sara Kiesler. "The Snackbot: Document the Design of a Robot for Long-term Human-Robot Interaction." Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction - HRI '09 (2009): 7-14. Web. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6256098>.

Moshkina, Lilia, and Frank Meyer. "Neato Robotics® Robots as a Robust Mobile Base for Modular HRI Research." Association for the Advancement of Artificial Intelligence

Spring Symposium Series (2016): 151-54. Web. <http://www.aaai.org/ocs/index.php/SSS/SSS16/paper/view/12749/11938>.

Vezzoli, Carlo, and Dalia Sciama. "Life Cycle Design: From General Methods to Product Type Specific Guidelines and Checklists: A Method Adopted to Develop a Set of Guidelines/checklist Handbook for the Eco-efficient Design of NECTA Vending Machines." Journal of Cleaner Production 14.15-16 (2006): 1319-325. ScienceDirect. Web. <http://www.sciencedirect.com/science/article/pii/S0959652605002398>.